Mechanism Design in Large Games: Incentives and Privacy *

Michael Kearns † Mallesh M. Pai ‡ Aaron Roth § Jonathan Ullman ¶

September 8, 2013

Abstract

We study the problem of implementing equilibria of complete information games in settings of incomplete information, and address this problem using "recommender mechanisms." A recommender mechanism is one that does not have the power to enforce outcomes or to force participation, rather it only has the power to suggestion outcomes on the basis of voluntary participation. We show that despite these restrictions, recommender mechanisms can implement equilibria of complete information games in settings of incomplete information under the condition that the game is *large*—i.e. that there are a large number of players, and any player's action affects any other's payoff by at most a small amount.

Our result follows from a novel application of *differential privacy*. We show that any algorithm that computes a correlated equilibrium of a complete information game while satisfying a variant of differential privacy—which we call *joint differential privacy*—can be used as a recommender mechanism while satisfying our desired incentive properties. Our main technical result is an algorithm for computing a correlated equilibrium of a large game while satisfying joint differential privacy.

Although our recommender mechanisms are designed to satisfy game-theoretic properties, our solution ends up satisfying a strong privacy property as well. No group of players can learn "much" about the type of any player outside the group from the recommendations of the mechanism, even if these players collude in an arbitrary way. As such, our algorithm is able to implement equilibria of complete information games, without revealing information about the realized types.

*We gratefully acknowledge the support of NSF Grant CCF-1101389. We thank Nabil Al-Najjar, Eduardo Azevdeo, Eric Budish, Tymofiy Mylovanov, Andy Postlewaite, Al Roth and Tim Roughgarden for helpful comments and discussions.

†Department of Computer and Information Science, University of Pennsylvania.

‡Department of Economics, University of Pennsylvania.

§Department of Computer and Information Science, University of Pennsylvania.

¶School of Engineering and Applied Sciences, Harvard University.

Contents

1 Introduction

A useful simplification common in game theory is the model of *games of complete (or full) information*. Informally, in a game of complete information, each player knows with certainty the utility function of every other player. In games of complete information, there are a number of solution concepts at our disposal, such as Nash equilibrium and correlated equilibrium. Common to these is the idea that each player is playing a best response against his opponents—because of randomness, players might be uncertain about what actions their opponents are taking, but they understand their opponents' incentives exactly.

In many situations, it is unreasonable to assume that players have exact knowledge of each other's utilities. For example, players may have few means of communication outside of the game, or may regard their type as valuable private information. These are *games of incomplete (or partial) information*, which are commonly modeled as Bayesian games in which the players' utilities functions, or types, are drawn from a commonly known prior distribution. The most common solution concept in such games is Bayes-Nash Equilibrium. This stipulates that every player i, as a function of his type, plays an action that maximizes his expected payoff, in expectation *both* over the random draw of the types of his opponents from the prior distribution and over the possible randomization of the other players.

Unsurprisingly, equilibrium welfare can suffer in games of incomplete information, because coordination becomes more difficult amongst players who do not know each other's types. One way to measure the quality of equilibria is via the "price of anarchy"—how much worse the social welfare can be in an equilibrium outcome, as opposed to the welfare-maximizing outcome. The price of anarchy can depend significantly on the notion of equilibrium. For example, Roughgarden [Rou12] notes that even smooth games[1] that have a constant price of anarchy under full information solution concepts (e.g. Nash or correlated equilibrium) can have an unboundedly large price of anarchy under partial information solution concepts (e.g. Bayes-Nash Equilibrium). Therefore, given a game of partial information, where all that we can predict is that players will choose some Bayes-Nash Equilibrium (if even that), it may be preferable to implement an equilibrium of the complete information game defined by the actual realized types of the players. Doing so would guarantee welfare bounded by the price of anarchy of the full information game, rather than suffering from the large price of anarchy of the partial information setting. In a smooth game, we would be just as happy implementing a correlated equilibrium as a Nash equilibrium, since the price of anarchy is no worse over correlated equilibria.

In this paper we ask whether it is possible to help coordinate on an equilibrium of the realized full information game using a certain type of proxy that we call a "recommender mechanism." That is, we augment the game with an additional option for each player to use a proxy. If players opt in to using the proxy, and reveal their type to the proxy, then it will suggest some action for them to take. However, players may also simply opt out of using the proxy and play the original game using any strategy they choose. We make the assumption that if players use the proxy, then they must report their type truthfully (or, alternatively, that the proxy has the ability to verify a player's type and punish those who report dishonestly). However, the proxy has very limited power in other respects, because it does not have the ability to modify payoffs of the game (i.e. make payments) or to enforce that its recommendations be followed.

Our main result is that it is indeed possible to implement approximate correlated equilibria of the realized full information game using recommender mechanisms, assuming the original game is "large". Informally,

[1] Of particular interest to us are *smooth games*, defined by Roughgarden [Rou09]. Almost all known price of anarchy bounds (including those for the well studied model of traffic routing games) are bounds on smooth games, and many are quite good. Although price of anarchy bounds are typically proven for exact Nash equilibria of the full information games, in smooth games, the price of anarchy bounds extend without loss to (and even beyond) approximate correlated equilibria, again of the full information game.

a game is large if there are many players and that each player has individually only a small affect on the utility of any other player. We show that in such games there exists a recommender mechanism such that for any prior on agent types, it is an approximate Bayes-Nash equilibrium for every agent in the game to opt in to the proxy, and then follow its recommended action. Moreover, when players do so, the resulting play forms an approximate correlated equilibrium of the full information game. The approximation error we require tends to 0 as the number of players grows.

1.1 Overview of Techniques and Results

A tempting approach is to use the following form of proxy: The proxy accepts a report of each agent's type, which defines an instance of a full information game. It then computes a correlated equilibrium of the full information game, and suggests an action to each player which is a draw from this correlated equilibrium. By definition of a correlated equilibrium, if all players opt into the proxy, then they can do no better than subsequently following the recommended action. However, this proxy does not solve the problem, as it may not be in a player's best interest to opt in, even if the other $n-1$ players do opt in! Intuitively, by opting out, the player can cause the proxy to compute a correlated equilibrium of the wrong game, or to compute a different correlated equilibrium of the same game![2] The problem is an instance of the well known equilibrium-selection problem—even in a game of full information, different players may disagree on their preferred equilibrium, and may have trouble coordinating. The problem is only more difficult in settings of incomplete information. In our case, by opting out of the mechanism, a player can have a substantial affect on the computed equilibrium, even if each player has only small affect on the utilities of other players.

Our solution is to devise a means of computing correlated equilibria such that any single player's reported type to the algorithm only has a small effect on the distribution of suggested actions to all other players. The precise notion of "small effect" that we use is a variant of the well studied notion of *differential privacy*. It is not hard to see that computing an equilibrium of even a large game is not possible under the standard constraint of differential privacy, because although agent's actions have only a small affect on the utilities of other players in large games, they can have large affect on their own utility functions. Thus, it is not possible to privately announce a best response for player i while protecting the privacy of i's type. Instead, we introduce a variant which we call *joint differential privacy*, which requires that simultaneously for every player i, the joint distribution on the suggested actions to all players $j \neq i$ be differentially private in the type of agent i. We show that a proxy mechanism which calculates an α-approximate correlated equilibrium of the game induced by players reported types, under the constraint of ϵ-joint differential privacy makes it an $(\epsilon + \alpha)$-approximate Bayes-Nash equilibrium for players to opt into the proxy, and then follow their suggested action, as desired.

[2]As a simple example, consider a large number n of people who must each choose whether to go to the beach (B) or mountains (M). People privately know their types— each person's utility depends on his own type, his action, and the fraction of other people p who go to the beach. A Beach type gets a payoff of $10p$ if he visits the beach, and $5(1-p)$ if he visits the mountain. A mountain type gets a payoff $5p$ from visiting the beach, and $10(1-p)$ from visiting the mountain. Note that the game is 'insensitive' (an agent's visit decision has a small impact on others' payoffs). Further, note that "everyone visits beach" and "everyone visits mountain" are both equilibria of the game, regardless of the realization of types. Consider the mechanism that attempts to implement the following social choice rule—"if the number of beach types is less than half the population, send everyone to the beach, and vice versa." It should be clear that if mountain types are just in the majority, then each mountain type has an incentive to opt out of the mechanism, and vice versa. As a result, even though the game is "large" and agents' actions do not affect others' payoffs significantly, simply computing equilibria from reported type profiles does not in general lead to even approximately truthful mechanisms. This is a general phenomenon that is not specific to our example. Finding an exact correlated equilibrium subject to any objective is a linear programming problem, and in general small changes in the objective (or constraints) of an LP can lead to wild changes in its solution.

Our main technical result is an instantiation of this plan: a pair of algorithms for computing α-approximate correlated equilibria in large games, such that we can take the approximation parameter $\epsilon + \alpha$ tending to zero. The first algorithm is efficient, but has a suboptimal dependence on the number of actions k in the game. The other algorithm is inefficient, but has a nearly optimal dependence on k. Both have an optimal dependence on the number of players n in the game, which we show by exhibiting a matching lower bound.

We introduce joint differential privacy, large games, and our game theoretic solution concepts in Section 2. In Section 3, we formally introduce our notion of a proxy. We then prove that privately computing a correlated equilibrium is sufficient to implement a correlated equilibrium of the full information game as a Bayes Nash equilibrium of the incomplete information game with a proxy where agents can opt out. Then, starting in Section 4, we show how to privately compute correlated equilibria, which we do by using no-regret algorithms together with carefully calibrated noise.

1.2 Related Work and Discussion

Market and Mechanism Design Our work is related to the large body of literature on mechanism/ market design in "large games," which uses the large number of agents to provide mechanisms which have good incentive properties, even when the small market versions do not. It stretches back to [RP76] who showed that market (Walrasian) equilibria are approximately strategy proof in large economies. More recently [IM05], [KP09], [KPR10] have shown that various two-sided matching mechanisms are approximately strategy proof in large markets. There are similar results in the literature for one-sided matching markets, market economies, and double auctions. The most general result is that of [AB11] who design incentive compatible mechanisms for large economies that satisfy a smoothness assumption. While we only allow agents to opt in/ opt out rather than mis-report, we do not assume any such smoothness condition. Further, the literature on mechanism design normally gives the mechanism the power to "enforce" actions, while here our mechanism can only "recommend" actions.

Our work is also related to mediators in games [MT03, MT09]. This line of work aims to modify the equilibrium structure of full information games by introducing a mediator, which can coordinate agent actions if they choose to opt in using the mediator. Mediators can be used to convert Nash equilibria into dominant strategy equilibria [MT03], or implement equilibrium that are robust to collusion [MT09]. Our notion of a recommender mechanism is related, but is even weaker than that of a mediator. For example, our mechanisms do not need the power to make payments [MT03], or the power to enforce suggested actions [MT09]. Our mediators are thus closer to the communication devices in the "communication equilibria" of Forges [For86]—that work investigates the set of achievable payoffs via such mediators rather than how to design one, which we do here. It also does not allow players to opt out of using the mediator.

Large Games Our results hold under a "largeness condition", i.e. a player's action affects the payoff of all others by a small amount. These are closely related to the literature on large games, see e.g. [ANS00] or [Kal04]. There has been recent work studying large games using tools from theoretical computer science (but in this case, studying robustness of equilibrium concepts)—see [GR08, GR10].

Differential Privacy Differential privacy was first defined by [DMNS06], and is now the standard privacy "solution concept" in the theoretical computer science literature. It quantifies the *worst-case harm* that can befall an individual from allowing his data to be used in a computation, as compared to if he did not provide his data. There is by now a very large literature on differential privacy, readers can consult [Dwo08] for a more thorough introduction to the field. Here we mention work at the intersection of privacy and game theory, and defer a longer discussion of related work in the privacy literature to Appendix A.

[MT07] were the first to observe that a differentially private algorithm is also approximately truthful. This line of work was extended by [NST12] to give mechanisms in several special cases which are exactly truthful by combining private mechanisms with non-private mechanisms which explicitly punish non-truthful reporting. [HK12] showed that the mechanism of [MT07] (the "exponential mechanism") is in fact maximal in distributional range, and so can be made exactly truthful with the addition of payments. This immediate connection between privacy and truthfulness does not carry over to the notion of joint-differential privacy that we study here, but as we show, it is regained if the object that we compute privately is an equilibrium of the underlying game.

Another interesting line of work considers the problem of designing truthful mechanisms for agents who explicitly experience a cost for privacy loss as part of their utility function [CCK+13, NOS12, Xia13]. The main challenge in this line of work is to formulate a reasonable model for how agents experience cost as a function of privacy. We remark that the approaches taken in the former two can also be adapted to work in our setting, for agents who explicitly value privacy. [Gra12] studies the problem of implementation for various assumptions about players' preference for privacy and permissible game forms. A related line of work which also takes into account agent values for privacy considers the problem of designing markets by which analysts can procure private data from agents who explicitly experience costs for privacy loss [FL12, GR11, LR12, RS12]. See [PR13] for a survey.

2 Model & Preliminaries

We consider games $\mathcal{G}$ of up to n players $\{1, 2, \ldots, n\}$, indexed by i. Player i can take actions in a set A, $|A| = k$. To allow our games to be defined also for fewer than n players, we will imagine that the null action $\perp \in A$, which corresponds to "opting out" of the game. We index actions by j. A tuple of actions, one for each player, will be denoted $\mathbf{a} = (a_1, a_2, \ldots a_n) \in A^n$.[3]

Let $\mathcal{U}$ be the set of player types.[4] There is a utility function $u : \mathcal{U} \times A^n \to \Re$ that determines the payoff for a player given his type t_i and a joint action profile $\mathbf{a}$ for all players. When it is clear from context, we will refer to the utility function of player i, writing $u_i : A^n \to \Re$ to denote $u(t_i, \cdot)$. We write a generic profile of utilities $\mathbf{u} = (u_1, u_2, \ldots u_n) \in \mathcal{U}^n$. We will be interested in implementing equilibria of the *complete information game* in settings of *incomplete information*. In the complete information setting, the types t_i of each player is fixed and commonly known to all players. In such settings, we can ignore the abstraction of 'types' and consider each player i simply to have a fixed utility function u_i. In models of *incomplete* information, players know their own type, but do not know the types of others. In the Bayesian model of incomplete information, there is a commonly known prior distribution τ from which each agent's type is jointly drawn: $(t_1, \ldots, t_n) \sim \tau$. We now define the solution concepts we will use, both in the full information setting and in the Bayesian setting.

