Conservative or Liberal? Personalized Differential Privacy

Zach Jorgensen [#1], Ting Yu [*#2], Graham Cormode [τ3]

[#] *North Carolina State University, USA* [1] zjorgen@ncsu.edu

[*] *Qatar Computing Research Institute, Qatar* [2] tyu@qf.org.qa

[τ] *University of Warwick, UK* [3] g.cormode@warwick.ac.uk

***Abstract*—Differential privacy is widely accepted as a powerful framework for providing strong, formal privacy guarantees for aggregate data analysis. A limitation of the model is that the same level of privacy protection is afforded for all individuals. However, it is common that the data subjects have quite different expectations regarding the acceptable level of privacy for their data. Consequently, differential privacy may lead to insufficient privacy protection for some users, while over-protecting others.**

We argue that by accepting that not all users require the same level of privacy, a higher level of utility can often be attained by not providing excess privacy to those who do not want it. We propose a new privacy definition called *personalized differential privacy* (PDP), a generalization of differential privacy in which users specify a personal privacy requirement for their data. We then introduce several novel mechanisms for achieving PDP. Our primary mechanism is a general one that automatically converts any existing differentially private algorithm into one that satisfies PDP. We also present a more direct approach for achieving PDP, inspired by the well-known exponential mechanism. We demonstrate our framework through extensive experiments on real and synthetic data.

I. Introduction

Differential privacy [6], [9] is a powerful framework for providing strong, formal privacy guarantees for aggregate data analysis. Differential privacy ensures that no individual user can significantly affect the output of an aggregate computation; consequently, an adversary observing the output is unable to determine, with high probability, whether a particular user's data was present in the input. A common approach for achieving differential privacy is to inject random noise that is carefully calibrated according to the sensitivity of the computation (i.e., the maximum impact that any one user can have on the output), and a global privacy parameter.

In this work we propose a novel privacy definition to address an important limitation of differential privacy—that it provides only a uniform level of privacy protection for all users in a dataset. This "one size fits all" approach ignores the reality that data privacy is a personal and multifaceted concept, and that different individuals may have very different expectations for the privacy of their personal data. Indeed, several studies in the psychology literature have observed that individuals typically fall into several distinct groups or clusters based on their privacy attitudes [4], [2], [1]. In particular, Berendt et al. conducted a large-scale study of attitudes to privacy in e-commerce [4]. They were able to distinguish a clear group of "privacy fundamentalists" and a distinct group of individuals with only marginal concern for privacy. The remaining respondents exhibited privacy attitudes in between the two extremes and differed on the focus of the privacy concern. Similar clusterings of users based on privacy attitudes have been observed by other researchers [2], [1], and have identified other factors which contribute, such as as cultural values [18], and income, age and political views [19].

In practice, when faced with a dataset comprising multiple users with different privacy expectations, a data analyst employing differential privacy has limited options. One possibility is to set the global privacy level high enough to satisfy even the privacy fundamentalists in the dataset. This is likely to introduce an unacceptable amount of noise into the analysis outputs, resulting in poor utility. On the other hand, setting a lower privacy level may force the analyst to exclude a significant portion of the dataset from analysis (i.e., the data belonging to the fundamentalists), which may also significantly harm utility. In contrast to traditional differential privacy, the personalized privacy model we introduce allows privacy requirements to be specified at the *user-level*. Within the proposed model, we present mechanisms that are able to take these individual privacy requirements into account to guarantee precisely the required level of privacy to each user, while in many cases providing significantly better utility than the naïve options discussed above.

As well as the possibility of improved data utility, allowing users a degree of control over the disclosure of their data offers other important benefits. There is evidence that when users are given control over their privacy they are more inclined to want to contribute their data for analysis in the first place, and to do so truthfully [24]. Indeed, the importance of control in the context of privacy is emphasized in a widely accepted definition of privacy from the psychology literature, due to Westin, as "the ability of the individual to control the terms under which personal information is acquired and used" [30].

A. Contributions

In this paper we consider the setting in which a trusted data analyst desires to publish aggregate statistics computed from a dataset comprising personal data of many individual users. Every user may potentially have a different privacy requirement for his or her data and the analyst would like to publish useful aggregate information about the data, while simultaneously complying with the individual privacy requirements of the contributors. To that end, we propose a new privacy framework called *Personalized Differential Privacy* (PDP), a generalization of differential privacy in which the

privacy requirements are specified at the user-level, rather than by a single, global privacy parameter. The privacy guarantees of our framework have the same strength and attack resistance as differential privacy, but are personalized to the preferences of all users in the input domain. We also show that the composition properties of differential privacy carry over to PDP, allowing complex privacy-preserving algorithms to be constructed from individual PDP components.

We introduce two novel mechanisms for achieving PDP. Our main goal is to design mechanisms that can take advantage of the non-uniform privacy requirements to attain better utility than could be achieved with differential privacy. Our first mechanism is general and can be used to easily and automatically convert *any* existing differentially private algorithm into a PDP algorithm. The mechanism is a two-step procedure that involves a non-uniform sampling step at the individual tuple level, followed by the invocation of an appropriate differentially private mechanism on the sampled dataset. In the sampling step, the inclusion probabilities for each tuple are calculated according to the individual privacy requirements of the corresponding user. We show that the two sources of randomness introduced by this two-step procedure combine to yield the precise personalized guarantee demanded. Our second mechanism is a more direct approach to achieve PDP, inspired by the well-known exponential mechanism of McSherry and Talwar [17]. The mechanism is applicable to common aggregates such as counts, medians, and min/max and can be shown to outperform the sampling-based mechanism in certain scenarios. In particular, we find that it is generally preferable to the sampling-based mechanism for *counts*, which are especially sensitive to the effects of sampling.

We conducted an extensive experimental study of several instantiations of the PDP framework, on both synthetic and real datasets. In particular, we studied both of our mechanisms for the important *count* and *median* functions. We then investigated the application of our sampling-based mechanism to the more complex task of multiple linear regression. Our results demonstrate the broad applicability of our framework and the utility gains that are possible by taking personal privacy preferences into account.

In the next section, we begin by describing differential privacy and introducing notation, as well as reviewing related work. In Section III we introduce our new privacy definition, followed by a discussion of how to satisfy the definition for arbitrary tasks, in Section IV. Section V presents our experimental study and Section VI concludes the paper.

II. Preliminaries

We introduce some notation and initial definitions, and briefly review the notion of differential privacy, upon which our work is based. We then discuss related work.

We model a *dataset* as a set of tuples from a universe $\mathcal{D}$, with one or more attributes $A_1,\ldots,A_d$. Every tuple in a dataset is assumed to be associated with a user in $\mathcal{U}$, where $\mathcal{U}$ is the universe of users (e.g., all of the customers of an online store, all of the patients of a given hospital, etc.).

Definition 1 (Dataset)**.** *A* dataset $D \subset \mathcal{D}$ *is a set of tuples* $D = \{t_1,\ldots,t_i,\ldots\}$ *from universe* $\mathcal{D}$*, where* $t_i \in A_1 \times \ldots \times A_d \times \mathcal{U}$*; the* A_i *are attributes; and* $\mathcal{U}$ *denotes the universe of users. We write* $t_{\mathcal{U}}$ *to denote the user associated with tuple* t*.*

The attributes $A_1,\ldots,A_d$ may be numeric or categorical. Note also that a dataset D will not necessarily contain a tuple for every $u \in \mathcal{U}$. Moreover, depending on the semantics of the data, or the analysis task being considered, it may be possible for a dataset to contain multiple tuples for the same user (e.g., all of the products a user has purchased), while in other cases it may not make sense for a dataset to contain more than one tuple per user (e.g., a tuple contains a user's annual salary).

For both differential privacy and personalized differential privacy, the notion of *neighboring datasets* is an important one. We say that two datasets are neighboring if one is a proper subset of the other and the larger dataset contains exactly one additional tuple.

Definition 2 (Neighboring datasets)**.** *Two datasets* $D, D' \subset \mathcal{D}$ *are said to be* neighboring*, or* neighbors*, denoted* $D \sim D'$*, if* $D \subset D'$ *and* $|D'| = |D| + 1$ *(or vice versa).*

We write $D \overset{t}{\sim} D'$ to denote that D and D' are neighbors and that $t \in D'$ and $t \notin D$.