Denote a distribution over A^n by π, the marginal distribution over the actions of player i by π_i, and the marginal distribution over the (joint tuple of) actions of every player but player i by π_{-i}. We now present two standard solution concepts— approximate correlated and coarse correlated equilibrium.

Definition 1 (Approximate Correlated Equilibrium)**.** Let $(u_1, u_2, \ldots u_n)$ be a tuple of utility functions, one for each player. Let π be a distribution over tuples of actions A^n. We say that π is an α-*approximate correlated equilibrium* of the (complete information) game defined by $(u_1, u_2, \ldots u_n)$ if for every player

[3] In general, subscripts will refer indices i.e. players and periods, while superscripts will refer to components of vectors.

[4] It is trivial to extend our results when agents have different typesets, $\mathcal{U}_i$. $\mathcal{U}$ will then be $\bigcup_{i=1}^{n} \mathcal{U}_i$.

$i \in [N]$, and any function $f\colon A \to A$,

$$\mathop{\mathbb{E}}_{\pi}\left[u_i(\mathbf{a})\right] \geq \mathop{\mathbb{E}}_{\pi}\left[u_i(f(a_i), a_{-i})\right] - \alpha$$

We now define a solution concept in the Bayesian model. Let τ be a commonly known joint distribution over $\mathcal{U}^n$, and let $\tau_{|t_i}$ be the posterior distribution on types conditioned on the type of player i being t_i. A (pure) *strategy* for player i is a function $s_i : \mathcal{U} \to A$, and we write $\mathbf{s} = (s_1, \ldots, s_n)$ to denote a vector of strategy profiles.

Definition 2 (Approximate (Pure Strategy) Bayes Nash Equilibrium)**.** Let τ be a distribution over $\mathcal{U}^n$, and let $\mathbf{s} = (s_1, \ldots, s_n)$ be a vector of strategies. We say that $\mathbf{s}$ is an *α-approximate (pure strategy) Bayes Nash Equilibrium* under τ if for every player i, for every $t_i \in \mathcal{U}$, and for every alternative strategy s'_i:

$$\mathop{\mathbb{E}}_{\mathbf{t}_{-i} \sim \tau_{|t_i}}\left[u_i(t_i, s_i(t_i), s_{-i}(t_{-i}))\right] \geq \mathop{\mathbb{E}}_{\mathbf{t}_{-i} \sim \tau_{|t_i}}\left[u_i(t_i, s'_i(t_i), s_{-i}(t_{-i}))\right] - \epsilon$$

We restrict attention to 'insensitive' games. Roughly speaking a game is γ-sensitive if a player's choice of action affects any other player's payoff by at most γ. Note that we do not constrain the effect of a player's *own* actions on his payoff— *a player's action can have a large impact on his own payoff*. Formally:

Definition 3 (γ-Sensitive)**.** A game is said to be γ-sensitive if for any two distinct players $i \neq i'$, any two actions a_i, a'_i and type t_i for player i and any tuple of actions a_{-i} for everyone else:

$$|u_{i'}(a_i, a_{-i}) - u_{i'}(a'_i, a_{-i})| \leq \gamma. \tag{1}$$

A key tool in our paper is the design of differentially private "proxy" algorithms for suggesting actions to play. Agents can able to opt out of participating in the proxy: so each agent can submit to the proxy either their type t_i, or else a null symbol $\perp$ which represents opting out. A proxy algorithm is then a function from a profile of utility functions (and $\perp$ symbols) to a probability distribution over $\mathcal{R}^n$, i.e. $\mathcal{M}\colon (\mathcal{U} \cup \{\perp\})^n \to \Delta\mathcal{R}^n$. Here $\mathcal{R}$ is an appropriately defined range space.

First we recall the definition of differential privacy, both to provide a basis for our modified definition, and since it will be a technical building block in our algorithms. Roughly speaking, a mechanism is differentially private if for every $\mathbf{u}$ and every i, knowledge of the output $\mathcal{M}(\mathbf{u})$ as well as u_{-i} does not reveal 'much' about u_i.

Definition 4 ((Standard) Differential Privacy)**.** A mechanism $\mathcal{M}$ satisfies (ε, δ)-differential privacy if for any player i, any two types for player i, t_i and $t'_i \in \mathcal{U} \cup \{\perp\}$, and any tuple of types for every else $t_{-i} \in (\mathcal{U} \cup \{\perp\})^{n-1}$ and any $S \subseteq \mathcal{R}^n$,

$$\mathop{\mathbb{P}}_{\mathcal{M}}\left[(\mathcal{M}(t_i; t_{-i})) \in S\right] \leq e^{\varepsilon} \mathop{\mathbb{P}}_{\mathcal{M}}\left[\left(\mathcal{M}(t'_i; t_{-i})\right) \in S\right] + \delta.$$

We would like something slightly different for our setting. We propose a relaxation of the above definition, motivated by the fact that when computing a correlated equilibrium, the action recommended to a player is only observed by her. Roughly speaking, a mechanism is *jointly differentially private* if, for each player i, knowledge of the other $n-1$ recommendations (and submitted types) does not reveal 'much' about player i's report. Note that this relaxation is necessary in our setting if we are going to privately compute correlated equilibria, since knowledge of player i's recommended action can reveal a lot of information about his type. It is still very strong- the privacy guarantee remains *even if* everyone else colludes against a given player i, so long as i does not himself make the component reported to him public.

Definition 5 (Joint Differential Privacy)**.** A mechanism $\mathcal{M}$ satisfies (ε, δ)-joint differential privacy if for any player i, any two possible types for player i, t_i and $t_i' \in \mathcal{U} \cup \{\perp\}$, any tuple of utilities for everyone else t_{-i} and $S \subseteq \mathcal{R}^{n-1}$,

$$\mathop{\mathbb{P}}_{\mathcal{M}}\left[(\mathcal{M}(t_i; t_{-i}))_{-i} \in S\right] \leq e^{\varepsilon} \mathop{\mathbb{P}}_{\mathcal{M}}\left[\left(\mathcal{M}(t_i'; t_{-i})\right)_{-i} \in S\right] + \delta.$$

3 Joint Differential Privacy and Truthfulness

The main result of this paper is a reduction that takes an arbitrary large game $\mathcal{G}$ of incomplete information and modifies it to have equilibrium implementing equilibrium outcomes of the corresponding full information game defined by the *realized* agent types. Specifically, we modify the game by introducing the option for players to use a *proxy* that can recommend actions to the players. The modified game is called $\mathcal{G}'$. For any prior on agent types, it will be an approximate Bayes Nash equilibrium of $\mathcal{G}'$ for every player to opt in to using the proxy, and to subsequently follow its recommendation. Moreover, the resulting set of actions will correspond to an approximate correlated equilibrium of the complete information game $\mathcal{G}$ defined by the realized agent types. For concreteness, we consider Bayesian games, however our results are not specific to this model of incomplete information.

More precisely, the modified game $\mathcal{G}'$ will be identical to $\mathcal{G}$ with an added option. Each player i has the opportunity to submit their type to a proxy, which will then suggest to them an action $\hat{a}_i \in A$ to play. They can use this advice however they like: that is, they can choose any function $f : A \to A$ and choose to play the action $a_i = f(\hat{a}_i)$. Alternately, they can opt out of the proxy (and not submit their type), and choose an action to play $a_i \in A$ directly. In the end, each player experiences utility $u(t_i, (a_1, \ldots, a_n))$, just as in the original game $\mathcal{G}$. We assume that types are verifiable—agent i does not have the ability to opt in to the proxy but report a false type $t_i' \neq t_i$. However, he does have the ability to opt out (and submit $\perp$), and the proxy has no power to do anything other than suggest which action he should play. In the end, each player is free to play any action a_i, regardless of what the proxy suggests, even if he opts in.

Formally, given a game $\mathcal{G}$ defined by an action set A, a type space $\mathcal{U}$, and a utility function u, we define a *proxy game* $\mathcal{G}'_M$, parameterized by an algorithm $M : \{\mathcal{U} \cup \{\perp\}\}^n \to A^n$. In $\mathcal{G}'$, each agent has two types of actions: they can *opt in* to the proxy, which means they submit their type, receive an action recommendation $\hat{a}$, and choose a function $f : A \to A$ which determines how they use that recommendation. We denote this set of choices $A_1' = \{(\top, f) | f : A \to A\}$. Alternately, they can *opt out* of the proxy, which means that they do not submit their type, and directly choose an action to play. We denote this set of choices $A_2' = \{(\perp, a) | a \in A\}$. Together, the action set in $\mathcal{G}'_M$ is $A' = A_1' \cup A_2'$. Given a set of choices by the players, we define a vector $\mathbf{x}$ such that $\mathbf{x}_i = t_i$ for each player i who chose $(\top, f_i) \in A_1'$ (each player who opted in), and $\mathbf{x}_i = \perp$ for each player i who chose $(\perp, a_i) \in A_2'$ (each player who opted out). The proxy then computes $M(\mathbf{x}) = \hat{\mathbf{a}}$. Finally, this results in a vector of actions $\mathbf{a}$ from the game $\mathcal{G}$, one for each player. For each player who opted in, they play the action $\mathbf{a}_i = f_i(\hat{\mathbf{a}}_i)$. For each person who opted out, they play the action $\mathbf{a}_i = a_i$. Finally, each player receives utility $u(t_i, \mathbf{a})$ as in the original game $\mathcal{G}$.

We now show that if the algorithm M satisfies certain properties, then for any prior on agent types, it is always an approximate Bayes Nash equilibrium for every player to opt in and follow the recommendation of the proxy.

Theorem 6. *Let M be an algorithm that satisfies (ϵ, δ)-joint differential privacy, and be such that for every vector of types $\mathbf{t} \in \mathcal{U}^n$, $M(\mathbf{t})$ induces a distribution over actions that is an α-approximate correlated equilibrium of the full information game $\mathcal{G}$ induced by the type vector $\mathbf{t}$. Then for every prior distribution on* *types τ, it is an η-approximate Bayes Nash equilibrium of $\mathcal{G}'_M$ for every player to play $(\top, f)$ for the identity function $f(a) = a$. (i.e. for every player to opt into the proxy, and then follow its suggested action), where $\eta = \epsilon + \delta + \alpha$.*

Remark 7. *Observe that when agents play according to the approximate Bayes Nash equilibrium of $\mathcal{G}'_M$ guaranteed by Theorem 6, then the resulting distribution over actions played, and the resulting utilities of the players, correspond to an α-approximate correlated equilibrium of the full information game $\mathcal{G}'_M$, induced by the realized type vector* $\mathbf{t}$.

Proof of Theorem 6. Fix any prior distribution on player types τ, and let $s_1, \ldots, s_n$ be the strategies corresponding to the action $(\top, f)$ for each player, where f is the identify function. (i.e. the strategy corresponding to opting into the proxy and following the suggested action). There are two types of deviations that a player i can consider: $(\top, f'_i)$ for some function $f'_i : A \to A$ not the identity function, and $(\bot, a_i)$ for some action a_i. First, we consider deviations of the first kind. Let $s'_i(t_i)$ be the strategy corresponding to playing $(\top, f'_{\hat{s}(t_i)})$ for some function $\hat{s}(t_i)$. For every type t_i:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathbf{t}_{-i} \sim \tau_{|t_i}} [u_i(t_i, s_i(t_i), s_{-i}(t_{-i}))] &= \sum_{\mathbf{t}_{-i}} \Pr_{\tau_{|t_i}}[\mathbf{t}_{-i}] \cdot \mathop{\mathbb{E}}_{\mathbf{a} \sim M(\mathbf{t})} [u_i(\mathbf{a})] \\
&\geq \sum_{\mathbf{t}_{-i}} \Pr_{\tau_{|t_i}}[\mathbf{t}_{-i}] \cdot \mathop{\mathbb{E}}_{\mathbf{a} \sim M(\mathbf{t})} \left[u_i(f'_{\hat{s}(t_i)}(\mathbf{a}_i), \mathbf{a}_{-i})\right] - \alpha \\
&= \mathop{\mathbb{E}}_{\mathbf{t}_{-i} \sim \tau_{|t_i}} \left[u_i(t_i, s'_i(t_i), s_{-i}(t_{-i}))\right] - \alpha
\end{aligned}$$

where the inequality follows from the fact that M computes an α-approximate correlated equilibrium. Now, consider a deviation of the second kind. Let $s'_i(t_i)$ be the strategy corresponding to playing $(\bot, a_{\hat{s}(t_i)})$ for some function $\hat{s}(t_i)$. For every type t_i:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathbf{t}_{-i} \sim \tau_{|t_i}} [u_i(t_i, s_i(t_i), s_{-i}(t_{-i}))] &= \sum_{\mathbf{t}_{-i}} \Pr_{\tau_{|t_i}}[\mathbf{t}_{-i}] \cdot \mathop{\mathbb{E}}_{\mathbf{a} \sim M(\mathbf{t})} [u_i(\mathbf{a})] \\
&\geq \sum_{\mathbf{t}_{-i}} \Pr_{\tau_{|t_i}}[\mathbf{t}_{-i}] \cdot \mathop{\mathbb{E}}_{\mathbf{a} \sim M(\mathbf{t})} \left[u_i(a_{\hat{s}(t_i)}, \mathbf{a}_{-i})\right] - \alpha \\
&\geq \sum_{\mathbf{t}_{-i}} \Pr_{\tau_{|t_i}}[\mathbf{t}_{-i}] \cdot \exp(-\epsilon) \cdot \mathop{\mathbb{E}}_{\mathbf{a} \sim M(\bot, \mathbf{t}_{-i})} \left[u_i(a_{\hat{s}(t_i)}, \mathbf{a}_{-i})\right] - \delta - \alpha \\
&\geq \sum_{\mathbf{t}_{-i}} \Pr_{\tau_{|t_i}}[\mathbf{t}_{-i}] \cdot \mathop{\mathbb{E}}_{\mathbf{a} \sim M(\bot, \mathbf{t}_{-i})} \left[u_i(a_{\hat{s}(t_i)}, \mathbf{a}_{-i})\right] - \epsilon - \delta - \alpha \\
&= \mathop{\mathbb{E}}_{\mathbf{t}_{-i} \sim \tau_{|t_i}} \left[u_i(t_i, s'_i(t_i), s_{-i}(t_{-i}))\right] - \varepsilon - \delta - \alpha
\end{aligned}$$

where the first inequality follows from the α-approximate correlated equilibrium condition, the second follows from the (ϵ, δ)-joint differential privacy condition, and the third follows from the fact that for $\epsilon \geq 0$, $\exp(-\epsilon) \geq 1 - \epsilon$ and that utilities are bounded in $[0, 1]$. □

The main technical contribution of the paper is an algorithm M which satisfies (ϵ, δ)-joint differential privacy, and computes an α-approximate correlated equilibrium of games which are γ-large. We in fact give two such algorithms: one that runs in time polynomial in n and $|A| = k$, and one that runs in time exponential in n and k. The efficient algorithm computes an α_1-approximate correlated equilibrium, and the inefficient algorithm computes an α_2-approximate correlated equilibrium, where:

$$\alpha_1 = \tilde{O}\left(\frac{\gamma k^{3/2}\sqrt{n\log(1/\delta)}}{\epsilon}\right), \quad \alpha_2 = \tilde{O}\left(\frac{\gamma \log k \log^{3/2}(\mathcal{U})\sqrt{n}\log(1/\delta)}{\epsilon}\right).$$

In combination with Theorem 6, the existence of these algorithms together with optimal choices of ϵ and δ give our main result:

Theorem 8. *Let $\mathcal{G}$ be any γ-large game. Then there exists a proxy game $\mathcal{G}'$ such that for any prior distribution on types τ, it is an η-approximate Bayes-Nash equilibrium to opt into the proxy and follow its advice. Moreover, the resulting distribution on actions forms an η-approximate correlated equilibrium of the full information game induced by the realized types. If we insist that the proxy be implemented using a computationally efficient algorithm, then we can take:*

$$\eta = \tilde{O}\left(\sqrt{\gamma}n^{1/4}k^{3/4}\right)$$

If we can take the proxy to be computationally inefficient, then we can take:

$$\eta = \tilde{O}\left(\sqrt{\gamma}n^{1/4}\sqrt{\log k}\log^{3/4}|\mathcal{U}|\right)$$

Remark 9. *In large games, the parameter γ tends to zero as n grows large. For $\gamma = 1/n$, our approximation error is $\eta = \tilde{O}(k^{3/4}/n^{1/4})$ and $\eta = \tilde{O}(\sqrt{\log k}\log^{3/4}|\mathcal{U}|/n^{1/4})$ respectively. Note that the approximation error η in the equilibrium concepts tends to zero in any γ-large game such that $\gamma = o(\frac{1}{\sqrt{n}k^{3/2}})$ or $\gamma = o(\frac{1}{\sqrt{n}\log k \log^{3/2}|\mathcal{U}|})$ respectively.*

4 No-Regret Algorithms

4.1 Definitions and Basic Properties

Here we recall some basic results on no-regret learning. See [Nis07] for a text-book exposition.

Let $\{1, 2, \ldots, k\}$ be a finite set of k *actions*. Let $L = (l_1, \ldots, l_T) \in [0,1]^{T\times k}$ be a *loss matrix* consisting of T vectors of losses for each of the k actions. Let $\Pi = \left\{\pi \in [0,1]^k \mid \sum_{j=1}^k \pi^j = 1\right\}$ be the set of distributions over the k actions and let π_U be the uniform distribution. An *online learning algorithm* $\mathcal{A}\colon \Pi \times [0,1]^k \to \Pi$ takes a distribution over k actions and a vector of k losses, and produces a new distribution over the k actions. We use $\mathcal{A}_t(L)$ to denote the distribution produced by running $\mathcal{A}$ sequentially $t-1$ times using the loss vectors $l_1, \ldots, l_{t-1}$, and then running $\mathcal{A}$ on the resulting distribution and the loss vector l_t. That is:

$$\mathcal{A}_0(L) = \pi_U,$$
$$\mathcal{A}_t(L) = \mathcal{A}(\mathcal{A}_{t-1}(L), l_t).$$

We use $\mathcal{A}(L) = (\mathcal{A}_0(L), \mathcal{A}_1(L), \ldots, \mathcal{A}_T(L))$ when T is clear from context.

Let $\pi_0, \ldots, \pi_T \in \Pi$ be a sequence of T distributions and let L be a T-row loss matrix. We define the quantities:

$$\lambda(\pi, l) = \sum_{j=1}^{k} \pi^j l^j,$$
$$\lambda(\pi_0, \ldots, \pi_T, L) = \frac{1}{T}\sum_{t=1}^{T} \lambda(\pi_t, l_t),$$
$$\lambda(\mathcal{A}(L'), L) = \lambda(\mathcal{A}_0(L'), \mathcal{A}_1(L'), \ldots, \mathcal{A}_T(L'), L).$$

Note that the notation retains the flexibility to run the algorithm $\mathcal{A}$ on one loss matrix, but measure the loss $\mathcal{A}$ incurs on a different loss matrix. This flexibility will be useful later.

Let $\mathcal{F}$ be a family of functions $f\colon \{1, 2, \ldots, k\} \to \{1, 2, \ldots, k\}$. For a function f and a distribution π, we define the distribution $f \circ \pi$ to be

$$(f \circ \pi)^j = \sum_{j' : f(j') = j} \pi^{j'}.$$

The distribution $f \circ \pi$ corresponds to the distribution on actions obtained by first choosing an action according to π, then applying the function f.

Now we define the following quantities:

$$\lambda(\pi_1, \ldots, \pi_T, L, f) = \lambda(f \circ \pi_1, f \circ \pi_2, \ldots, f \circ \pi_T, L),$$
$$\rho(\mathcal{A}, L, f) = \lambda(\mathcal{A}, L) - \lambda(\mathcal{A}, L, f),$$
$$\rho(\mathcal{A}, L, \mathcal{F}) = \max_{f \in \mathcal{F}} \rho(\mathcal{A}, L, f).$$

As a mnemonic, we offer the following. λ refers to expected loss, ρ refers to regret. Next, we define the families $\mathcal{F}_{\mathsf{fixed}}, \mathcal{F}_{\mathsf{swap}}$:

$$\mathcal{F}_{\mathsf{fixed}} = \left\{ f_j(j') = j, \text{ for all } j' \mid j \in \{1, 2, \ldots, k\} \right\}$$
$$\mathcal{F}_{\mathsf{swap}} = \{ f : \{1, 2, \ldots, k\} \to \{1, 2, \ldots, k\} \}$$

Looking ahead, we will need to be able to handle not just *a priori fixed* sequences of losses, but also adapted. To see why, note that for a game setting, a player's loss will depend on the distribution of actions played by everyone in that period, which will depend, in turn, on the losses everyone experienced in the previous period and how everyone's algorithms reacted to that.

Definition 10 (Adapted Loss)**.** A loss function $\mathcal{L}$ is said to be adapted to an algorithm $\mathcal{A}$ if in each period t, the experienced losses $l_t \in [0, 1]^k$ can be written as:

$$l_t = \mathcal{L}(l_0, \mathcal{A}(l_0), l_1, \mathcal{A}(l_1), \ldots, l_{t-1}, \mathcal{A}(l_{t-1})).$$

The following well-known result shows the existence of algorithms that guarantee low regret even against adapted losses (see e.g. [Nis07]).

Theorem 11. *There exists an algorithm $\mathcal{A}_{\mathsf{fixed}}$ such that for any adapted loss $\mathcal{L}$, $\rho(\mathcal{A}_{\mathsf{fixed}}, \mathcal{L}, \mathcal{F}_{\mathsf{fixed}}) \le \sqrt{\frac{2 \log k}{T}}$. There also exists an algorithm $\mathcal{A}_{\mathsf{swap}}$ such that $\rho(\mathcal{A}_{\mathsf{swap}}, \mathcal{L}, \mathcal{F}_{\mathsf{swap}}) \le k\sqrt{\frac{2 \log k}{T}}$.*

4.2 Noise Tolerance of No-Regret Algorithms

The next lemma states that when a no-regret algorithm is run on a noisy sequence of losses, it does not incur too much additional regret with respect to the real losses.

Lemma 12 (Regret Bounds for Bounded Noise)**.** *Let $L \in [\frac{1}{3}, \frac{2}{3}]^{T\times k}$ be any loss matrix. Let $Z = (z_t^j) \in [-b, b]^{T\times k}$ be an arbitrary matrix with bounded entries, and let $\widehat{L} = L + Z$. Let $\mathcal{A}$ be an algorithm. Let $\mathcal{F}$ be any family of functions. Then*

$$\rho(\mathcal{A}(\widehat{L}), L, \mathcal{F}) \leq \rho(\mathcal{A}(\widehat{L}), \widehat{L}, \mathcal{F}) + 2b.$$

Corollary 13. *Let $L \in [\frac{1}{3}, \frac{2}{3}]^{T\times k}$ be any loss matrix and let $Z \in \mathbb{R}^{T\times k}$ be a random matrix such that $\mathbb{P}_Z\left[Z \in [-b, b]^{T\times k}\right] \geq 1 - \beta$ for some $b \in [0, \frac{1}{3}]$, and let $\widehat{L} = L + Z$. Then*

1. $\mathbb{P}_Z\left[\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), L, \mathcal{F}_{\mathsf{fixed}}) > \sqrt{\frac{2\log k}{T}} + 2b\right] \leq \beta,$

2. $\mathbb{P}_Z\left[\rho(\mathcal{A}_{\mathsf{swap}}(\widehat{L}), L, \mathcal{F}_{\mathsf{swap}}) > k\sqrt{\frac{2\log k}{T}} + 2b\right] \leq \beta.$

Note that the technical conditioned $b \in [0, \frac{1}{3}]$ is needed to ensure that the noisy loss matrix $\widehat{L}$ is contained in $[0, 1]^{T\times k}$, which is required to apply the regret bounds of Theorem 11.

We prove a tighter bound on the additional regret in the case where the entries of Z are iid samples from a Laplace distribution.

Lemma 14 (Regret Bounds for Laplace Noise)**.** *Let $L \in [\frac{1}{3}, \frac{2}{3}]^{T\times k}$ be any loss matrix. Let $Z = (z_t^j) \in \mathbb{R}^{T\times k}$ be a random matrix formed by taking each entry to be an independent sample from $\mathrm{Lap}(\sigma)$, and let $\widehat{L} = L + Z$. Let $\mathcal{A}$ be an algorithm. Let $\mathcal{F}$ be any family of functions. Then for any $\eta \leq \sigma$.*

$$\mathop{\mathbb{P}}_{Z}\left[\rho(\mathcal{A}(\widehat{L}), L, \mathcal{F}) - \rho(\mathcal{A}(\widehat{L}), \widehat{L}, \mathcal{F}) > \eta\right] \leq 2|\mathcal{F}|e^{-\eta^2 T/24\sigma^2}.$$

Corollary 15. *Let $L \in [\frac{1}{3}, \frac{2}{3}]^{T\times k}$ be any loss matrix and let $Z \in \mathbb{R}^{T\times k}$ be a random matrix formed by taking each entry to be an independent sample from $\mathrm{Lap}(\sigma)$ for $\sigma < \frac{1}{6\log(4KT/\beta)}$ and let $\widehat{L} = L + Z$. Then*

1. $\mathbb{P}_Z\left[\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), L, \mathcal{F}_{\mathsf{fixed}}) > \sqrt{\frac{2\log k}{T}} + \sigma\sqrt{\frac{24\log(4k/\beta)}{T}}\right] \leq \beta,$

2. $\mathbb{P}_Z\left[\rho(\mathcal{A}_{\mathsf{swap}}(\widehat{L}), L, \mathcal{F}_{\mathsf{swap}}) > k\sqrt{\frac{2\log k}{T}} + \sigma\sqrt{\frac{24k\log(4k/\beta)}{T}}\right] \leq \beta.$

As before, the technical condition upper bounding σ is to ensure that the noisy loss matrix $\widehat{L}$ is contained in $[0, 1]^{T\times k}$, so that the regret bounds of $\mathcal{A}$ apply.

4.3 From No Regret to Equilibrium

Let $(u_1, \ldots, u_n)$ be utility functions for each of n players. Let $S = \{(\pi_{i,1}, \ldots, \pi_{i,T})\}_{i=1}^n$ be a collection of n sequences of distributions over k actions, one for each player. Let $\{(l_{i,1}, \ldots, l_{i,T})\}_{i=1}^n$ be a collection of n sequences of loss vectors $l \in [0,1]^k$ formed by the action distribution. More formally, for every j, $l_{i,t}^j = 1 - \mathbb{E}_{\pi_{-i,t}}[u_i(j, a_{-i})]$. Define the maximum regret that any player has to her losses

$$\rho_{\mathsf{max}}(S, L, \mathcal{F}) = \max_i \rho(S_i, L_i, \mathcal{F})$$

where $S_i = (\pi_{i,0}, \ldots, \pi_{i,T})$ and $L_i = (l_{i,1}, \ldots, l_{i,T})$.

Given the collection S, we define the correlated action distribution Π_S be the average distribution of play. That is, Π_S is the distribution over A^n defined by the following sampling procedure: Choose t uniformly at random from $\{1, 2, \ldots, T\}$, then, for each player i, choose a_i randomly according to the distribution $\pi_{i,t}$, independently of the other players.

The following well known theorem (see, e.g. [Nis07]) relates low-regret sequences of play to the equilibrium concepts (Definition 1):

Theorem 16. *If the maximum regret with respect to $\mathcal{F}_{\mathsf{fixed}}$ is small, i.e. $\rho_{\mathsf{max}}(S, L, \mathcal{F}_{\mathsf{fixed}}) \leq \alpha$, then the correlated action distribution Π_S is an α-approximate coarse correlated equilibrium. Similarly, if $\rho_{\mathsf{max}}(S, L, \mathcal{F}_{\mathsf{swap}}) \leq \alpha$, then Π_S is an α-approximate correlated equilibrium.*

In this section we show that no-regret algorithms are noise-tolerant, that is we still get good regret bounds with respect to the real losses if we run a no-regret algorithm on noisy losses (real losses plus low-magnitude noise).

Let $L \in [0,1]^{T \times k}$ be a loss matrix. Define $\overline{L} = \frac{L+1}{3}$ (entrywise) and note that $\overline{L} \in [\frac{1}{3}, \frac{2}{3}]^{T \times k}$. The following lemma states that running $\mathcal{A}$ on $\overline{L}$ doesn't significantly increase the regret with respect to the real losses.