A. Differential Privacy

A *mechanism* $\mathcal{M} : \mathcal{D} \to R$ is a randomized algorithm that takes a dataset as input and returns an output from some range R. The notation $\mathcal{M}(x)$ then denotes the probability distribution on R induced by input x. Informally, a mechanism is said to be *differentially private* if the probability distribution $\mathcal{M}(D)$ on any dataset D is approximately the same as $\mathcal{M}(D')$, for every $D \sim D'$. In other words, the mechanism's behavior should be (mostly) insensitive to the presence or absence of any one tuple in the input. More formally,

Definition 3 (ε-Differential Privacy [6], [9])**.** *Mechanism* $\mathcal{M} : \mathcal{D} \to R$ *satisfies* ε-differential privacy *if for all pairs* $D \sim D' \subset \mathcal{D}$ *and any set* $O \subseteq R$ *of possible outputs,*

$$\Pr[\mathcal{M}(D) \in O] \leq \mathrm{e}^{\varepsilon} \times \Pr[\mathcal{M}(D') \in O]$$

This definition protects against, for example, an adversary who knows the full input except for one tuple t: they are still unable to deduce even whether t was in the input. In the definition, $\varepsilon > 0$ is a publicly known *privacy parameter* that controls the strength of the differential privacy guarantee: a larger ε yields weaker privacy, while a smaller ε leads to stronger privacy. When ε is small, $\mathrm{e}^{\varepsilon} \approx 1 + \varepsilon$.

An important practical property of differential privacy is composability. That is, if we have k mechanisms $\mathcal{M}_1,\ldots,\mathcal{M}_k$, each of which independently satisfies ε_i-differential privacy, and we run these mechanisms on an input D in sequence, then the sequence is ε'-differentially private, where $\varepsilon' = \sum_{i=1}^{k} \varepsilon_i$ [7].

For real valued functions, i.e., $f : \mathcal{D} \to \mathbb{R}^d$, the most common way to satisfy differential privacy is to inject carefully chosen random noise into the output. The magnitude of the noise is adjusted according to the *global sensitivity* of the function, or the maximum extent to which any one tuple in the input can affect the output. Formally,

Definition 4 (Global Sensitivity [9]). *The global sensitivity of a function $f : \mathcal{D} \to \mathbb{R}^d$, is*

$$\Delta_f = \max_{D \sim D'} \left\| f(D) - f(D') \right\|_1$$

where $\|\cdot\|_1$ is the L_1 norm.

Note that global sensitivity does not depend on the input data but is a property of function f alone. A function f can be made ε-differentially private by adding random noise drawn from the Laplace distribution with mean zero and scale $\frac{\Delta_f}{\varepsilon}$ to its output. We will subsequently use the notation $\text{Lap}(\lambda)$ to denote the Laplace distribution with mean 0 and scale λ.

Theorem 1 (Laplace Mechanism [9]). *For a function $f : \mathcal{D} \to \mathbb{R}^d$, the mechanism that returns $f(D) + z^d$, where each z_i is drawn i.i.d. from $\text{Lap}(\frac{\Delta_f}{\varepsilon})$ satisfies ε-differential privacy.*

For functions where adding noise does not make sense or the output space is non-numeric, the *exponential mechanism* [17] can be used to achieve differential privacy. The exponential mechanism uses the concept of a score function, denoted $s(D,r)$, that returns a real-valued score to indicate the quality of output r with respect to the true output $f(D)$. A higher score is assumed to mean that r is closer to the true output $f(D)$. For a given score function s, the exponential mechanism, denoted $\mathcal{E}^s_\varepsilon$ is defined as follows.

Theorem 2 (Exponential Mechanism [17]). *For a score function $s : \mathcal{D} \times R \to \mathbb{R}$, the mechanism $\mathcal{E}^s_\varepsilon(D)$ that outputs $r \in R$ with probability proportional to $\exp(\frac{\varepsilon s(D,r)}{2\Delta_s})$, satisfies ε-differential privacy.*

Here, Δ_s is the global sensitivity of the score function and is defined slightly differently than in the context of the Laplace mechanism: $\Delta_s = \max_{D \sim D', r \in R} |s(D,r) - s(D',r)|$.

B. Related Work

A line of work, started by Xiao and Tao [27], introduced personalized privacy for k-anonymity. K-anonymity requires that every record in a dataset be indistinguishable from at least $k-1$ others, in terms of their identifying attributes [26]. Xiao and Tao's generalization was to allow each user to specify the minimum k they were comfortable with. Subsequently a slew of related approaches (e.g., [31], [28], [22], [29]) extended this to other methods for achieving k-anonymity and related definitions. However, these definitions have been criticized due to the feasibility of attacks that can lead to disclosure of sensitive attributes [16], [13], [32], and more robust notions, led by differential privacy, are now preferred.

Our primary mechanism for achieving personalized differential privacy involves the use of sampling to introduce non-uniform uncertainty at the tuple level. Although this is the first work to use sampling to realize a personalized notion of privacy, it has previously been combined with differential privacy for other purposes [11], [14], [15], [12], [10]. Li et al. showed that uniform random sampling in combination with differential privacy amplifies the privacy guarantee [14]. The result in [14] was motivated by the observation that since random sampling is often already an inherent part of data collection, one can take advantage of that existing randomness to lower privacy costs. Prior to this, Kasiviswanathan et al. implicitly used this amplification effect to build a private PAC learner for parity functions [11]. Gehrke et al. proposed a relaxation of differential privacy, called crowd-blending privacy which, although strictly weaker than differential privacy alone, when preceded by a random sampling step also satisfies differential privacy [10]. Aside from targeting a very different objective, our use of sampling in the present work also differs from all prior work described above by sampling tuples from the input data independently, with *non-uniform* probabilities: the inclusion probability for each tuple depends on the corresponding individual's privacy requirement (as well as a global threshold). It is this non-uniformity that enables the personalized privacy guarantees of PDP.

In a very recent manuscript [3], Alaggan et al. have independently developed a similar privacy notion to ours, called *heterogeneous differential privacy*, which to our knowledge is the only other work to consider differential privacy with non-uniform privacy guarantees. Their work differs from ours both in the presentation and in the technical contributions. In particular, the "stretching" mechanism proposed in [3], which is based on the Laplace mechanism and works by rescaling the input values according to the corresponding privacy parameters, applies only to a limited subset of real-valued functions; specifically, it cannot be used for functions like *median*, *min/max*, and many others, that rely on the exponential mechanism. It is also fundamentally incompatible with some other types of queries, such as counting the number of non-zero values in a dataset, where rescaling does not alter the answer to the query. In contrast, our primary mechanism for achieving PDP has no such restrictions; it can be used to automatically convert *any* differentially private algorithm—whether it is an instance of the Laplace mechanism, the exponential mechanism or even a composition of multiple differentially private components—into one that satisfies our personalized privacy definition. Additionally, unlike [3], we present an extensive empirical analysis comparing our mechanisms to several baseline approaches to demonstrate the advantages of PDP over differential privacy, in terms of utility.

Finally, it is worth mentioning a recent line of work on *privacy auctions* (surveyed in [21]), that is ostensibly similar to the present work. This line of work is mainly concerned with how to accurately compute statistics over a population of users who demand financial compensation for any privacy loss incurred by their participation. Users specify a (possibly non-uniform) valuation on their privacy that expresses their privacy cost incurred by participating in an ε-differentially private analysis (as a function of ε), and hence the amount of compensation due if their data is used. The analyst's job is to choose the users from whom to "buy data", such that the analyst's financial cost is minimized, while the computed statistic meets some utility goal. In other words, users are not guaranteed a certain level of privacy, but rather that they will be compensated in proportion to their privacy valuation, should their data be used. The privacy auction mechanisms ultimately provide a *uniform* privacy guarantee. In contrast, our setting allows users to individually specify a specific minimum level of privacy for their data, and the mechanisms that we develop guarantee *at least* the required privacy levels of all users.

III. Personalized Differential Privacy

In this section, we introduce *Personalized Differential Privacy (PDP)* and discuss its properties. In Section IV we present several mechanisms that satisfy the definition.

In contrast to traditional differential privacy, in which the privacy guarantee is controlled by a single, global privacy parameter (i.e., ε in Def. 3), PDP makes use of a *privacy specification*, in which each user in $\mathcal{U}$ independently specifies the privacy requirement for their data. More formally,

Definition 5 (Privacy Specification)**.** *A* privacy specification *is a mapping $\Phi : \mathcal{U} \to \mathbb{R}_+$ from users to personal privacy preferences, where a smaller value represents a stronger privacy preference. The notation Φ^u is used to denote the privacy preference corresponding to user $u \in \mathcal{U}$.*

For convenience, we may describe a specific instance of a privacy specification as a set of ordered pairs, e.g., $\Phi := \{(u_1, \varepsilon_1), (u_2, \varepsilon_2), \ldots\}$ where $u_i \in \mathcal{U}$ and $\varepsilon_i \in \mathbb{R}_+$. We also assume that a privacy specification contains a privacy preference for every $u \in \mathcal{U}$, or that a default privacy level, say $\varepsilon_{\text{def}} = 1.0$, is used. As will become clear, the privacy preference values in our model can be interpreted similarly to the ε parameter in traditional ε-differential privacy, so we expect privacy preferences to fall in the range $(0.01, 1.0)$. In practice it may be unreasonable to expect typical users to choose a meaningful numerical privacy setting. Rather, we envision a scenario in which a domain expert associates appropriate values with user-friendly descriptors (e.g., *low*, *medium* and *high privacy*) and users choose from those. This represents one possibility; in general, choosing an appropriate privacy parameter for differentially private systems is an open problem and we do not consider it further in this paper.

Our model assumes that the privacy specification is public knowledge. This mirrors the situation in traditional differential privacy, where the global privacy setting ε is assumed to be a public parameter. However, this means that the user's privacy parameter must not indicate anything about their sensitive values. We believe this to be a reasonable assumption, given that the privacy specification is defined at the user-level, rather than the tuple level. That is, one can think of the privacy setting as being a function of the user that owns the data rather than a function of the data itself. For example, a politician might have a higher privacy preference for her online browsing history (for instance) than the average user, by virtue of her profession—not because the data itself is inherently any more or less sensitive than that of an average user. In practice, a user might specify their privacy preferences at registration time (e.g., upon joining a service), before any data is generated; then all data that is subsequently produced by that user will use that previously-specified privacy setting. Nevertheless, it should be emphasized that the desired privacy guarantees may not hold in settings where this assumption does not hold. We now formalize our personalized privacy definition.

Definition 6 (Personalized Differential Privacy (PDP))**.** *In the context of a privacy specification Φ and a universe of users $\mathcal{U}$, a randomized mechanism $\mathcal{M} : \mathcal{D} \to R$ satisfies* Φ-personalized differential privacy (or Φ-PDP)*, if for every pair of neighboring datasets $D, D' \subset \mathcal{D}$, with $D \overset{t}{\sim} D'$, and for all sets $O \subseteq R$ of possible outputs,*

$$\Pr[\mathcal{M}(D) \in O] \leq e^{\Phi^t_{\mathcal{U}}} \times \Pr[\mathcal{M}(D') \in O].$$

Intuitively, PDP offers the same strong, semantic notion of privacy that traditional differential privacy provides, but the privacy guarantee for PDP is personalized to the needs of every user simultaneously. As we will demonstrate later in the paper, PDP opens the door for attaining a higher level of utility when not all users require the same strong privacy level.

A. Properties of PDP

We start by formalizing the relationship between PDP and traditional differential privacy.

Theorem 3 (Differential Privacy Implies PDP)**.** *Let $\mathcal{U}$ denote a universe of users and let $\mathcal{D}$ denote the associated universe of tuples. Any mechanism $\mathcal{M} : \mathcal{D} \to R$ that satisfies ε-differential privacy also satisfies Φ-PDP, with privacy specification $\Phi = \{(u, \varepsilon) | u \in \mathcal{U}\}$.*

The proof follows immediately from the definitions of differential privacy (Def. 3) and PDP (Def. 6).

The ability to compose nicely is an important property for practical privacy definitions. The composition properties of traditional differential privacy extend naturally to PDP. For simplicity, our statement assumes that mechanisms operate on datasets with the same schema (i.e., they have the same attributes). The proof is deferred to the Appendix.

Theorem 4 (Composition)**.** *Let $\mathcal{M}_1 : \mathcal{D}_1 \to R$ and $\mathcal{M}_2 : \mathcal{D}_2 \to R$ denote two mechanisms that satisfy PDP for Φ_1 and Φ_2, respectively. Let $\mathcal{U}_1$ and $\mathcal{U}_2$ denote the associated universes of users. Finally, let $\mathcal{D}_3 = \mathcal{D}_1 \cup \mathcal{D}_2$. Then, for any $D \subset \mathcal{D}_3$, the mechanism $\mathcal{M}_3(D) = g(\mathcal{M}_1(D \cap \mathcal{D}_1), \mathcal{M}_2(D \cap \mathcal{D}_2))$ satisfies Φ_3-PDP, where $\Phi_3 = (\{(u, \Phi_1^u + \Phi_2^u) | u \in \mathcal{U}_1 \cap \mathcal{U}_2\} \cup \{(v, \Phi_1^v) | v \in \mathcal{U}_1 \setminus \mathcal{U}_2\} \cup \{(w, \Phi_2^w) | w \in \mathcal{U}_2 \setminus \mathcal{U}_1\})$, and g is an arbitrary function of the outputs of $\mathcal{M}_1$ and $\mathcal{M}_2$.*

In other words, the privacy afforded to a user degrades when multiple computations are run over the same data.

IV. Mechanisms for Achieving PDP

In this section, we present general mechanisms for achieving PDP for arbitrary functions. We begin by establishing some naïve baseline mechanisms which represent the limited options that an analyst when employing traditional differential privacy in the presence of non-uniform privacy preferences. In that sense, the baselines will allow us to compare PDP against traditional differential privacy.

A. Baseline Mechanisms

The naïve baseline mechanisms that we introduce now technically achieve PDP, but fail to take advantage of the personalized privacy preferences to benefit utility. In the rest of the section, we will use the notation $DP_\varepsilon^f(D)$ to denote any mechanism that computes the function f on an input D and satisfies the *traditional* ε-differential privacy definition. For example, if f is the *mean* function, DP_ε^f could be an instance of the Laplace mechanism; or, if f is the *median*, DP_ε^f

might be realized by an instance of the exponential mechanism. However, DP_ε^f could also be a more complex composition of multiple differentially private components.

The first baseline mechanism is simply a direct application of Theorem 3. That is, we find the strongest privacy preference in a given privacy specification (i.e., $\alpha = \min_u \Phi^u$) and then invoke DP_α^f using that as the global privacy parameter.

Definition 7 (Minimum)**.** *Given function $f : \mathcal{D} \to R$, dataset $D \subset \mathcal{D}$, and a privacy specification Φ, the Φ-PDP Minimum mechanism $M_f(D,\Phi)$ releases $DP_\alpha^f(D)$, where $\alpha = \min_u \Phi^u$.*

Proof: We need to show that for any arbitrary neighboring datasets $D \overset{t}{\sim} D' \subset \mathcal{D}$, and any $O \subseteq \text{Range}\big(M_f\big)$,

$$\Pr[M_f(D,\Phi) \in O] \leq e^{\Phi^t\mathcal{U}} \Pr[M_f(D',\Phi) \in O].$$

Since $DP_\alpha^f(D)$ satisfies α-differential privacy for $\alpha = \min_u \Phi^u$, it also satisfies Φ_α-PDP for $\Phi_\alpha = \{(u,\alpha)|u \in \mathcal{U}\}$ (by Thm. 3).

$$\begin{aligned}\text{Thus,} \qquad \Pr[M_f(D,\Phi) \in O] &\leq e^{\min_u \Phi^u} \Pr[M_f(D',\Phi) \in O] \\ &\leq e^{\Phi^t\mathcal{U}} \Pr[M_f(D',\Phi) \in O],\end{aligned}$$

as desired. ∎

Although *Minimum* satisfies PDP, it gains no benefit from the personalized privacy preferences; most users will receive a much stronger level of privacy than they require. If we have a dataset where there are relatively few very privacy conscious users and a larger set of less concerned users, another option is to simply discard all of the tuples belonging to the privacy conscious users. We would then add noise according to the strictest remaining user. This is the idea behind the next baseline mechanism.