Lemma 17. *For every algorithm $\mathcal{A}$, every family $\mathcal{F}$, and every loss matrix $L \in [0,1]^{T \times k}$,*

$$\rho(\mathcal{A}(\overline{L}), L, \mathcal{F}) \leq 3\rho(\mathcal{A}(\overline{L}), \overline{L}, \mathcal{F}).$$

In particular, for every $L \in [0,1]^{T \times k}$

$$\rho(\mathcal{A}_{\mathsf{fixed}}(\overline{L}), L, \mathcal{F}_{\mathsf{fixed}}) \leq \sqrt{\frac{18 \log k}{T}} \quad \text{and} \quad \rho(\mathcal{A}_{\mathsf{swap}}(\overline{L}), L, \mathcal{F}_{\mathsf{swap}}) \leq k\sqrt{\frac{18 \log k}{T}}.$$

In light of Lemma 17, for the rest of this section we will take L to be a loss matrix in $[\frac{1}{3}, \frac{2}{3}]^{T \times k}$. This rescaling will only incur an additional factor of 3 in the regret bounds we prove. Let $Z \in \mathbb{R}^{T \times k}$ be a real valued *noise matrix*. Let $\widehat{L} = L + Z$ (entrywise). In the next section we consider the case where Z is an arbitrary matrix with bounded entries. We prove a tighter bound for the case where Z consists of independent draws from a Laplace distribution.

5 Private Equilibrium Computation

Having demonstrated the noise tolerance of no-regret algorithms, we now argue that for appropriately chosen noise, the output of the algorithm constitutes a jointly-differentially private mechanism (Definition 5). We prove two results of this type. First, in Section 5.2 we consider games with 'few' actions per player. While the algorithm is conceptually more straightforward, it is not useful in certain cases of interest. For example, in the routing games we described in the introduction, the set of actions available to a player consists of all routes between her starting point and her destination. Even if the graph (road network) is small, the number of feasible routes can be extremely large (exponential in the number of edges (roads)). However, in such games, the set of types (utility functions) is small (i.e. the set of all source-destination pairs). Motivated by this observation, in Section 5.3 we consider games with large action spaces, but bounded type spaces.

5.1 Privacy Preliminaries

Before presenting our algorithms for computing correlated equilibria under joint differential privacy, we still state some useful tools for achieving differential privacy. An important result we will use is that differentially private mechanisms 'compose' nicely.

Theorem 18 (Adaptive Composition [DRV10]). *Let $\mathcal{A}\colon \mathcal{U} \to \mathcal{R}^T$ be a T-fold adaptive composition*[5] *of (ε, δ)-differentially private mechanisms. Then $\mathcal{A}$ satisfies $(\varepsilon', T\delta + \delta')$-differential privacy for*

$$\varepsilon' = \varepsilon\sqrt{2T\ln(1/\delta')} + T\varepsilon(e^{\varepsilon} - 1).$$

In particular, for any $\varepsilon \le 1$, if $\mathcal{A}$ is a T-fold adaptive composition of $(\varepsilon/\sqrt{8T\ln(1/\delta)}, 0)$-differentially privacy mechanisms, then $\mathcal{A}$ satisfies (ε, δ)-differential privacy.

Finally, differentially private mechanisms often involve adding Laplacian random noise. We will denote a (mean 0) and scale σ Laplacian random variable by $\mathrm{Lap}(\sigma)$. The following foundational result shows that adding Laplacian noise to a insensitive function makes it differentially private.

Theorem 19 (Privacy of the Laplace Mechanism [DMNS06]). *Let $Q\colon \mathcal{U} \to \mathbb{R}$ be any γ-sensitive function. Define the mechanism $\mathcal{M}(\mathbf{u}) = Q(\mathbf{u}) + \mathrm{Lap}(\sigma)$. If $\sigma = \gamma/\varepsilon$, then $\mathcal{M}$ is $(\varepsilon, 0)$-differentially private.*

The following concentration inequality for Laplacian random variables will be useful.

Theorem 20 ([GRU12]). *Suppose $\{Y_i\}_{i=1}^T$ are i.i.d. $\mathrm{Lap}(\sigma)$ random variables, and scalars $q_i \in [0,1]$. Define $Y := \frac{1}{T}\sum_i q_i Y_i$. Then for any $\alpha \le \sigma$,*

$$\Pr[Y \ge \alpha] \le \exp\left(-\frac{\alpha^2 T}{6\sigma^2}\right).$$

5.2 Games with Few Actions

To orient the reader at a high-level, our proof has two main steps. First, we construct a 'wrapper' NRLaplace$^{\mathcal{A}}$ that will ensure privacy. The wrapper takes as input the parameters of the game, the reported tuple of utilities, and any no-regret algorithm $\mathcal{A}$. This wrapper will attempt to compute an equilibrium using the method outlined in Section 4.3—for T periods, it will compute a mixed strategy for each player by running $\mathcal{A}$ on the previous period's losses. In order to ensure privacy, instead of using the true losses as input to $\mathcal{A}$, it will use losses perturbed by suitably chosen Laplace noise. After running for T periods, the wrapper will output to each player the sequence of T mixed strategies computed for that player. In Theorem 21 we show that this constitutes a jointly differentially private mechanism. Then, in Theorem 22, we show that the output of this wrapper converges to an approximate correlated equilibrium when the input algorithm is the no-swap-regret algorithm $\mathcal{A}_{\mathsf{swap}}$.

5.2.1 Noisy No-Regret Algorithms are Differentially Private

Theorem 21 (Privacy of NRLaplace$^{\mathcal{A}}$). *For any $\mathcal{A}$, the algorithm NRLaplace$^{\mathcal{A}}$ satisfies (ε, δ)-joint differential privacy.*

[5] See [DRV10] for further discussion

NRLAPLACE$^{\mathcal{A}}(u_1, \dots u_n)$
PARAMS: $\varepsilon, \delta, \gamma \in (0, 1], n, k, T \in \mathbb{N}$
LET: $\pi_{1,1}, \dots, \pi_{n,1}$ each be the uniform distribution over $\{1, 2, \dots, k\}$.
LET: $\sigma = \varepsilon^{-1}\gamma\sqrt{8nkT\ln(1/\delta)}$
FOR: $t = 1, 2, \dots, T$
LET: $l_{i,t}^{j} = 1 - \mathbb{E}_{\pi_{-i,t}}\left[u_i(j, a_{-i})\right]$ for every player i, action j.
LET: $z_{i,t}^{j}$ be an i.i.d. draw from $\mathrm{Lap}(\sigma)$ for every player i, action j.
LET: $\widehat{l}_{i,t}^{j} = l_{i,t}^{j} + z_{i,t}^{j}$ for every player i, action j.
LET: $\pi_{i,t+1} = \mathcal{A}(\pi_{i,t}, \widehat{l}_{i,t})$ for every player i.
END FOR
OUTPUT: $(\pi_{i,1}, \dots, \pi_{i,T})$ to player i, for every i.

We now sketch the proof. Fix any player i and any utility functions u_{-i}. We argue that the output to all other players is differentially private as a function of u_i. It will be easier to analyze a modified mechanism that outputs the noisy losses $(\widehat{l}_{-i,1}, \dots, \widehat{l}_{-i,T})$, rather than the mixed strategies $(\pi_{-i,1}, \dots, \pi_{-i,T})$. Since the noisy losses are sufficient to compute $(\pi_{-i,1}, \dots, \pi_{-i,T})$, proving that the noisy losses are jointly differentially private is sufficient to prove that the mixed strategies are a well.

To get intuition for the proof, first consider the first period of noisy losses $\widehat{l}_{-i,1}$. For each player $i' \neq i$, and each action $j \in [k]$, the loss $l_{i',j}^{1}$ depends on $\pi_{i,1}$, which is independent of the utility of player i. Thus, in the first round there is no loss of privacy. In the second round, the loss $l_{i',j}^{2}$ depends on $\pi_{i,2}$, which depends on the losses for player i in period 1, and thus depends on the utility of player i. The loss $l_{i',j}^{2}$ also depends on the mixed strategies $\pi_{i'',2}$ for players $i'' \neq i, i'$, but as we've argued these mixed strategies are independent of u_i. We will take a pessimistic view and assume that changing player i's utility function from u_i to u_i' will change $\pi_{i,2}$ *arbitrarily*. The assumption that $u_{i'}$ is only γ-sensitive to the action of player i, ensures that the expected losses of player i', $l_{i',j}^{2}$ only change by at most γ. Thus, by Theorem 19 and our choice of the noise parameter σ, each noisy loss $\widehat{l}_{i',j}^{2}$ will be $\varepsilon/\sqrt{8nkT\ln(1/\delta)}$ differentially private as a function of u_i.

Understanding the third round will be sufficient to argue the general case. Just as in period 2, the loss $l_{i',j}^{3}$ depends on $\pi_{i,3}$. However, the $l_{i',j}^{3}$ also depends on $\pi_{i'',3}$ for players $i'' \neq i, i'$ and now these strategies do indeed depend on u_i, as we saw when reasoning about period 2. However, the key observation is that $\pi_{i'',3}$ depends on the utility of player i only through the noisy losses $l_{i'',j}^{2}$ that we computed in the previous round. Since we already argued that these losses are differentially private as a function of u_i, it will not compromise privacy further to use these noisy losses when computing $l_{i',j}^{3}$. Thus, conditioned on the noisy losses output in periods 1 and 2, the losses $l_{i',j}^{3}$ depend only on the mixed strategy of player i in period 3. As we argued before, the amount of noise we add to these losses will be sufficient to ensure $\varepsilon/\sqrt{8nkT\ln(1/\delta)}$ differential privacy as a function of u_i.

In summary, we have shown that for every period t, every player $i' \neq i$, and every action j, the noisy loss $\widehat{l}_{i',j}^{t}$ is an $\varepsilon/\sqrt{8nkT\ln(1/\delta)}$-differentially private function of u_i and of the previous $t-1$ periods' noisy losses, which are themselves already differentially private. In total we compute $T(n-1)k$ noisy losses. Hence, the adaptive composition theorem (Theorem 18) ensures that the entire sequence of noisy losses $\widehat{l}_{-i}^{1}, \dots, \widehat{l}_{-i}^{T}$ is ε-differentially private as a function of u_i. Since this analysis holds for every player i, and shows that the output to all of the remaining players is ε-differentially private as a function of u_i, the entire mechanism is ε-jointly differentially private.

5.2.2 Noisy No-Regret Algorithms Compute Approximate Equilibria

Therefore we have shown how that the this 'wrapper' algorithm is jointly differentially private in the sense of Definition 5. We now proceed to show that using this algorithm with $\mathcal{A}_{\mathsf{swap}}$ will result in an approximate correlated equilibrium (Corollary 22).

Theorem 22 (Computing CE)**.** *Let $\mathcal{A} = \mathcal{A}_{\mathsf{swap}}$. Fix the environment, i.e. the number of players n, the number of actions k, the sensitivity of the game γ, and the degree of privacy desired, (ϵ, δ). One can then select the number of rounds the algorithm must run T, and two numbers α, β satisfying:*

$$\gamma\varepsilon^{-1}\sqrt{8nkT\log(1/\delta)} \leq \frac{1}{6\log(4nkT/\beta)}, \tag{2}$$

such that probability at least $1-\beta$, the algorithm NRLAPLACE$^{\mathcal{A}_{\mathsf{swap}}}$*, returns an α-approximate correlated equilibrium for:*[6]

$$\alpha = \tilde{O}\left(\frac{\gamma k^{3/2}\sqrt{n\log(1/\delta)}\log(1/\beta)}{\varepsilon}\right)$$

Before we proceed to the proof, some discussion is appropriate. It is already well known that no-regret alogrithms converge 'quickly' to approximate equilibria– recall Theorems 11 and 16. In the previous section, we showed that adding noise still leads to low regret (and therefore to approximate equilibrium). The tradeoff therefore is this. To get a more 'exact' equilibrium, the algorithm has to be run for more rounds. By the arguments in Theorem 21, this will result in a less private outcome. The current theorem makes precise the tradeoff between the two.

This is a *strongly positive* result—in several large games of interest, e.g. anonymous matching games, $\gamma = O(n^{-1})$. Therefore, for games of this sort $\alpha = \tilde{O}(\sqrt{k/n})$. If k is fixed, but n is large, therefore, a relatively exact equilibrium of the underlying game can be implemented, while still being jointly differentially private to the desired degree.

Proof of Theorem 22. By our choice of the parameter σ, in the algorithm NRLAPLACE$^{\mathcal{A}_{\mathsf{swap}}}$, which is

$$\sigma = \gamma\varepsilon^{-1}\sqrt{8nkT\log(1/\delta)},$$

and by assumption of the theorem, (2), we have $\sigma \leq 1/6\log(4nkT/\beta)$. Applying Theorem 15 we obtain:

$$\mathbb{P}_Z\left[\rho(\pi_{i,1},\ldots,\pi_{i,T},L_i,\mathcal{F}_{\mathsf{swap}}) > \sqrt{\frac{2\log k}{T}} + \sigma\sqrt{\frac{24k\log(4nk/\beta)}{T}}\right] \leq \frac{\beta}{n}$$

for any player i, where L_i is the loss matrix derived from the given utility functions u_i and the distributions $\{\pi_{i,t}\}_{i\in[n],t\in[T]}$. Now we can take a union bound over all players i, yielding:

$$\mathbb{P}_Z\left[\max_i \rho(\pi_{i,1},\ldots,\pi_{i,T},L_i,\mathcal{F}_{\mathsf{swap}}) > \sqrt{\frac{2\log k}{T}} + \sigma\sqrt{\frac{24k\log(4nk/\beta)}{T}}\right] \leq \beta,$$

$$\Longrightarrow \mathbb{P}_Z\left[\rho_{\mathsf{max}}(\pi,L,\mathcal{F}_{\mathsf{swap}}) > \sqrt{\frac{2\log k}{T}} + \sigma\sqrt{\frac{24k\log(4nk/\beta)}{T}}\right] \leq \beta.$$

[6] Here, $\tilde{O}$ hides lower order $\mathrm{poly}(\log N, \log K, \log T, \log(1/\Delta), \log(1/\varepsilon), \log\log(1/\beta), \log\log(1/\delta))$ factors.

By Theorem 16, therefore, the empirical distribution of play is a $\left(\sqrt{\frac{2\log k}{T}} + \sigma\sqrt{\frac{24k\log(4nk/\beta)}{T}}\right)$-approximate correlated equlibrium.