Definition 8 (Threshold)**.** *Given function $f : \mathcal{D} \to R$, dataset $D \subset \mathcal{D}$, and a privacy specification Φ, the Φ-PDP Threshold mechanism $T_f(D,\Phi,t)$ first constructs from D the dataset $D_t = \{x \in D|\Phi^{x_\mathcal{U}} \geq t\}$ and then releases $DP_t^f(D_t)$.*

Proof: First, consider an arbitrary pair of neighboring datasets $D \overset{x}{\sim} D' \subset \mathcal{D}$, with $\Phi^x_\mathcal{U} < t$. In this case, the datasets D_t and D'_t, constructed from D and D' by removing all the tuples belonging to users with privacy settings below t, will be equivalent since x will be one of the tuples omitted from both D_t and D'_t. Then clearly, for any $O \subseteq \text{Range}(T_f)$, $\Pr[T_f(D,\Phi,t) \in O] = \Pr[T_f(D',\Phi,t) \in O]$, which satisfies definition 6.

Next we consider the case where $\Phi^x \geq t$. In this case, $D_t \overset{x}{\sim} D_t{}'$. Now, observe that $T_f(D,\Phi,t) = DP_t^f(D_t)$, which satisfies t-differential privacy, and therefore also satisfies Φ_t-PDP for $\Phi_t = \{(u,t)|u \in \mathcal{U}\}$ (by Thm. 3). Then, for any $O \subseteq \text{Range}(T_f)$,

$$\begin{aligned}\Pr[T_f(D,\Phi,t) \in O] &\leq e^t \Pr[T_f(D',\Phi,t) \in O] \\ &\leq e^{\Phi^x\mathcal{U}} \Pr[T_f(D',\Phi,t) \in O],\end{aligned}$$

as desired. ∎

B. Achieving PDP via Sampling

We now present a smarter general purpose mechanism for achieving PDP that in many cases is able to attain a higher level of utility than the baselines. The mechanism works by introducing two independent sources of randomness into a computation: (1) non-uniform random sampling at the tuple level, where the inclusion probability for a tuple depends on the personal privacy preference of the corresponding user (and a global threshold t), and (2) additional uniform randomness introduced by invoking a traditional differentially private mechanism on the sampled input, where the privacy parameter ε depends on t. Combined, the two sources of randomness yield the precise amount of privacy required by each tuple.

Definition 9 (The Sample Mechanism)**.** *Consider a function $f : \mathcal{D} \to R$, a dataset $D \subset \mathcal{D}$, and a privacy specification Φ. Let $RS(D,\Phi,t)$ denote the procedure that independently samples each tuple $x \in D$ with probability*

$$\pi_x = \begin{cases} \frac{e^{\Phi^x\mathcal{U}} - 1}{e^t - 1} & \text{if } \Phi^{x_\mathcal{U}} < t \\ 1 & \text{otherwise} \end{cases}$$

where $\min_u \Phi^u \leq t \leq \max_u \Phi^u$ is a configurable threshold. The Sample mechanism is defined as

$$S_f(D,\Phi,t) = DP_t^f\left(RS(D,\Phi,t)\right)$$

where DP_t^f is any t-differentially private mechanism that computes the function f.

Theorem 5. *The Sample mechanism S_f satisfies Φ-PDP.*

Proof: We will use the notation D_{-x} (or D_{+x}) to mean the dataset resulting from removing from (adding to) D the tuple x. Thus, we can represent two neighboring datasets as D and D_{-x}. We will show that for any D, D_{-x} and any $O \in \text{Range}(S_f)$,

$$\Pr[S_f(D,\Phi,t) \in O] \leq e^{\Phi^x} \Pr[S_f(D_{-x},\Phi,t) \in O].$$

Observe that all of the possible outputs of $RS(D,\Phi,t)$ can be divided into those in which x was selected, and those in which x was not selected. Thus, we can write $\Pr[S_f(D,\Phi,t) \in O]$ as

$$\begin{aligned}&\sum_{Z \subseteq D_{-x}} \big(\pi_x \Pr[RS(D_{-x},\Phi,t) = Z] \Pr[DP_t^f(Z_{+x}) \in O]\big) \\ &\quad + \sum_{Z \subseteq D_{-x}} \big((1-\pi_x) \Pr[RS(D_{-x},\Phi,t) = Z] \Pr[DP_t^f(Z) \in O]\big) \\ &= \sum_{Z \subseteq D_{-x}} \big(\pi_x \Pr[RS(D_{-x},\Phi,t) = Z] \Pr[DP_t^f(Z_{+x}) \in O]\big) \\ &\quad + (1-\pi_x) \Pr[S_f(D_{-x},\Phi,t) \in O].\end{aligned} \tag{1}$$

Since DP_t^f satisfies Φ_t-PDP, for $\Phi_t = \{(u,t)|u \in \mathcal{U}\}$ (Thm. 3),

$$\Pr[DP_t^f(Z_{+x}) \in O] \leq e^t \Pr[DP_t^f(Z) \in O].$$

Thus, equation (1) can be rewritten as

$$\begin{aligned}&\Pr[S_f(D,\Phi,t) \in O] \\ &\leq \sum_{Z \subseteq D_{-x}} \big(\pi_x \Pr[RS(D_{-x},\Phi,t) = Z](e^t \Pr[DP_t^f(Z) \in O])\big) \\ &\quad + (1-\pi_x) \Pr[S_f(D_{-x},\Phi,t) \in O] \\ &= \pi_x(e^t \Pr[S_f(D_{-x},\Phi,t) \in O]) + (1-\pi_x) \Pr[S_f(D_{-x},\Phi,t) \in O] \\ &= (1 - \pi_x + \pi_x e^t) \Pr[S_f(D_{-x},\Phi,t) \in O].\end{aligned} \tag{2}$$

In substituting for π_x in (2), there are two cases for x that we must consider: the case in which $\Phi^x \geq t$, and the case in which

$\Phi^x < t$. Let us consider the former case first. By definition, when $\Phi^x \geq t$, tuple x is selected with probability $\pi_x = 1$; thus substituting 1 for π_x in (2), we have

$$\begin{aligned}\Pr[S_f(D,\Phi,t) \in O] &\leq (1-\pi_x+\pi_x e^t)\Pr[S_f(D_{-x},\Phi,t) \in O] \\ &= e^t Pr[S_f(D_{-x},\Phi,t) \in O] \\ &\leq e^{\Phi^x} Pr[S_f(D_{-x},\Phi,t) \in O],\end{aligned}$$

as desired. Let us now consider the case in which $\Phi^x < t$. Expanding π_x in $(1-\pi_x+\pi_x e^t)$ in equation (2), we get:

$$\begin{aligned}(1-\pi_x+\pi_x e^t) &= 1-\tfrac{e^{\Phi^x}-1}{e^t-1}+\tfrac{e^{\Phi^x}-1}{e^t-1}e^t \\ &= \frac{e^t-e^{\Phi^x}+e^{\Phi^x+t}-e^t}{e^t-1} \\ &= \frac{-e^{\Phi^x}+e^{\Phi^x+t}}{e^t-1} = \frac{(e^t-1)e^{\Phi^x}}{e^t-1} = e^{\Phi^x}.\end{aligned}$$

Thus, we have $\Pr[S_f(D,\Phi,t) \in O] \leq e^{\Phi^x}\Pr[S_f(D_{-x},\Phi,t) \in O]$, and therefore $S_f(D,\Phi,t)$ satisfies Φ-PDP. ■

Remark. Our Sampling mechanism is inspired by a result from [14], where the authors observed that random sampling has a "privacy amplification" effect when combined with differential privacy. Further discussion of prior work related to sampling in the context of differential privacy is made in Section II-B.

Discussion. We make a few important observations regarding the Sample mechanism. First, we emphasize that the Sample mechanism is not limited to simple aggregates like counts, sums, etc. In fact, the Sample mechanism is immediately applicable to arbitrarily complex functions, so long as there is a known differentially private algorithm for computing f, i.e., DP^f. The mechanism DP^f could be a simple instantiation of the Laplace or exponential mechanisms, or a more complex composition of several differentially private mechanisms. The Sample mechanism essentially treats DP^f as a black box that operates on a dataset of tuples.

Second, we note that the Sample mechanism effectively introduces two types of randomness—and hence two types of error—into f. The threshold t optionally[1] provides a means of balancing these types of error. A small t results in fewer tuples being discarded by the sampling step (and lower sampling error), but results in more randomness (e.g., noise) due to DP_t^f. Observe that when $t = \max_u \Phi^u$, every tuple is provided with the precise amount of privacy it requires. When $t = \min_u \Phi^u$, the Sample mechanism collapses down to the Minimum baseline mechanism.

The tunable threshold is useful because the two types of error can impact the resulting output differently. Using $t = \max_u \Phi^u$, so that all users receive exactly the required amount of privacy, may not always give the best results; often, by using a lower threshold we can significantly reduce the sampling error while not introducing too much extra noise. As a concrete example, consider the *count* aggregate. In this case, we have $D \in \{0,1\}^n$, and $DP_t^f = f(D) + \text{Lap}(1/t)$, where $f = \text{count}(D)$ is the function that counts the number of non-zero tuples in D. Observe that the magnitude of the Laplace noise depends only on the sensitivity of f (which is 1 in this case) and t. Thus, the larger D is, the smaller the noise relative to the count. The error due to sampling, on the other hand, depends not only on the privacy specification, but also on the density of the data. Thus for sufficiently large datasets, setting a lower threshold (i.e., $t \ll \max_u \Phi^u$) could greatly increase the sample rate for the users with strong privacy requirements, at the cost of slightly more noise, but a lower total error. The following example illustrates.

Example 1. *For the count aggregate, suppose that we have a dataset D with $n = 200$ tuples, each corresponding to one user, with a selectivity value of 0.1. For simplicity, assume that users fall into either one of two groups, w.r.t. privacy preferences:* conservative *users have a strong privacy requirement, say $\varepsilon_C = 0.1$, and* liberal *users have a relatively weak requirement, say $\varepsilon_L = 1$. If we set $t = \varepsilon_L$, then each of the conservative tuples would be retained with probability $\pi_C = \frac{e^{\varepsilon_C}-1}{e^t-1} = \frac{e^{0.1}-1}{e^1-1} \approx 0.0612$. Let $\tilde{D} = RS(D,\Phi,t=\varepsilon_L)$ denote the sampled dataset. If we assume that a majority of the users, say 65%, are conservative, then the squared error due to sampling is calculated as*

$$\text{Err}(count(\tilde{D})) = \text{Var}(\text{count}(\tilde{D}) + \text{Bias}(\text{count}(\tilde{D}))^2 = (n \cdot 0.65 \cdot 0.1) \cdot \pi_C \cdot (1-\pi_C) + ((n \cdot 0.65 \cdot 0.1) \cdot (\pi_C - 1))^2 \approx 150,$$

while the (additional) error due to the Laplace noise injected by $DP_{t=1}^f$ is $\text{Var}(\text{Lap}(1/1)) = 2(1/1)^2 = 2$. However, observe that if we instead set $t = 0.2$, we get $\pi_C \approx 0.475$ and $\text{Err}(\text{count}(\tilde{D}))$ is reduced to ≈ 50, while the noise-related error increases to 50. Thus, the total squared error is reduced from 152 to 100.

More complex functions that have a relatively high sensitivity, but are robust to sampling, especially for larger datasets, will see less benefit from threshold optimization. Recall that global sensitivity is a worst case measure of the impact a single tuple can have on the output of a function; however, for many functions, while the global sensitivity may be quite high, the impact that most tuples will actually have on the output is relatively small. For such functions, the same level of sampling error buys a significantly greater reduction in error due to DP^f, as the sampling rate is independent of the sensitivity.

Precisely optimizing t for an arbitrary f may be non-trivial in practice because, although the error of DP_t^f may be quantified without knowledge of the dataset, the impact of sampling does depend on the input data. Therefore, care must be taken so as to not leak privacy through the tuning process. A possible option, in some cases, is to make use of old data that is no longer sensitive (or not as sensitive), and that comes from a similar distribution, to approximately optimize the threshold without violating privacy[2]. In other cases, it might be feasible to use a portion of the privacy budget of the more conservative users and estimate the required quantities from that subset of the data. We postpone an in-depth study of threshold optimization strategies for future work. In Section V, we will demonstrate that for many functions, the simple heuristics of setting $t = \max_u \Phi^u$ or $t = \frac{1}{|\mathcal{U}|}\sum_u \Phi^u$, often give good results on real data and privacy specifications.

[1] Simply setting $t = \max_u \Phi^u$ offers good results any many cases, as we will show later in our experimental study.

[2] The idea of using older data is commonly used to estimate parameters for differentially private systems (e.g., in the GUPT system [20]).

C. Direct Approach

In the previous section, we introduced a two-step mechanism that achieves PDP by a sampling step, followed by a standard differentially private mechanism. Next we develop a more direct approach for achieving PDP, analogous to the exponential mechanism [17] for differential privacy. Our approach can be applied easily to aggregates like counts, medians, min/max and others. We first review the exponential mechanism and show that the score functions used to instantiate it for many aggregate functions can be represented in a common general form. We then show how that general form can be made to satisfy PDP.

In this section we assume that a dataset contains tuples with a single *numeric* attribute $\mathcal{A}$. We use a slightly different definition of *neighboring datasets* to other sections to simplify the presentation. Here, we assume that D and D' differ only in the *value* of a tuple t (instead of in the *presence* of t)[3]. We also consider datasets that have different values for an arbitrary number of tuples and will use the notation $D \oplus D'$ to denote the set of tuples in which D and D' differ.

Given a function $f : \mathcal{D} \to R$, recall that the exponential mechanism $\mathcal{E}^s_\varepsilon(D)$ outputs $r \in \text{Range}(f)$ with probability proportional to $\exp(\frac{\varepsilon s(D,r)}{2\Delta_s})$, where $s(D,r)$ is a real-valued score function that outputs a higher score the better r is relative to $f(D)$, and Δ_s is the sensitivity of s. We observe that one form of score function can be used to instantiate $\mathcal{E}$ for many common aggregate functions. For brevity, we consider three exemplar aggregates: *count*, *median*, *min*. For the count function, $\mathcal{A}$ is a binary indicator; for the other functions we assume that the value is an integer in the range $[lo, hi]$. The general score function is:

$$s(D,r) = \max_{f(D')=r} -|D \oplus D'| \tag{3}$$

That is, the score is inversely related to the number of changes to D that would be required for r to become the true answer. It is easy to see that $\Delta_s = 1$ for all three of the exemplar aggregates. Next observe that $s(D,r)$ is maximized when $r = f(D)$ (that is, when r is the true answer), and the score becomes smaller (more negative) the further r is from the true answer. For example, suppose we have $D = \langle 3,5,6,9,11\rangle$. With respect to *min*, we have $s(D,2) = s(D,4) = s(D,5) = -1$, because we only need to change one element of D to make any of those the minimum value; making 11 the minimum would require changing four values, so $s(D,11) = -4$. Similarly, for *median* we have, $s(D,5) = s(D,9) = -1$ and $s(D,3) = s(D,10) = s(D,11) = -2$, since changing a single element could cause 5 or 9 to become the median, while making 3, 10 or 11 the median requires two changes.

We need to understand the structure of this function further to satisfy PDP. For any D, r, there may be many D' that maximize equation (3). For instance, in the example above, we can make 5 the median by changing either 6, 9 or 11 to any value that is ≤ 5. In traditional differential privacy it is sufficient to treat all such D' equivalently. However, in the context of PDP, where each element has its own privacy setting, it becomes necessary to make a distinction among the different D' that maximize (3) for a given r. To make the intuition more concrete, consider the privacy specification $\Phi = \langle 0.1, 1, 1, 0.5, 1\rangle$ corresponding to $D = \langle 3,5,6,9,11\rangle$ from the earlier example. According to the PDP definition, we need $\left|\frac{\Pr[\mathcal{E}^s(D)=5]}{\Pr[\mathcal{E}^s(D')=5]}\right| \leq e^{0.5}$ when D' is formed by modifying the 9 (e.g., $D' = \langle 3,4,5,6,11\rangle$), but we only require that the ratio is $\leq e^{1.0}$ when D' is formed by changing the 6 or the 11. However, the definition must hold regardless of what D' happens to be. That is, when computing the probability distribution for $\mathcal{E}^s(D)$, the probability for r must assume that D' could be *any* neighboring dataset. Thus, in the example above, the probability of outputting 5 must be based on the strongest privacy requirement among the elements 6, 9 and ll, i.e., 0.5.