To finish, substitute $\sigma = \gamma\varepsilon^{-1}\sqrt{8nkT\log(1/\delta)}$ into the expression above. Therefore, with probability at least $1-\beta$, no player has regret larger than

$$\alpha = \sqrt{\frac{2\log k}{T}} + \frac{\gamma k\sqrt{192n\log(1/\delta)\log(4nk/\beta)}}{\varepsilon}$$

Since T is a parameter of the algorithm, we can choose T to minimize α. Since α is monotonically decreasing in T, we would like to choose T as large as possible. However, our argument requires (2), which (roughly) requires $\sqrt{T} \lesssim 1/\gamma\sqrt{nk}$, where we have suppressed dependence on some of the parameters. By choosing T so that $\sqrt{T} \sim 1/\gamma\sqrt{nk}$ we can make the first term of the error $\sim \gamma\sqrt{nk}$, which would make it be of a smaller order to the second term. It is easy to verify that we can choose T is such a way that T satisfies the assumption and the resulting value of α satisfies the conclusion of the theorem. □

5.3 Upper bounds for Games with Bounded Type Spaces

In the previous section, we showed that a private equilibrium can be computed with a $O(\sqrt{k/n})$ approximate equilibrium. While this result is positive for some settings (e.g. anonymous matching games for large populations), it has no bite in settings where the number of actions is as large (or larger) than the number of players. The reason is that, with large numbers of actions, the no-regret algorithm will need information about the losses incurred by 'many' different actions. Giving the no-regret algorithms this information requires that we either sacrifice privacy, or introduce a lot of noise to ensure privacy, which would make the computed equilibrium a meaningless approximation.

In order to get a better bound on the accuracy as a function of the number of queries, we will need a mechanism that is capable of answering a large number of queries accurately. One such mechanism is the so-called Median Mechanism of Roth and Roughgarden [RR10], paired with the privacy analysis of Hardt and Rothblum [HR10].[7]

Roughly, the Median Mechanism allows us to take a tuple $\mathbf{u} = (u_1, \ldots, u_n) \in U$ and answer any collection of Q (adaptively chosen) γ-sensitive queries about $\mathbf{u}$ while 1) satisfying (ε, δ)-differential privacy and 2) answer each query accurately to within error $\alpha_{\mathcal{M}_M} = O(\gamma\sqrt{n\log U}\log Q)$.[8] The relevant comparison here is to the use of Laplace noise, which would introduce error roughly $\alpha_{\text{Lap}} = O(\gamma\sqrt{Q})$. Thus, when Q is much larger than n, and U is not too large, the Median Mechanism answers queries with much greater accuracy than adding Laplace noise.

Intuitively, our mechanism for computing equilibria in large games follows the same blueprint as in the previous section, but uses noisy losses generated by the Median Mechanism rather than noisy losses generated by the addition of Laplacian noise. However, there are some subtleties that arise from the fact that the Median Mechanism will add *correlated* noise to the different losses, whereas the Laplacian noise was generated *independently* for each loss. The challenge is that, when a particular player i uses the Median Mechanism to generate the noise, the queries she makes depend on her type. Thus, when a player $i' \neq i$ uses the Median mechanism, the noise used may itself depend on player i's type, which was not the case

[7] Originally, the median mechanism of [RR10] was only defined and analyzed for the case of linear queries. A 'folk' result, first observed by Hardt and Rothblum [Har] is that the Median Mechanism (when instantiated with a net of all possible size n datasets) can be applied to arbitrary γ-sensitive queries, which immediately yields Theorem 26 when paired with the privacy analysis of [HR10]. The simple proof can be found in [DR13].

[8] We have suppressed the dependence on some parameters for this informal discussion.

previously. We resolve this issue essentially by having each player generate losses for each of the U possible types, regardless of their own type, and then discard the answers not corresponding to that player's actual type. This modification ensures that the queries made to the Median Mechanism never depend explicitly on the types of the players. Although this modification increases the number of queries we need to make by a factor of U, since the Median Mechanism has error that depends only logarithmically on the number of queries, we are still able to handle very large type spaces.

We defer a formal treatment to the appendix, and simply state our results for this section here.

Theorem 23 (Computing CE)**.** *Fix the environment, i.e the number of players n, the number of actions k, number of possible utility functions U, sensitivity of the game γ, the desired privacy (ϵ, δ), and the failure probability β. There exists an algorithm such that with probability at least $1 - \beta$, the algorithm returns an α-approximate CE for:*[9]

$$\alpha = \tilde{O}\left(\frac{\gamma\sqrt{n}\log^{3/2} U \log(k/\beta)\log(1/\delta)}{\varepsilon}\right)$$

5.4 A Lower Bound

In the case where $\gamma = O(1/n)$ and $k = O(1)$, both of our algorithms from the previous Section compute a differentially private, α-approximate equilibrium for $\alpha \sim 1/\sqrt{n}$ (ignoring all other parameters). It is natural to ask whether or not we can achieve significantly smaller values of α using some other algorithm. In this section we prove a lower bound showing that this is not the case. Specifically, we show that there is no algorithm that privately computes an α-approximate equilibrium of an arbitrary n-player 2-action game, for $\alpha \ll 1/\sqrt{n \log n}$. In other words, there cannot exist an algorithm that privately computes a 'signficantly' more exact equilibrium.

Our proof is by a reduction to the problem of differentially private *subset-sum query release*, for which strong information theoretic lower bounds are known [DN03, DY08]. The problem is as follows: Consider a database $D \in \{0,1\}^n$, which we denote $(d_1, \dots, d_n)$. A subset-sum query $q \subseteq [n]$ is defined by a subset of the n database entries and asks "What fraction of the entries in D are contained in q and are set to 1?" Formally, we define the query q as $q(D) = \frac{1}{n}\sum_{i \in q} d_i$. Given a set of subset-sum queries $\mathcal{Q} = \{q_1, \dots, q_m\}$, we say that an algorithm $\mathcal{M}(D)$ *releases* $\mathcal{Q}$ *to accuracy* α if $\mathcal{M}(D) = (a_1, \dots, a_m)$ such that $|a_j - q_j(D)| \leq \alpha$ for every $j \in [m]$.

We show that an algorithm for computing approximate equilibrium in arbitrary games could also be used to release arbitrary sets of subset-sum queries accurately. The following theorem shows that a differentially private mechanism to compute approxmiate equilibrium implies a differentially private algorithm to compute subset-sums.

Theorem 24. *For any $\alpha > 0$, if there is an (ε, δ)-jointly differentially private mechanism $\mathcal{M}$ that computes an α-approximate coarse correlated equilibria in $(n + m\log n)$-player, 2-action, $1/n$-sensitive games, then there is an (ε, δ)-differentially private mechanism $\mathcal{M}'$ that releases 36α-approximate answers to any m subset-sum queries on a database of size n.*

Applying the results of [DY08], a lower bound on equilibrium computation follows easily.

Corollary 25. *Any $(\varepsilon = O(1), \delta = o(1))$-differentially private mechanism $\mathcal{M}$ that computes an α-approximate coarse correlated equilibria in n-player 2-action games with $O(1/n)$-sensitive utility functions must satisfy $\alpha = \Omega(\frac{1}{\sqrt{n \log n}})$.*

[9] $\tilde{O}$ hides lower order $\mathrm{poly}(\log n, \log\log k, \log T, \log\log U \log(1/\gamma), \log(1/\varepsilon), \log\log(1/\beta), \log\log(1/\delta))$ terms.

Here, we provide a sketch of the proof of Theorem 24. Let $D \in \{0,1\}^n$ be an n-bit database and $\mathcal{Q} = \{q_1, \ldots, q_m\}$ be a set of m subset-sum queries. For the sketch, assume that we have an algorithm that computes exact equilibria. We will split the $(n+m)$ players into n "data players" and m "query players." Roughly speaking, the data players will have utility functions that force them to play "0" or "1", so that their actions actually represent the database D. Each of the query players will represent a subset-sum query q, and we will try to set up their utility function in such a way that it forces them to take an action that corresponds to an approximate answer to $q(D)$. In order to do this, first assume there are $n+1$ possible actions, denoted $\left\{0, \frac{1}{n}, \frac{2}{n}, \ldots, 1\right\}$. We can set up the utility function so that for each action a, he receives a payoff that is maximized when an a fraction of the data players in q are playing 1. That is, when playing action a, his payoff is maximized when $q(D) = a$. Conversely, he will play the action a that is closest to the true answer $q(D)$. Thus, we can read off the answer to q from his equilibrium action. Using each of the m query players to answer a different query, we can compute answers to m queries. Finally, notice that joint differential privacy says that all of the actions of the query players will satisfy (standard) differential privacy with respect to the inputs of the data players, thus the answers we read off will be differentially private (in the standard sense) with respect to the database.

This sketch does not address two important issues. The first is that we do not assume that the algorithm computes an exact equilibrium, only that it computes an approximate equilibrium. This relaxation means that the data players do not have to play the correct bit with probability 1, and the query players do not have to choose the answer that exactly maximizes their utility. In the proof we show that the error in the answers we read off is only a small factor larger than the error in the equilibrium computed.

The second is that we do not want to assume that the (query) players have $n+1$ available actions. Instead, we use $\log n$ players per query, and use each to compute roughly one bit of the answer, rather than the whole answer. However, if the query players' utility actually depends on a specific bit of the answer, then a single data player changing his action might result in a large change in utility. In the proof, we show how to compute bits of the answer using $1/n$-sensitive utility functions.

6 Discussion

In this work, we have introduced a new variant of differential privacy (joint differential privacy), and have shown how it can be used as a tool to construct extremely weak proxy mechanisms which can implement equilibria of full information games, even when the game is being played in a setting of only partial information. Moreover, our privacy solution concept maintains the property that no coalition of players can learn (much) more about any player's type outside of the coalition than they could have learned in the original Bayesian game, and thus players have almost no incentive not to participate even if they view their type as sensitive information. Although our proxies are weak in most respects (they cannot enforce actions, they cannot make payments or charge fees, they cannot compel participation), we do make the assumption that player types are verifiable in the event that they choose to opt into the proxy. This assumption is reasonable in many settings: for example, in financial markets, there may be legal penalties for a firm misrepresenting relevant facts about itself, and in traffic routing games, the proxy may be embodied as a physical device (e.g. a GPS device) that can itself verify player types (e.g. physical location). Nevertheless, we view relaxing this assumption as an important direction for future work.

References

[AB11] E. Azevedo and E. Budish. Strategyproofness in the large as a desideratum for market design. Technical report, University of Chicago, 2011.

[ANS00] N.I. Al-Najjar and R. Smorodinsky. Pivotal players and the characterization of influence. *Journal of Economic Theory*, 92(2):318–342, 2000.

[BLR08] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. In Cynthia Dwork, editor, *STOC*, pages 609–618. ACM, 2008.

[CCK+13] Yiling Chen, Stephen Chong, Ian A Kash, Tal Moran, and Salil Vadhan. Truthful mechanisms for agents that value privacy. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pages 215–232. ACM, 2013.

[De12] Anindya De. Lower bounds in differential privacy. In *TCC*, pages 321–338, 2012.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503, 2006.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC '06*, pages 265–284, 2006.

[DMT07] Cynthia Dwork, Frank McSherry, and Kunal Talwar. The price of privacy and the limits of lp decoding. In *STOC*, pages 85–94, 2007.

[DN03] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *PODS*, pages 202–210, 2003.

[DNV12] Cynthia Dwork, Moni Naor, and Salil Vadhan. The privacy of the analyst and the power of the state. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 400–409. IEEE, 2012.

[DR13] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*. 2013.

[DRV10] Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60. IEEE Computer Society, 2010.

[Dwo08] C. Dwork. Differential privacy: A survey of results. *Theory and Applications of Models of Computation*, pages 1–19, 2008.

[DY08] Cynthia Dwork and Sergey Yekhanin. New efficient attacks on statistical disclosure control mechanisms. In *CRYPTO*, pages 469–480, 2008.

[FL12] Lisa Fleischer and Yu-Han Lyu. Approximately optimal auctions for selling privacy when costs are correlated with data. In *EC*, pages 568–585, 2012.

[For86] Françoise Forges. An approach to communication equilibria. *Econometrica: Journal of the Econometric Society*, pages 1375–1385, 1986.

[GHRU11] Anupam Gupta, Moritz Hardt, Aaron Roth, and Jonathan Ullman. Privately releasing conjunctions and the statistical query barrier. In *STOC '11*, pages 803–812, 2011.

[GR08] R. Gradwohl and O. Reingold. Fault tolerance in large games. In *EC*, pages 274–283. ACM, 2008.

[GR10] R. Gradwohl and O. Reingold. Partial exposure in large games. *Games and Economic Behavior*, 68(2):602–613, 2010.

[GR11] Arpita Ghosh and Aaron Roth. Selling privacy at auction. In *EC*, pages 199–208, 2011.

[Gra12] R. Gradwohl. Privacy in implementation. 2012.

[GRU12] Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. In *TCC*, pages 339–356, 2012.

[Har] Moritz Hardt. Personal communication.

[HK12] Zhiyi Huang and Sampath Kannan. The exponential mechanism for social welfare: Private, truthful, and nearly optimal. In *FOCS*, 2012.

[HR10] Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *FOCS*, pages 61–70, 2010.

[HRU13] Justin Hsu, Aaron Roth, and Jonathan Ullman. Differential privacy for the analyst via private equilibrium computation. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 341–350. ACM, 2013.

[IM05] N. Immorlica and M. Mahdian. Marriage, honesty, and stability. In *SODA*, pages 53–62. Society for Industrial and Applied Mathematics, 2005.

[Kal04] E. Kalai. Large robust games. *Econometrica*, 72(6):1631–1665, 2004.

[KP09] F. Kojima and P.A. Pathak. Incentives and stability in large two-sided matching markets. *American Economic Review*, 99(3):608–627, 2009.

[KPR10] F. Kojima, P.A. Pathak, and A.E. Roth. Matching with couples: Stability and incentives in large markets. Technical report, National Bureau of Economic Research, 2010.

[LR12] Katrina Ligett and Aaron Roth. Take it or leave it: Running a survey when privacy comes at a cost. *CoRR*, abs/1202.4741, 2012.

[MT03] Dov Monderer and Moshe Tennenholtz. k-implementation. In *Proceedings of the 4th ACM conference on Electronic commerce*, pages 19–28. ACM, 2003.

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103, 2007.

[MT09] Dov Monderer and Moshe Tennenholtz. Strong mediated equilibrium. *Artificial Intelligence*, 173(1):180–195, 2009.

[Nis07] N. Nisan. *Algorithmic game theory*. Cambridge Univ Pr, 2007.

[NOS12] Kobbi Nissim, Claudio Orlandi, and Rann Smorodinsky. Privacy-aware mechanism design. In *EC*, pages 774–789, 2012.

[NST12] Kobbi Nissim, Rann Smorodinsky, and Moshe Tennenholtz. Approximately optimal mechanism design via differential privacy. In *ITCS*, pages 203–213, 2012.

[PR13] Mallesh Pai and Aaron Roth. Privacy and mechanism design. *Sigecom Exchanges*, pages 8–29, 2013.

[Rou09] T. Roughgarden. Intrinsic robustness of the price of anarchy. In *STOC*, pages 513–522. ACM, 2009.

[Rou12] T. Roughgarden. The price of anarchy in games of incomplete information. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 862–879. ACM, 2012.

[RP76] D.J. Roberts and A. Postlewaite. The incentives for price-taking behavior in large exchange economies. *Econometrica*, pages 115–127, 1976.

[RR10] Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *STOC '10*, pages 765–774, 2010.

[RS12] Aaron Roth and Grant Schoenebeck. Conducting truthful surveys, cheaply. In *EC*, pages 826–843, 2012.

[Xia13] David Xiao. Is privacy compatible with truthfulness? In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 67–86. ACM, 2013.

A Additional Related Work

The most well studied problem is that of accurately answering numeric-valued queries on a data set. A basic result of [DMNS06] is that any low sensitivity query (i.e. the addition or removal of a single entry can change the value of the query by at most 1) can be answered efficiently and (ϵ-differential) privately while introducing only $O(1/\epsilon)$ error. Another fundamental result of [DKM+06, DRV10] is that differential privacy composes gracefully. Any algorithm composed of T subroutines, each of which are $O(\epsilon)$-differentially private, is itself $\sqrt{T}\epsilon$-differentially private. Combined, these give an efficient algorithm for privately answering any T low sensitivity queries with $O(\sqrt{T})$ effort, a result which we make use of.

Using computationally inefficient algorithms, it is possible to privately answer queries much more accurately [BLR08, DRV10, RR10, HR10, GHRU11, GRU12]. Combining the results of the latter two yields an algorithm which can privately answer arbitrary low sensitivity queries as they arrive, with error that scales only logarithmically in the number of queries. We use this when we consider games with large action spaces.

Our lower bounds for privately computing equilibria use recent information theoretic lower bounds on the accuracy queries can be answered while preserving differential privacy [DN03, DMT07, DY08, De12]. Namely, we construct games whose equilibria encode answers to large numbers of queries on a database.

Variants of differential privacy related to joint differential privacy have been considered in the setting of query release, specifically for analyst privacy [DNV12]. Specifically, the definition of one-analyst-to-many-analyst privacy used by [HRU13] can be seen as an instantiation of joint differential privacy.

B Proofs of Noise Tolerance of No Regret Algorithms (Section 4)

Proof of Lemma 17. Let $\pi_0, \ldots, \pi_T \in \Pi_k$ be any sequence of distributions and let $f \colon \{1, 2, \ldots, k\} \to \{1, 2, \ldots, k\}$ be any function. Then

$$\begin{aligned}
\rho(\pi_0, \ldots, \pi_T, L, f) &= \lambda(\pi_0, \ldots, \pi_T, L) - \lambda(f \circ \pi_0, \ldots, f \circ \pi_T, L) \\
&= 3\left(\lambda(\pi_0, \ldots, \pi_T, \overline{L}) - \lambda(f \circ \pi_0, \ldots, f \circ \pi_T, \overline{L})\right) \\
&= 3\left(\rho(\pi_0, \ldots, \pi_T, \overline{L}, f)\right).
\end{aligned}$$

The second equality follows from the definition of λ and from linearity of expectation. The Lemma now follows by setting $(\pi_0, \ldots, \pi_T) = \mathcal{A}_T(\overline{L})$, taking a maximum over $f \in \mathcal{F}$, and plugging in the guarantees of Theorem 11. $\square$

Proof of Lemma 12. Let $(\pi_0, \ldots, \pi_T)$ be any sequence of distributions and let $f \colon \{1, 2, \ldots, k\} \to \{1, 2, \ldots, k\}$

be any function. Then:

$$
\begin{aligned}
&\rho(\pi_0,\ldots,\pi_T,L,f)-\rho(\pi_0,\ldots,\pi_T,\widehat{L},f)\\
&=(\lambda(\pi_0,\ldots,\pi_T,L)-\lambda(f\circ\pi_0,\ldots,f\circ\pi_T,L))-(\lambda(\pi_0,\ldots,\pi_T,\widehat{L})-\lambda(f\circ\pi_0,\ldots,f\circ\pi_T,\widehat{L})).\\
&=(\lambda(\pi_0,\ldots,\pi_T,L)-\lambda(\pi_0,\ldots,\pi_T,\widehat{L}))+(\lambda(f\circ\pi_0,\ldots,f\circ\pi_T,\widehat{L})-\lambda(f\circ\pi_0,\ldots,f\circ\pi_T,\widehat{L}))\\
&=\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}\pi_t^j(l_t^j-\widehat{l}_t^j)\right)+\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}(f\circ\pi_t)^j(l_t^j-\widehat{l}_t^j)\right) && \text{(by definition of }\lambda)\\
&=\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{K}\pi_t^j z_t^j\right)+\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}(f\circ\pi_t)^j z_t^j\right) && \text{(by definition of }z) \quad (3)\\
&\le b\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{K}\pi_t^j\right)+b\left(\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{K}(f\circ\pi_t)^j\right) && (\forall j,t\ |z_t^j|\le b)\\
&=2b,
\end{aligned}
$$

where the final equality follows from the fact that $\pi_t, f\circ\pi_t$ are probability distributions. □

Proof of Corollary 13. We will prove only item 1, the proof for 2 is analogous. First, by the assumption of the theorem, we will have $\widehat{L}\in[0,1]^{T\times k}$ except with probability at most β. Therefore, by Theorem 11,

$$\mathop{\mathbb{P}}_{Z}\left[\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}),\widehat{L},\mathcal{F}_{\mathsf{fixed}})>\sqrt{\frac{2\log k}{T}}\right]\le\beta$$

Further, by Lemma 12, we know that $\widehat{L}\in[0,1]^{T\times k}$ implies

$$\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}),L,\mathcal{F})\le\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}),\widehat{L},\mathcal{F})+2b.$$

Combining, we have the desired result, i.e.

$$\mathop{\mathbb{P}}_{Z}\left[\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}),L,\mathcal{F}_{\mathsf{fixed}})>\sqrt{\frac{2\log k}{T}}+2b\right]\le\beta.$$

□

Proof of Lemma 14. Let $(\pi_0,\ldots,\pi_T)$ be any sequence of distributions and let $f\colon\{1,2,\ldots,k\}\to\{1,2,\ldots,k\}$ be any function. Recall by (3),

$$\rho(\pi_0,\ldots,\pi_T,L,f)-\rho(\pi_0,\ldots,\pi_T,\widehat{L},f)=\left(\tfrac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}\pi_t^j z_t^j\right)+\left(\tfrac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}(f\circ\pi_t)_j z_t^j\right). \quad (4)$$

We wish to place a high probability bound on the quantities:

$$Y_{\pi_0,\ldots,\pi_T}=\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k}\pi_t^j z_t^j.$$

Changing the order of summation,

$$Y_{\pi_0,\ldots,\pi_T} = \sum_{a_1,\ldots,a_T \in A} \left(\prod_{t=1}^{T} \pi_t^{a_t}\right)\left(\frac{1}{T}\sum_{t=1}^{T} z_t^{a_t}\right),$$

the equality follows by considering the following two ways of sampling elements z_t^j. The first expression represents the expected value of z_t^j if t is chosen uniformly from $\{1,2,\ldots,T\}$ and then j is chosen according to π_t. The second expression represents the expected value of z_t^j if $(a_1,\ldots,a_T)$ are chosen independently from the product distribution $\pi_1 \times \pi_2 \times \cdots \times \pi_T$ and then a_t is chosen uniformly from $(a_1,\ldots,a_T)$. These two sampling procedures induce the same distribution, and thus have the same expectation. Thus we can write:

$$\mathop{\mathbb{P}}_{Z}\left[Y_{\pi_0,\ldots,\pi_T} > \eta\right] \leq \max_{a_1,\ldots,a_T \in A} \mathop{\mathbb{P}}_{Z}\left[\frac{1}{T}\sum_{t=1}^{T} z_t^{a_t} > \eta\right] \leq \mathop{\mathbb{P}}_{Z}\left[\frac{1}{T}\sum_{t=1}^{T} z_t^{1} > \eta\right].$$

where the second inequality follows from the fact that the variables z_t^j are identically distributed. Applying Theorem 20, we have that for any $\eta < \sigma$,

$$\mathop{\mathbb{P}}_{Z}\left[Y_{\pi_0,\ldots,\pi_T} > \eta\right] \leq e^{-\eta^2 T/6\sigma^2}. \tag{5}$$

Let $(\pi_0,\ldots,\pi_T) = \mathcal{A}(\widehat{L})$. By Equation (4) we have

$$\begin{aligned}
&\mathop{\mathbb{P}}_{Z}\left[\rho(\mathcal{A}(\widehat{L}), L, f) - \rho(\mathcal{A}(\widehat{L}), \widehat{L}, f) > \eta\right] \\
&\leq \mathop{\mathbb{P}}_{Z}\left[\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k} \pi_t^j z_t^j > \eta/2\right] + \mathop{\mathbb{P}}_{Z}\left[\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{k} (f\circ\pi_t)_j z_t^j > \eta/2\right] \leq 2e^{-\eta^2 T/24\sigma^2}
\end{aligned}$$

where the last inequality follows from applying (5) to the sequences $(\pi_0,\ldots,\pi_T)$ and $(f\circ\pi_0,\ldots,f\circ\pi_T)$. The Lemma now follows by taking a union bound over $\mathcal{F}$. ☐

Proof of Corollary 15. First, we demonstrate that $\widehat{L} \in [0,1]^{T\times k}$ except with probability at most β, which will be necessary to apply the regret bounds of Theorem 11. Specifically:

$$\mathop{\mathbb{P}}_{Z}\left[\exists z_t^j \text{ s.t. } |z_t^j| > \frac{1}{3}\right] \leq Tk \mathop{\mathbb{P}}_{Z}\left[|z_1^1| > \frac{1}{3}\right] \leq 2Tke^{-1/6\sigma} \leq \beta/2, \tag{6}$$

where the first inequality follows from the union bound, the second from the definition of Laplacian r.v.'s and the last inequality follows from the assumption that $\sigma \leq 1/6\log(4Tk/\beta)$.

The Theorem now follows by conditoning on the event $\widehat{L} \in [0,1]^{T\times k}$ and combining the regret bounds of Theorem 11 with the guarantees of Lemma 14. For parsimony, we will only demonstate the first inequality, the second is analogous. Recall again by Theorem 11, we have that whenever $\widehat{l} \in [0,1]^{T\times k}$:

$$\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), \widehat{L}, \mathcal{F}_{\mathsf{fixed}}) \leq \sqrt{\frac{2\log k}{T}}.$$

Further, by Lemma 14, we know that:

$$\mathbb{P}_{Z}\left[\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), L, \mathcal{F}_{\mathsf{fixed}}) - \rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), \widehat{L}, \mathcal{F}_{\mathsf{fixed}}) > \eta\right] \leq 2|\mathcal{F}_{\mathsf{fixed}}|e^{-\eta^2 T/24\sigma^2}$$
$$= 2ke^{-\eta^2 T/24\sigma^2}.$$

Substituting $\eta = \sigma\sqrt{\frac{24\log(4k/\beta)}{T}}$, we get:

$$\mathbb{P}_{Z}\left[\rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), L, \mathcal{F}_{\mathsf{fixed}}) - \rho(\mathcal{A}_{\mathsf{fixed}}(\widehat{L}), \widehat{L}, \mathcal{F}_{\mathsf{fixed}}) > \eta\right] \leq \beta/2. \tag{7}$$

The result follows by combining (6) and (7). □

C Proofs for Computing Equilibria in Games with Few Actions (Section 5.2)

Proof of Theorem 21. Fix any player i, any pair of utility functions for i, u_i, u_i', and a tuple of utility functions u_{-i} for everyone else. To show differential privacy, we need to analyze the change in the distribution of the joint output for all players other than i, $(\pi_{-i,1}, \dots, \pi_{-i,T})$ when the input is (u_i, u_{-i}) as opposed to (u_i', u_{-i}).

It will be easier to analyze the privacy of a modified mechanism that outputs $(\widehat{l}_{-i,1}, \dots, \widehat{l}_{-i,T})$. Observe that this output is sufficient to compute $(\pi_{-i,1}, \dots, \pi_{-i,T})$ just by running $\mathcal{A}$. Thus, if we can show the modified output satisfies differential privacy, then same must be true for the mechanism as written.

For every player $i' \neq i$, action $j \in \{1, 2, \dots, k\}$, and $t \leq T$, we define the query $\mathcal{Q}^j_{i',t}(\cdot \mid \widehat{l}_{-i,1}, \dots, \widehat{l}_{-i,t-1})$ on the utility functions u_i, as well as u_{-i} the output of the mechanism in rounds $1, \dots, t-1$.

Query $\mathcal{Q}^j_{i',t}(u_i, u_{-i} \mid \widehat{l}_{-i,1}, \dots, \widehat{l}_{-i,t-1})$

Using u_{-i}, u_i and $\widehat{l}_{-i,1}, \dots, \widehat{l}_{-i,t-1}$, compute $l^j_{i',t}$. Observe that this can be done in the following steps:

1. Using $\widehat{l}_{-i,1}, \dots, \widehat{l}_{-i,t-1}$, $\mathcal{A}$, and u_{-i}, compute $\pi_{-i,1}, \dots, \pi_{-i,t-1}$.
2. Using $\pi_{-i,1}, \dots, \pi_{-i,t-1}$, $\mathcal{A}$, and u_i, compute $\pi_{i,1}, \dots, \pi_{i,t-1}$.
3. Using $\pi_{t-1} = (\pi_{i,t-1}, \pi_{-i,t-1})$, $\mathcal{A}$, and u_i, compute $l^j_{i',t}$.