By modifying the exponential mechanism using the general score function of equation (3), with weighting to incorporate the privacy specification Φ in place of the fixed ε, we arrive at the following PDP mechanism.

Definition 10 ($\mathcal{PE}$ Mechanism)**.** *Given a function $f : \mathcal{D} \to R$, an arbitrary input dataset $D \subset \mathcal{D}$, a privacy specification Φ, the mechanism $\mathcal{PE}^f_\Phi(D)$ outputs $r \in R$ with probability*

$$\Pr[\mathcal{PE}^f_\Phi(D) = r] = \frac{\exp\left(\frac{1}{2} d_f(D,r,\Phi)\right)}{\sum_{q\in R} \exp\left(\frac{1}{2} d_f(D,q,\Phi)\right)} \tag{4}$$

$$\text{where } d_f(D,r,\Phi) = \max_{f(D')=r} \sum_{i \in D \oplus D'} -\Phi^{iu} \tag{5}$$

It is easy to verify that for the special case where the privacy preferences in Φ are uniform, $\mathcal{PE}^f_\Phi$ reduces to an instance of the original exponential mechanism. We now prove that $\mathcal{PE}$ satisfies PDP. The proof (deferred to the Appendix) modifies that of the original exponential mechanism [17].

Theorem 6. *The $\mathcal{PE}$ mechanism satisfies Φ-PDP.*

Concrete Examples: We describe how d_f can be efficiently computed for our exemplar aggregates. In general, finding an efficient algorithm to compute d_f for an arbitrary f may be non-trivial.

Count: Let count : $\{0,1\}^n \to R$, $R = \{0,1,\ldots,n\}$, be the function that returns the number of 1's in the input. Consider an arbitrary input D for which $\text{count}(D) = x, x \leq n$. To compute $d_{\text{count}}(D,r,\Phi)$ for an arbitrary $r \in R$, there are three possible cases to consider: (1) when $r > x$, $d_{\text{count}}(D,r,\Phi)$ is the sum of the $r-x$ smallest privacy settings among all the 0 bits in D; (2) when $r < x$, $d_{\text{count}}(D,r,\Phi)$ is the sum of the $x-r$ smallest privacy settings corresponding to the 1 bits in D; (3) when $x = r$, $d_{\text{count}}(D,r,\Phi) = 0$. Note that this algorithm requires sorting the privacy specification Φ, although this need only be done once.

Median: Let $R = \{lo, lo+1, \ldots, hi\}$ for $lo, hi \in \mathbb{Z}$. For a *sorted* dataset $D \in R^n$, the median function $\text{med} : R^n \to R$ returns the element with rank $m = \lfloor n/2 \rfloor$ in D (for simplicity, we assume that $|D|$ is odd). For an arbitrary $r \in R$, let i denote the rank of r in D. To compute $d_{\text{med}}(D,r,\Phi)$, there are three cases to consider: (1) if $i < m$, then the D' that minimizes equation (5) is the one derived from D by changing the $m-i$ elements to the right of element i with the smallest privacy settings in Φ; thus $d_{\text{med}}(D,r,\Phi)$ is just zero minus the sum of those privacy settings. For example, if $D = \langle 3,5,6,9,11\rangle$ and $\Phi =$

[3]This alternate definition is used in the differential privacy literature when it simplifies the task at hand [8].

$\langle 0.1,1,1,0.5,1\rangle$, then we have $d_{\text{med}}(D,3,\Phi) = d_{\text{med}}(D,4,\Phi) = -1.5$ and $d_{\text{med}}(D,5,\Phi) = -0.5$, and so on. (2) Conversely, when $i > m$, we must consider all elements *to the left* of i in D. In this case, $d_{\text{med}}(D,r,\Phi)$ will be zero minus the sum of the $i-m$ smallest privacy parameters among those elements of D with rank $< i$. For example, considering the D,Φ given above, $d_{\text{med}}(D,11,\Phi) = -0.6$. (3) When $i = m$, $d_{\text{med}}(D,r,\Phi) = 0$. Compared to finding a median with differential privacy, the PDP implementation additionally needs to sort Φ, leading to only a slight increase in computational overhead.

Min: As before, let $R = \{lo, lo+1, \ldots, hi\}$ for $lo, hi \in \mathbb{Z}$. For a *sorted* dataset $D \in R^n$, the min function $\min : R^n \to R$ returns the smallest element of input D. For an arbitrary $r \in R$, there are again three cases: (1) when $r > \min(D)$, observe that for r to become the minimum, *all* elements $q \in D$ in which $q < r$ would have to be changed so that their values are $\geq r$. Thus $d_{\min}(D,r,\Phi)$ would be equal to the sum of the privacy settings of all elements in D with a value less than r. (2) When $r < \min(D)$, for r to become the minimum, the value of any single element in D could be changed to r; thus, $d_{\min}(D,r,\Phi)$ equals the minimum privacy setting among the elements in D. (3) When $r = \min(D)$, $d_{\min}(D,r,\Phi) = 0$.

For large R, we can improve efficiency for median and min by observing that all $r \in R$ that fall between two consecutive elements in D, say p and q, have the same rank as either p or q. Therefore, we can divide the output space into ranges and compute probabilities for each range (multiplied by the size of the range). When a range is selected by the mechanism, the returned value is sampled uniformly from within it.

V. Experimental Study

Next, we apply our PDP mechanisms to two common aggregate functions, *count* and *median*, as well as to the more complex task of *multiple linear regression*. Although count and median are relatively simple functions, they are important primitives for building more complex algorithms [5]. For the count and median functions, we compare the Sample mechanism and the exponential-like $\mathcal{PE}$ mechanism. For linear regression, we use the Sampling mechanism to transform a recent differentially private approach for linear regression, introduced by Zhang et al. [33], yielding a PDP version of the algorithm.

Our main goal for this experimental study is to demonstrate that by taking personal privacy preferences into account, our proposed PDP mechanisms can often attain more accurate data analysis results, compared to traditional differential privacy, which provides only a uniform privacy guarantee. To that end, we compare to the baseline mechanisms *Minimum* and *Threshold* (see Section IV-A), in terms of *root mean squared error* (RMSE) on real and synthetic data, under different data distributions and privacy specifications.

Datasets: We evaluate mechanisms for count and median on synthetic data. For count, we generate datasets with 1,000 records, each with a single binary attribute. The fraction of records with a value of '1' is controlled by a *density parameter*, δ, in the range (0,1) (default $\delta = 0.15$). For the median function we generate datasets with 1,001 records, where the attribute values are randomly drawn from a normal distribution with mean μ and standard deviation σ (defaults, $\mu = 500$, $\sigma = 200$), rounded to the nearest integer in the range $[1,1000]$. For the linear regression task, we use a dataset containing 100,000 records (representing 100,000 users) from the 2012 US Census [23], detailed in in Section V-B.

Privacy Specification: To generate the privacy specifications for our experiments, we randomly divided the users (records) into three groups: *conservative*, representing users with high privacy concern; *moderate*, representing users with medium concern; and *liberal*, representing users with low concern[4]. The fraction of users in the conservative and moderate groups were determined by the parameters f_C and f_M; the fraction of users in the liberal group is $1.0-(f_C+f_M)$. The default values used in our experiments were $f_C = 0.54$ and $f_M = 0.37$ and were chosen based on findings reported in [2] in the context of a user survey about privacy concern. The privacy preferences for the users in the conservative and moderate groups were drawn uniformly at random from the ranges $[\varepsilon_C, \varepsilon_M]$ and $[\varepsilon_M, \varepsilon_L]$, respectively (and rounding to the nearest hundredth), with $\varepsilon_C, \varepsilon_M, \varepsilon_L \in [0.01, 1.0]$; the users in the liberal group received a privacy preference of ε_L, which was fixed at $\varepsilon_L = 1.0$ for all of the experiments in this paper. The defaults for the other two parameters were $\varepsilon_C = 0.01$ and $\varepsilon_M = 0.2$, where a *smaller* value yields *greater* privacy. Table I lists the various parameters used in our experiments and the ranges of values we tested for each, with the default values underlined.

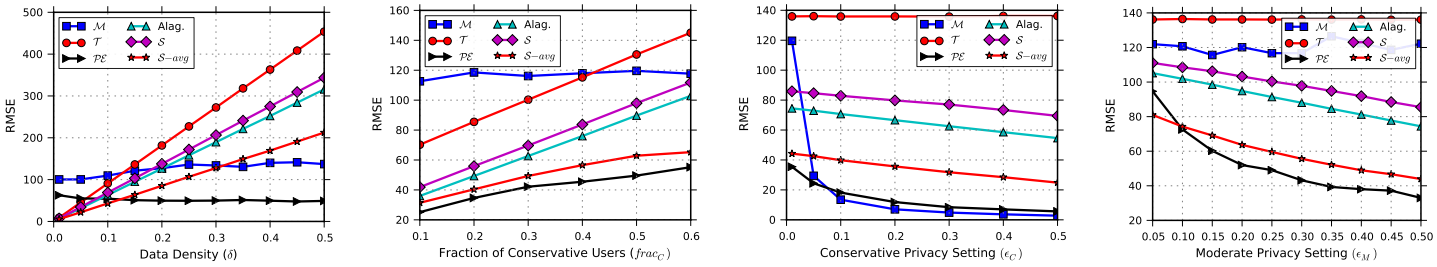
TABLE I. Experiment parameters (defaults underlined).

n (count, median)	1000 (count); 1001 (med.)
n (lin. regression)	$10000, 20000, \ldots, \underline{100000}$
δ (count only)	$0.01, 0.05, 0.1, \underline{0.15}, \ldots, 0.5$
ε_C	$\underline{0.01}, 0.05, 0.1, 0.2, \ldots, 0.5$
ε_M	$0.05, 0.1, 0.15, \underline{0.2}, \ldots, 0.5$
ε_L	1.0
σ (median only)	$100, \underline{200}, \ldots, 1000$
μ (median only)	500
f_C	$0.1, 0.2, \ldots, 0.6$; $\underline{0.54}$
f_M	0.37
f_L	$1.0-(f_C+f_M)$

A. PDP for Count and Median

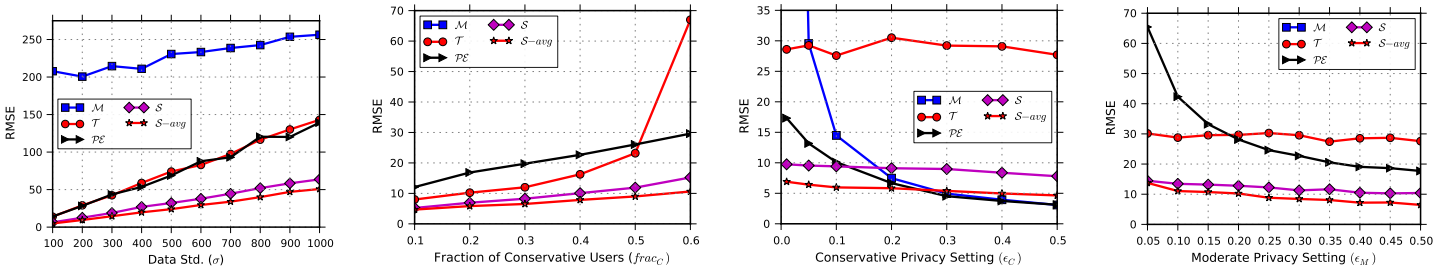
In this section we apply our two PDP mechanisms to the *count* and *median* functions. We compared the RMSE of four main approaches: the Minimum and Threshold baselines ($\mathcal{M}$ and $\mathcal{T}$, respectively), the Sampling mechanism with threshold $t = \max_u \Phi^u = \varepsilon_L$ (denoted $\mathcal{S}$), and the exponential-like $\mathcal{PE}$ mechanism. Recall that $\mathcal{M}$ invokes a standard differentially private mechanism (the Laplace mechanism for count and the exponential mechanism for median), using ε_C as the privacy parameter. $\mathcal{T}$ works by first discarding all but the liberal user data and then invoking a differentially private mechanism with ε_L as the privacy parameter. Additionally, we investigated a variation of $\mathcal{S}$ with the heuristic of setting the sampling threshold to $t = \frac{1}{n}\sum_u \Phi^u$ (i.e., the average privacy setting), as suggested in Section IV-B; we denote this approach $\mathcal{S}$-avg. Finally, for the count task, we also considered the Stretching mechanism introduced by Alaggan et al. [3] in the context

[4]The choice to partition users into low/medium/high privacy groups was based on findings from several studies by other researchers regarding user privacy attitudes (e.g., [4], [2], [1]).



(a) Impact of δ (b) Impact of f_C (c) Impact of ε_C (with $\varepsilon_M = 0.5$) (d) Impact of ε_M

Fig. 1. (*Count*) RMSE of each mechanism for the *count* task, as four parameters are varied.



(a) Impact of σ (b) Impact of f_C (c) Impact of ε_C (with $\varepsilon_M = 0.5$) (d) Impact of ε_M

Fig. 2. (*Median*) RMSE of each mechanism in for the *median* task, as four parameters are varied.

of their similar privacy model, HDP (it does not apply for the median task). Translated to our framework, the Stretching mechanism works by (1) multiplying the data value of each tuple i by a scaling factor $\frac{\varepsilon_i}{\varepsilon_L}$, where ε_i is the privacy setting for tuple i, and (2) releasing $f(D') + \text{Lap}(1/\varepsilon_L)$, where D' is the scaled dataset. We write *Alag.* to denote this approach in the results.

For a given configuration of the parameters in Table I, we computed the RMSE for each mechansim (between the private count/median and the true count/median) over 1,000 runs, using a different randomly generated dataset and privacy specification for each run.

Impact of Data Distribution. We first examine the impact of the data density δ (i.e., the fraction of 1's in the input data) on the *count* function. Figure 1(a) shows the RMSE for count as a function of increasing δ. The results indicate that when density is low (e.g., $\delta < 0.1$), $\mathcal{S}$ and $\mathcal{S}$-avg offer the lowest error, since most of the discarded tuples have a value of zero and the count is only affected when a 1 is discarded. However, we see that in general counts are highly sensitive to sampling, and as the density increases, the error of the sampling-based approaches quickly exceeds that of the $\mathcal{M}$ baseline. For denser datasets (e.g., $\delta > 0.1$), $\mathcal{PE}$ is the clear winner, outperforming all other approaches by a significant margin. For example, when $\delta = 0.3$, the error of $\mathcal{PE}$ is less than half that of the next best mechanism. An important observation is that, unlike the sampling-based mechanisms, $\mathcal{PE}$ is able to make use of all of the data in the input and, like the original exponential mechanism, is generally unaffected by the data density. This also means that, as the density (or the size of the dataset) increases, the *relative error* will tend toward zero, yielding highly accurate counts. For *Alag.* we observed slightly lower error than for $\mathcal{S}$; however, *Alag.* was still significantly outperformed by $\mathcal{PE}$. Finally, we observed that the simple threshold heuristic employed by $\mathcal{S}$-avg works surprisingly well here, offering a significant reduction in error compared to $\mathcal{S}$ (which uses $t = \max_u \Phi^u$).

For median, we vary the standard deviation σ. Figure 2(a), shows that as σ increases, and the data become more spread out, the errors of all approaches increase. When σ is small, there are many values concentrated around the median and the output is therefore less affected by individual values that are discarded due to sampling. Likewise, for $\mathcal{M}$ and $\mathcal{PE}$, when the values are concentrated around the median, most of the probability mass will be concentrated on a small range of output values close to the true median. Compared to the count function, the median is far more resistant to sampling; consequently $\mathcal{S}$ and $\mathcal{S}$-avg offer a considerable reduction in error, relative to the baseline approaches. For the same reason, $\mathcal{T}$ fares much better in this task, compared to count; however, it is still significantly outperformed by the $\mathcal{S}$. Although $\mathcal{PE}$ significantly outperforms $\mathcal{M}$, it appears to be no better than $\mathcal{T}$ in this scenario. Finally, we note that, in contrast to the count task, $\mathcal{S}$-avg offers only a slight improvement over using the default threshold; again, this appears to be a consequence of the median's stronger resistance to sampling.