Observe that the only step of the query computation that directly involves u_i is the second. Changing player i's utility function from u_i to u_i' can (potentially) affect $\pi_{i,t-1}$, and can (potentially) alter it to an arbitrary state $\overline{\pi}_{i,t-1}$. However, observe that

$$\mathcal{Q}^j_{i',t}(u_i \mid u_{-i}, \widehat{l}_{-i,1}, \dots, \widehat{l}_{-i,t-1}) = 1 - \mathop{\mathbb{E}}_{\pi_{-i',t}}[u_{i'}(j, a_{-i'})]$$
$$= 1 - \mathop{\mathbb{E}}_{\pi_{-(i,i'),t}}\left[\mathop{\mathbb{E}}_{\pi_{i,t}}[u_{i'}(j, a_i, a_{-(i',i)})]\right] \leq 1 - \mathop{\mathbb{E}}_{\pi_{-(i',i),t}}\left[\mathop{\mathbb{E}}_{\overline{\pi}_{i,t}}[u_{i'}(j, a_i, a_{-(i',i)}) + \gamma]\right]$$
$$= Q^j_{i',t}(u_i', \mid u_{-i}, \widehat{l}_{-i,1}, \dots, \widehat{l}_{-i,t-1}) + \gamma,$$

where the inequality comes from the fact that $u_{i'}$ is assumed to be γ-sensitive in the action of player i (Definition 3), and by linearity of expectation. A similar argument shows:

$$\mathcal{Q}^j_{i',t}(u_i \mid u_{-i}\widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,t-1}) \geq Q^j_{i',t}(u'_i \mid u_{-i}\widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,t-1}) - \gamma.$$

Note two facts about these queries: (1) The answer to $Q^j_{i',t}$ is exactly $l^j_{i',t}$, thus the noisy output to these queries (i.e. answer plus $\mathrm{Lap}(\sigma)$) is indeed equal to the output of the (modified) algorithm $\mathrm{NRLAPLACE}^{\mathcal{A}}$. (2) The noisy losses $\widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,t-1}$ have already been computed when the mechanism reaches round t, thus the mechanism fits the definition of adaptive composition.

Thus, we have rephrased the output $(\widehat{l}_{i',1}, \ldots, \widehat{l}_{i',T})$ as computing the answers to nkT (adaptively chosen) queries on $(u_1, \ldots, u_n)$, each of which is γ-sensitive to the input u_i. Thus the Theorem follows from our choice of $\sigma = \gamma\varepsilon^{-1}\sqrt{8nkT\log(1/\delta)}$ and Theorems 18 and 19. □

D Proofs for Computing Equilibria in Games with Many Actions (Section 5.3)

In this section we will give a more complete treatment of our algorithms for computing equilibria in games with many actions but bounded type spaces. First, we will formally state the privacy and accuracy guarantees of the Median Mechanism (see [RR10, HR10]).

Theorem 26 (Median Mechanism For General Queries)**.** *Consider the following R-round experiment between a mechanism $\mathcal{M}_M$, who holds a tuple $u_1, \ldots, u_N \in \mathcal{U}$, and a adaptive querier $\mathcal{B}$. For every round $r = 1, 2, \ldots, R$:*

1. *$\mathcal{B}(Q_1, a_1, \ldots, Q_{r-1}, a_{r-1}) = Q_r$, where Q_r is a γ-sensitive query.*
2. *$a_r \leftarrow_R \mathcal{M}_M(u_1, \ldots, u_n; Q_r)$.*

For every $\varepsilon, \delta, \gamma, \beta \in (0, 1], N, R, U \in \mathbb{N}$, there is a mechanism $\mathcal{M}_M$ such that for every $\mathcal{B}$

1. *The transcript $(Q_1, a_1, \ldots, Q_R, a_R)$ satisfies (ε, δ)-differential privacy.*
2. *With probability $1 - \beta$ (over the randomizations of $\mathcal{M}_M$), $|a_r - Q_r(u_1, \ldots, u_N)| \leq \alpha_{\mathcal{M}_M}$ for every $r = 1, 2, \ldots, R$ and for*

$$\alpha_{\mathcal{M}_M} = 16\varepsilon^{-1}\gamma\sqrt{N\log U}\log(2R/\beta)\log(4/\delta).$$

D.1 Noisy No-Regret via the Median Mechanism

We now define our algorithm for computing equilibria in games with exponentially many actions.

To keep notation straight, we will use $\mathbf{u} = (u_1, \ldots, u_N)$ to denote the utility functions specified by each of the n players, and $v \in \mathcal{U}$ to denote a utility function considered within the mechanism. Let $U = |\mathcal{U}|$, the size of the set of possible utility functions for any player.

First we sketch some intuition for how the mechanism works. In particular, why we cannot simply substitute the Median Mechanism for the Laplace mechanism and get a better error bound. Recall the queries we used in analyzing the Laplace-based algorithm $Q^j_{i,t}(\cdot \mid u_{-i}, \widehat{l}_{-i,1}, \ldots, \widehat{l}_{-i,T})$ in our previous analysis. We were able to argue that fixing u_{-i} and the previous noisy losses, the query was low-sensitivity as a function of its input u_i. This argument relied on the fact that we were effectively running *independent* copies of the Laplace mechanism, which guarantees that the answers given to each query do not explicitly depend on the previous queries that were asked (although the queries themselves may be correlated). However, in the mechanism we are about to define, the queries are all answered using a *single* instantiation of the Median mechanism. The Median mechanism correlates its answers across queries, and thus the answers to one query may depend on the previous queries that were made. This fact will be problematic, because the description of the queries $Q^j_{i,t}$ contains the utility functions u_{-i}. Thus, the queries we made to construct the output for players other than i will actually contain information about u_{-i}, and we cannot guarantee that this information does not leak into the answers given to other sets of players.

We address this problem by asking a larger set of queries whose description does not depend on any particular player's utility function. We will make the set of queries large enough that they will actually contain every query that we might possibly have asked in the Laplace-based algorithm, and each player can select from the larger set of answers only those which she needs to compute her losses. Since the queries do not depend on any utility function, we do not have to worry about leaking the description of the queries.

In order to specify the mechanism it will be easier to define the following family of queries first. Let i be any player, j any action, t any round of the algorithm, and v any utility function. The queries will be specified by these parameters and a sequence $\Lambda_1, \ldots, \Lambda_{t-1}$ where $\Lambda_{t'} \in \mathbb{R}^{n \times k \times \mathcal{U}}$ for every $1 \leq t' \leq t-1$. Intuitively, the query is given a description of the "state" of the mechanism in all previous rounds. Each state variable Λ_t encodes the losses that would be experienced by every possible player i and every action j and every utility function v, *given that the previous $t-1$ rounds of the mechanism were played using the real utility functions*. We will think of the variables $\Lambda_1, \ldots, \Lambda_{t-1}$ as having been previously sanitized, and thus we do not have to worry about the fact that these state variables encode information about the real utility functions.

$\mathcal{Q}^j_{i,t,v}(u_1, \ldots, u_N \mid \Lambda_1, \ldots, \Lambda_{t-1})$

Using $u_1, \ldots, u_N \mid \Lambda_1, \ldots, \Lambda_{t-1}$, compute $l^j_{i,t,v} = 1 - \mathbb{E}_{\pi_{-i,t}}\left[u_i(j, a_{-i})\right]$. This computation can be done in the following steps:

1. For every $i' \neq i$, use $\Lambda^j_{i',1,u_{i'}}, \ldots, \Lambda^j_{i',t-1,u_{i'}}$, $\mathcal{A}$, and $u_{i'}$ to compute $\pi_{i',1}, \ldots, \pi_{i',t-1}$.
2. Using $\pi_{-i,t-1}$, compute $l^j_{i,t,v}$.

Observe that $Q^j_{i,t,v}$ is γ-sensitive for every player i, step t, action j, and utility function v. To see why, consider what happens when a specific player i' switches her input from $u_{i'}$ to $u'_{i'}$. In that case that $i = i'$, this has no effect on the query answer, because player i's utility is never used in computing $Q^i_{t,j,v}$. In the case that $i' \neq i$ then the utility function of player i' can (potentially) affect the computation of $\pi_{i',t-1}$, and can (potentially) change it to an arbitrary state $\overline{\pi}_{i',t-1}$. But then γ-sensitivity follows from the γ-sensitivity of u_i, the definition of $l^j_{i,t,v}$, and linearity of expectation. Notice that $u_{i'}$ does not, however, affect the state of any other players, who will use the losses $\Lambda_1, \ldots, \Lambda_{t-1}$ to generate their states, not the actual states of the other players.

Now that we have this family of queries in places, we can describe the algorithm. Our mechanism uses two steps. At a high level, there is an inner mechanism, NRMEDIAN-SHARED, that will use the Median Mechanism to answer each query $Q^j_{i,t,v}\left(\cdot \mid \widehat{\Lambda}_1, \ldots, \widehat{\Lambda}_{t-1}\right)$, and will output a set of noisy losses $\widehat{\Lambda}_1, \ldots, \widehat{\Lambda}_T$. The properties of the Median Mechanism will guarantee that these losses satisfy (ε, δ)-differential privacy (in the standard sense of Definition 4).

There is also an outer mechanism that takes these losses and, for each player, uses the losses corresponding to her utility function to run a no-regret algorithm. This is NRMEDIAN which takes the sequence $\widehat{\Lambda}_1, \ldots, \widehat{\Lambda}_T$ and using the utility function u_i will compute the equilibrium strategy for player i. Since each player's output can be determined only from her own utility function and a set of losses that is (ε, δ)-differentially private with respect to every utility function, the entire mechanism will satisfy (ε, δ)-joint differential privacy.

NRMEDIAN-SHARED$^{\mathcal{A}}(u_1, \ldots u_N)$
  PARAMS: $\varepsilon, \delta, \gamma \in (0, 1], n, k, T \in \mathbb{N}$
  FOR: $t = 1, 2, \ldots, T$
    LET: $\widehat{l}^j_{i,t,v} = \mathcal{M}_M\left(u_1, \ldots, u_N; Q^j_{i,t,v}(\cdot \mid \widehat{\Lambda}_1, \ldots, \widehat{\Lambda}_{t-1})\right)$ for every i, j, v.
    LET: $\widehat{\Lambda}^j(i, t, v) = \widehat{l}^j_{i,t,v}$ for every i, j, v.
  END FOR
  OUTPUT: $(\widehat{\Lambda}_1, \ldots, \widehat{\Lambda}_T)$.

NRMEDIAN$^{\mathcal{A}}(u_1, \ldots u_N)$
  PARAMS: $\varepsilon, \delta, \Delta \in (0, 1], n, k, T \in \mathbb{N}$
  LET: $(\widehat{\Lambda}_1, \ldots, \widehat{\Lambda}_T) = $ NRMEDIAN-SHARED$^{\mathcal{A}}(u_1, \ldots, u_N)$.
  FOR: $i = 1, \ldots, N$
  LET: $\pi_{i,1}$ be the uniform distribution over $\{1, 2, \ldots, k\}$.
    FOR: $t = 1, \ldots, T$
      LET: $\pi_{i,t} = \mathcal{A}\left(\pi_{i,t-1}, \widehat{\Lambda}_{i,t-1,u_i}\right)$
    END FOR
    OUTPUT TO PLAYER i: $(\pi_{i,1}, \ldots, \pi_{i,T})$.
  END FOR

Theorem 27 (Privacy of NRMEDIAN)**.** *The algorithm* NRMEDIAN *satisfies* (ε, δ)*-joint differential privacy.*

Proof. Observe that NRMEDIAN can be written as $h(\mathbf{u}) = (f_1(g(\mathbf{u})), \ldots, f_N(g(\mathbf{u})))$ where f_i depends only on u_i for every player i. (Here, g is NRMEDIAN-SHARED and f_i is the i-th iteration of the main loop in NRMEDIAN). The privacy of the Median Mechanism (Theorem 26) directly implies that g is (ε, δ)-differentially private (in the standard sense).

Consider a player i and two profiles $\mathbf{u}, \mathbf{u}'$ that differ only in the input of player i, and consider the output $(f_{-i}(g(\mathbf{u})))$. Let $S \subseteq \mathrm{Range}(f_{-i})$ and let $R(\mathbf{u}) = \left\{o \in \mathrm{Range}(g) \mid f^{-i}(o) \in S\right\}$. Notice that f is deterministic, so R is well-defined. Also notice that R depends only on S and $\mathbf{u}_{-i}$ (in particular, not on u_i). Then we have

$$\begin{aligned}
\mathop{\mathbb{P}}_{h(\mathbf{u})}\left[h^{-i}(\mathbf{u}) \in S\right] &= \mathop{\mathbb{P}}_{g(\mathbf{u})}\left[g(\mathbf{u}) \in R(\mathbf{u}) = R(\mathbf{u}')\right] \\
&\leq e^{\varepsilon} \mathop{\mathbb{P}}_{g(\mathbf{u}')}\left[g(\mathbf{u}') \in R(\mathbf{u}) = R(\mathbf{u}')\right] + \delta \\
&\leq e^{\varepsilon} \mathop{\mathbb{P}}_{h(\mathbf{u})}\left[h^{-i}(\mathbf{u}') \in S\right] + \delta
\end{aligned}$$

where the first inequality follows from the (standard) (ε, δ)-differential privacy of g. Thus, NRMEDIAN satisfies (ε, δ)-joint differential privacy. $\square$

D.2 Computing Approximate Equilibria

Theorem 28 (Computing CE). *Let $\mathcal{A}$ be $\mathcal{A}_{\mathsf{swap}}$. Fix the environment, i.e the number of players n, the number of actions k, number of possible utility functions U, sensitivity of the game γ and desired privacy (ϵ, δ). Suppose β and T are such that:*

$$16\varepsilon^{-1}\gamma\sqrt{n\log U}\log(2nkTU/\beta)\log(4/\delta) \leq \tfrac{1}{6} \tag{8}$$

Then with probability at least $1-\beta$ the algorithm $\text{NRMEDIAN}^{\mathcal{A}_{\mathsf{fixed}}}$ *returns an α-approximate CE for:*[10]

$$\alpha = \tilde{O}\left(\frac{\gamma\sqrt{N}\log^{3/2}U\log(k/\beta)\log(1/\delta)}{\varepsilon}\right).$$

Again, considering 'low sensitivity' games where γ is $O(1/n)$, the theorem says that fixing the desired degree of privacy, we can compute an α-approximate equilibrium for $\alpha = \tilde{O}\left(\frac{(\log U)^{\frac{3}{2}}\log k}{\sqrt{N}}\right)$. The tradeoff to the old results is in dependence on the number of actions. The results in the previous section had a $\sqrt{k}$ dependence on the number of actions k. This would have no bite if k grew even linearly in n. We show that positive results still exist if the number of possible private types is is bounded - the dependence on the number of actions and the number of types is now logarithmic. However this comes with two costs. First, we can only consider situations where the number of types any player could have is bounded, and grows sub-exponentially in n. Second, we lose computational tractability– the running time of the median mechanism is exponential in the number of players in the game.

Proof. By the accuracy guarantees of the Median Mechanism:

$$\mathop{\mathbb{P}}_{\mathcal{M}_M}\left[\exists i,t,j,v \text{ s.t. } |\widehat{l}^j_{i,t,v} - l^j_{i,t,v}| > A_{\mathcal{M}_M}\right] \leq \beta$$

where

$$\alpha_{\mathcal{M}_M} = 16\gamma\varepsilon^{-1}\sqrt{n\log U}\log(2nkTU/\beta)\log(4/\delta)$$

By (8), $\alpha_{\mathcal{M}_M} \leq 1/6$. Therefore,

$$\mathop{\mathbb{P}}_{\mathcal{M}_M}\left[\exists i,j,t,v \text{ s.t. } |\widehat{l}^j_{i,t,v} - l^j_{i,t,v}| > \tfrac{1}{6}\right] \leq \beta$$

Applying Theorem 13 and substituting $A_{\mathcal{M}_M}$, we obtain:

$$\mathop{\mathbb{P}}_{Z}\left[\exists i \text{ s.t. } \rho(\pi_{i,1},\ldots,\pi_{i,T},L,\mathcal{F}_{\mathsf{swap}}) > \sqrt{\frac{2k\log k}{T}} + 2\alpha_{\mathcal{M}_M}\right] \leq \beta$$

Now we can choose $\sqrt{T} = k(\gamma\sqrt{n})^{-1}$ to conclude the proof. $\square$

[10] Here, $\tilde{O}$ hides lower order $\text{poly}(\log n, \log\log k, \log T, \log\log U\log(1/\gamma), \log(1/\varepsilon), \log\log(1/\beta), \log\log(1/\delta))$ terms.

E Proof of the Lower Bound (Theorem 24)

Given a database $D \in \{0,1\}^n$, $D = (d_1, \dots, d_n)$ and m queries $\mathcal{Q} = \{q_1, \dots, q_m\}$, we will construct the following ($N = n + m \log n$)-player 2-action game. We denote the set of actions for each player by $A = \{0, 1\}$. We also use $\{(j,h)\}_{j \in [m], h \in [\log n]}$ to denote the $m \log n$ players $\{n+1, \dots, n + m \log n\}$. For intuition, think of player (j,h) as computing the h-th bit of $q_j(D)$.

Each player $i \in [n]$ has the utility function

$$u_i(\mathbf{a}) = \begin{cases} 1 & \text{if } a_i = d_i \\ 0 & \text{otherwise} \end{cases}$$

That is, player i receives utility 1 if they play the action matching the i-th entry in D, and utility 0 otherwise. Clearly, these are 0-sensitive utility functions.

The specification of the utility functions for the query players (j,h) is somewhat more complicated. First, we define the functions $f_h, g_h \colon [0,1] \to [0,1]$ as

$$f_h(x) = 1 - \min_{r \in \{0, \dots, 2^{h-1}-1\}} \left| x - (2^{-(h+1)} + r 2^{-(h-1)}) \right|$$
$$g_h(x) = 1 - \min_{r \in \{0, \dots, 2^{h-1}-1\}} \left| x - (2^{-h} + 2^{-(h+1)} + r 2^{-(h-1)}) \right|$$

Each player (j,h) will have the utility function

$$u_{(j,h)}(a_{-(j,h)}, 0) = f_h(q_j(a_1, \dots, a_n))$$
$$u_{(j,h)}(a_{-(j,h)}, 1) = g_h(q_j(a_1, \dots, a_n))$$

Since $q(a_1, \dots, a_n)$ is defined to be $1/n$-sensitive in the actions $a_1, \dots, a_n$, and f_h, g_h are 1-Lipschitz in x, $u_{(j,h)}$ is also $1/n$-sensitive.

Also notice that since $\mathcal{Q}$ is part of the definition of the game, we can simply define the set of feasible utility functions to be all those we have given to the players. For the data players we only used 2 distinct utility functions, and each of the $m \log n$ query players may have a distinct utility function. Thus we only need the set $\mathcal{U}$ to be a particular set of utility functions of size $m \log n + 2$ in order to implement the reduction.

Now we can analyze the structure of α-approximate equilibrium in this game, and show how, given any equilibrium set of strategies for the query players, we can compute a set of $O(\alpha)$-approximate answers to the set of queries $\mathcal{Q}$.

We start by claiming that in any α-approximate CCE, every data player players the action d_i in most rounds. Specifically,

Claim 29. *Let π be any distribution over A^N that constitutes an α-approximate CCE of the game described above. Then for every data player i,*

$$\mathop{\mathbb{P}}_{\pi}[a_i \neq d_i] \leq \alpha.$$

Proof.

$$\begin{aligned} \mathop{\mathbb{P}}_{\pi}[a_i \neq d_i] &= 1 - \mathop{\mathbb{E}}_{\pi}[u_i(a_i, a_{-i})] \\ &\leq 1 - \left(\mathop{\mathbb{E}}_{\pi}[u_i(d_i, a_{-i})] - \alpha \right) && \text{(Definition of } \alpha\text{-approximate CCE)} \\ &= 1 - (1 - \alpha) = \alpha && \text{(Definition of } u_i\text{)} \end{aligned} \quad (9)$$

□

The next claim asserts that if we view the actions of the data players, $a_1, \ldots, a_n$, as a database, then $q(a_1, \ldots, a_n)$ is close to $q(d_1, \ldots, d_n)$ on average.

Claim 30. *Let π be any distribution over A^N that constitutes an α-approximate CCE of the game described above. Let $q \subseteq [n]$ be any subset-sum query. Then*

$$\mathop{\mathbb{E}}_{\pi}\left[|q(d_1, \ldots, d_n) - q(a_1, \ldots, a_n)|\right] \le \alpha.$$

Proof.

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\pi}\left[|q(d_1, \ldots, d_n) - q(a_1, \ldots, a_n)|\right] = \mathop{\mathbb{E}}_{\pi}\left[\frac{1}{n}\sum_{i \in q}(d_i - a_i)\right] \\
&\le \frac{1}{n}\sum_{i \in q}\mathop{\mathbb{E}}_{\pi}\left[|d_i - a_i|\right] = \frac{1}{n}\sum_{i \in q}\mathop{\mathbb{P}}_{\pi}\left[a_i \neq d_i\right] \\
&\le \frac{1}{n}\sum_{i \in q}\alpha \quad \le \alpha \qquad\qquad (\text{Claim 29, } q \subseteq [n]) \qquad (10)
\end{aligned}$$

□

We now prove a useful lemma that relates the expected utility of an action (under any distribution) to the expected difference between $q_j(a_1, \ldots, a_n)$ and $q_j(D)$.

Claim 31. *Let μ be any distribution over A^N. Then for any query player (j, h),*

$$\left|\mathop{\mathbb{E}}_{\mu}\left[u_{(j,h)}(0, a_{-(j,h)})\right] - f_h(q_j(D))\right| \le \mathop{\mathbb{E}}_{\mu}\left[|q_j(a_1, \ldots, a_n) - q_j(D)|\right], \textit{ and}$$

$$\left|\mathop{\mathbb{E}}_{\mu}\left[u_{(j,h)}(1, a_{-(j,h)})\right] - g_h(q_j(D))\right| \le \mathop{\mathbb{E}}_{\mu}\left[|q_j(a_1, \ldots, a_n) - q_j(D)|\right].$$

Proof. We prove the first assertion, the proof of the second is identical.

$$\begin{aligned}
&\left|\mathop{\mathbb{E}}_{\mu}\left[u_{(j,h)}(0, a_{-i})\right] - f_h(q_j(D))\right| \\
&= \left|\mathop{\mathbb{E}}_{\mu}\left[f_h(q_j(a_1, \ldots, a_n)) - f_h(q_j(D))\right]\right| \\
&\le \mathop{\mathbb{E}}_{\pi}\left[|q_j(a_1, \ldots, a_n) - q_j(D)|\right] \qquad\qquad (f_h \text{ is 1-Lipschitz}) \qquad (11)
\end{aligned}$$

□

The next claim, which establishes a lower bound on the expected utility player (j, h) will obtain for playing a fixed action, is an easy consequence of Claims 30 and 31.

Claim 32. *Let π be any distribution over A^N that constitutes an α-approximate CCE of the game described above. Then for every query player (j, h),*

$$\left|\mathop{\mathbb{E}}_{\pi}\left[u_{(j,h)}(0, a_{-i})\right] - f_h(q_j(D))\right| \le \alpha, \textit{ and}$$

$$\left|\mathop{\mathbb{E}}_{\pi}\left[u_{(j,h)}(1, a_{-i})\right] - g_h(q_j(D))\right| \le \alpha.$$

Now we state a simple fact about the functions f_h and g_h. Informally, this asserts that we can find alternating intervals of width nearly 2^{-h}, that nearly partition $[0,1]$, in which $f_h(x)$ is significantly larger than $g_h(x)$ or vice versa.

Observation 33. *Let $\beta \le 2^{-(h+1)}$. If*

$$x \in \bigcup_{r \in \left\{0,1,\ldots,2^{h-1}-1\right\}} \left(r2^{-h} + \beta, (r+1)2^{-h} - \beta\right)$$

then $f_h(x) > g_h(x) + \beta$. We denote this region $F_{h,\beta}$. Similarly, if

$$x \in \bigcup_{r \in \left\{0,1,\ldots,2^{h-1}-1\right\}} \left((r+1)2^{-h} + \beta, (r+2)2^{-h} - \beta\right)$$

then $g_h(x) > f_h(x) + \beta$. We denote this region $G_{h,\beta}$

For example, when $h = 3$, $F_{3,\beta} = [0, \frac{1}{8} - \beta] \cup [\frac{2}{8} + \beta, \frac{3}{8} - \beta] \cup [\frac{4}{8} + \beta, \frac{5}{8} - \beta] \cup [\frac{6}{8} + \beta, \frac{7}{8} - \beta]$.

By combining this fact, with Claim 32, we can show that if $q_j(D)$ falls in the region $F_{h,\alpha}$, then in an α-approximate CCE, player (j,h) must be playing action 0 'often'.

Claim 34. *Let π be any distribution over A^N that constitutes an α-approximate CCE of the game described above. Let $j \in [m]$ and $2^{-h} \ge 10\alpha$. Then, if $q_j(D) \in F_{h,9\alpha}$, $\mathbb{P}_\pi[a_i = 0] \ge 2/3$. Similarly, if $q_j(D) \in G_{h,9\alpha}$, then $\mathbb{P}_\pi[a_i = 1] \ge 2/3$.*

Proof. We prove the first assertion. The proof of the second is identical. If player (j,h) plays the fixed action 0, then, by Claim 32,

$$\mathop{\mathbb{E}}_{\pi}\left[u_{(j,h)}(0, a_{-(j,h)})\right] \ge f_h(q_j(D)) - \alpha.$$

Thus, if π is an α-approximate CCE, player (j,h) must receive at least $f_h(q_j(D)) - 2\alpha$ under π. Assume towards a contradiction that $\mathbb{P}\left[a_{(j,h)} = 0\right] < 2/3$. We can bound player (j,h)'s expected utility as follows:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathbf{a} \leftarrow_R \pi}\left[u_{(j,h)}(\mathbf{a})\right] \\
&= \mathbb{P}\left[a_{(j,h)} = 0\right] \mathop{\mathbb{E}}_{\pi}\left[u_{(j,h)}(0, a_{-(j,h)}) \mid a_{(j,h)} = 0\right] \\
&\quad + \mathbb{P}\left[a_{(j,h)} = 1\right] \mathop{\mathbb{E}}_{\pi}\left[u_{(j,h)}(1, a_{-(j,h)}) \mid a_{(j,h)} = 1\right] \\
&\le \mathbb{P}\left[a_{(j,h)} = 0\right] \left(f_h(q_j(D)) + \mathop{\mathbb{E}}_{\mathbf{a} \leftarrow_R \pi}\left[|q_j(a_1,\ldots,a_n) - q_j(D)| \mid a_{(j,h)} = 0\right]\right) \\
&\quad + \mathbb{P}\left[a_{(j,h)} = 1\right] \left(g_h(q_j(D)) + \mathop{\mathbb{E}}_{\mathbf{a} \leftarrow_R \pi}\left[|q_j(a_1,\ldots,a_n) - q_j(D)| \mid a_{(j,h)} = 1\right]\right) && (12) \\
&= f_h(q_j(D)) + \mathop{\mathbb{E}}_{\mathbf{a} \leftarrow_R \pi}\left[|q_j(a_1,\ldots,a_n) - q_j(D)|\right] - \mathbb{P}\left[a_{(j,h)} = 1\right] \left(f_h(q_j(D)) - g_h(q_j(D))\right) \\
&\le f_h(q_j(D)) + \alpha - 9\alpha\, \mathbb{P}\left[a_{(j,h)} = 1\right] && (13) \\
&< f_h(q_j(D)) - 2\alpha && (14)
\end{aligned}$$

Line (12) follows from the Claim 31 (applied to the distributions $\pi \mid a_{(j,h)} = 0$ and $\pi \mid a_{(j,h)} = 1$). Line (13) follows from Claim 30 (applied to the expectation in the second term) and the fact that $q_j(D) \in F_{h,9\alpha}$ (applied to the difference in the final term). Line (14) follows from the assumption that $\mathbb{P}\left[a_{(j,h)} = 0\right] < 2/3$. Thus we have established a contradiction to the fact that π is an α-approximate CCE. $\square$

Given the previous claim, the rest of the proof is fairly straightforward. For each query j, we will start at $h = 1$ and consider two cases: If player $(j, 1)$ plays 0 and 1 with roughly equal probability, then we must have that $q_j(D) \notin F_{1,9\alpha} \cup G_{1,9\alpha}$. It is easy to see that this will confine $q_j(D)$ to an interval of width 18α, and we can stop. If player $(j, 1)$ does play one action, say 0, a significant majority of the time, then we will know that $q_j(D) \in F_{1,9\alpha}$, which is an interval of width $1/2 - 9\alpha$. However, now we can consider $h = 2$ and repeat the case analysis: Either $(j, 2)$ does not significantly favor one action, in which case we know that $q_j(D) \notin F_{2,9\alpha} \cup G_{2,9\alpha}$, which confines $q_j(D)$ to the union of two intervals, each of width 18α. However, only one of these intervals will be contained in $F_{1,9\alpha}$, which we know contains $q_j(D)$. Thus, if we are in this case, we have learned $q_j(D)$ to within 18α and can stop. Otherwise, if player $(j, 2)$ plays, say, 0 a significant majority of the time, then we know that $q_j(D) \in F_{1,9\alpha} \cap F_{2,9\alpha}$, which is an interval of width $1/4 - 9\alpha$. It is not too difficult to see that we can repeat this process as long as $2^{-h} \geq 18\alpha$, and we will terminate with an interval of width at most 36α that contains $q_j(D)$.