Varying the Privacy Specification: In the previous experiments, a majority of the users (54%) were assumed to be in the conservative group; that is, most of the data records were assigned privacy preferences in the range $[\varepsilon_C, \varepsilon_M]$, while relatively few were assigned privacy preferences equal to ε_L. In cases where a larger fraction of the users are liberal (i.e., when f_C is small) we would expect the PDP mechanisms to perform even better. We confirmed this by varying f_C, while

keeping the other parameters at their defaults. Note that for each setting of f_C, the fraction of liberal users is equal to 1 - ($f_C + f_M$), so decreasing f_C increases the number of liberal users. The results are shown in Figures 1(b) (for count) and 2(b) (for median). The sharp increase in the error of $\mathcal{T}$ when $f_C = 0.6$ (Figure 2(b)) is when the median is being computed over only 3% of the dataset (30 records). Note that $\mathcal{M}$ does not depend on the f_C parameter, which is why its RMSE appears unaffected.

Next, we vary ε_C, which controls the lower bound on the range of privacy settings generated for the conservative users. Figures 1(c) and 2(c) show the results for count and median, respectively. Note that we used $\varepsilon_M = 0.5$ (rather than the default of $\varepsilon_M = 0.2$) for this experiment, to ensure $\varepsilon_C \leq \varepsilon_M$ in all cases. The key observation here is that the benefits of PDP, in terms of reduced error, diminish as the privacy requirements of the conservative users become weaker (closer to ε_L). This is because the error for the $\mathcal{M}$ decreases exponentially with increasing ε_C, while the reduction in error for the PDP mechanisms is much more subtle. For the count function, we see that when ε_C is larger than about 0.08, $\mathcal{M}$ becomes the best choice. For median, on the other hand, $\mathcal{S}$-avg remains the best choice until ε_C is larger than about 0.25, at which point the benefits of using PDP diminish. We note that the ε_C parameter is not used by $\mathcal{T}$, so observed variations are from the independent repetitions.

We now look at the impact of varying ε_M. Recall that increasing ε_M has the effect of raising the upper (lower) bound on the range of conservative (moderate) privacy settings. Therefore, we would expect the error for the PDP approaches to be smaller with a higher ε_M. Figures 1(d) and 2(d) show the results for count and median, respectively. For the sampling-based approaches, the error reduction was much more pronounced for count than for median, due to count's considerably lower resistance to sampling. The $\mathcal{PE}$ mechanism benefited greatly, with respect to both tasks, from the larger number of users with weaker privacy requirements.

B. PDP for Multiple Linear Regression

In this section, we demonstrate how the Sampling mechanism can be easily used to convert an existing differentially private algorithm into one that satisfies PDP. In particular, we focus on the task of linear regression, where the objective is to learn a linear model that can be used to predict the value of a response variable y from one or more predictor variables A. That is, given training dataset D_T, with rows of the form $A_{i1}, \ldots, A_{ik}, y_i$, we wish to learn a model $y = A \times w + b$, where the parameter vector w and the intercept term b are the outputs of the training process. For the experiments in this section, our goal was to (privately) learn a model to accurately predict an individual's income based on a set of other attributes (e.g., age, gender, number of children, etc.).

To do so, we adapt a differentially private linear regression algorithm, due to Zhang et al. [33], to satisfy PDP. The approach perturbs the coefficients of the objective function with Laplace noise, and then optimizes the perturbed objective function. The sensitivity—and hence the scale of the Laplace noise—depends on the number of attributes in the dataset. The algorithm can be easily extended to satisfy PDP by applying it in Definition 9. In other words, we choose a threshold t, sample the tuples in the input according to their privacy preferences and t, and then pass the sampled data directly into the differentially private algorithm described in [33], using t as the privacy parameter. We modified a publicly available Matlab implementation[5] of the original algorithm.

Dataset: We used a random sample of the 2012 US Census data from the Integrated Public Use Microdata Series [23] comprised of 100,000 records, each representing a unique individual living in the US. The dataset contained 12 attributes (five nominal and seven numeric): *receivesFoodstamps, gender, maritalStatus, employmentStatus, ownsHouse, nBedrooms, nVehicles, nChildren, age, timesMarried, nHoursWorked*, and *income*. We restricted the sample to only those individuals with a positive income. Of the nominal attributes, only *maritalStatus* had more than two values (i.e., married, single, divorced/widowed). Following [33], and others, we replace this attribute with two binary attributes *isMarried* and *isSingle*. Thus, the final dataset contained 13 attributes.

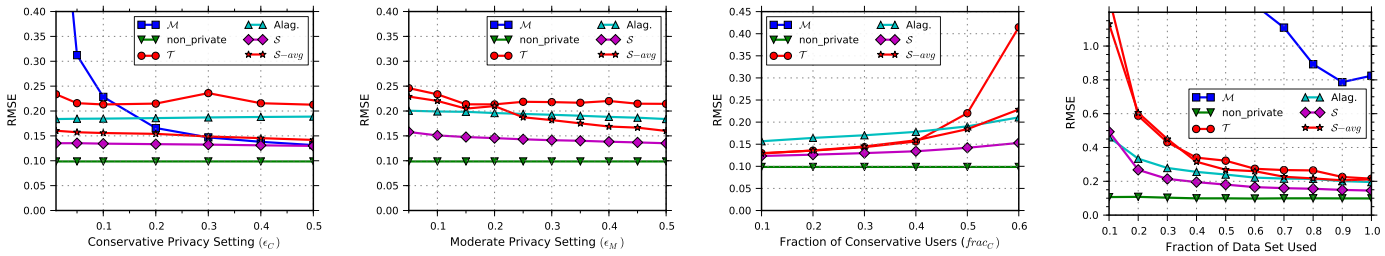
Experiment Setup: Our task is to model the income attribute based on the other attributes. We compared $\mathcal{S}$ and $\mathcal{S}$-avg to the two baseline mechanisms, $\mathcal{M}$ and $\mathcal{T}$ as well as the Stretching mechanism, *Alag.* [3]. Note that $\mathcal{PE}$ is not applicable to this task. For each experiment, we performed 500 runs of five-fold cross-validation, using a different randomly generated privacy specification for each run. We computed the RMSE of each approach over the 500 runs. As a preprocessing step, the linear regression implementation normalizes all attributes to the range $[-1, 1]$; thus, the reported errors are interpreted relative to that range.

Results: As we did for count and median, we vary ε_C, ε_M and f_C to obtain different privacy specifications (Figures 3(a), 3(b) and 3(c), respectively). We also investigated the impact of the dataset cardinality n (Figure 3(d)) by running the mechanisms (with default parameters) on different sized random subsets of the main dataset. We also plot the error of the non-private linear regression algorithm (denoted *non_private* in the plots).

Looking at Figure 3(a), we see that $\mathcal{S}$ significantly outperformed the baselines for $\varepsilon_C < 0.3$. In contrast to the previous experiments, the threshold heuristic $\mathcal{S}$-avg, performs less well here. This can be explained by the fact that the linear regression algorithm has a much higher sensitivity than count and median; consequently the effect of the Laplace noise is far more significant than the effect of sampling (up to a point), and so trading more noise for a higher sampling rate turns out to be a bad strategy. In contrast to what was seen for count, $\mathcal{S}$ significantly outperformed the HDP mechanism *Alag.* This is perhaps not too surprising since, intuitively, rescaling the values causes them to lose much of their meaning.

The observations regarding the impact of ε_M (Figure 3(b)) and f_C (Figure 3(c)) are similar to those for the median task. As ε_M gets larger, the average privacy preference for the conservative and moderate users increases (i.e., becomes weaker), leading to fewer records being discarded due to sampling, and consequently a lower RMSE. When f_C is small (i.e., there are fewer conservative users and more liberal users), the error for the sampling-based mechanisms is lower, but the

[5] http://sourceforge.net/projects/functionalmecha/



(a) Impact of ε_C (with $\varepsilon_M = 0.5$) (b) Impact of ε_M (c) Impact of f_C (d) Impact of n

Fig. 3. (*Linear Regression*) RMSE of each mechanism for *linear regression*, as four parameters are varied.

improvement relative to $\mathcal{T}$ is also quite small. On the other hand, when the conservative users make up half of the users in the dataset, the RMSE of $\mathcal{S}$ is about 36% lower than that of $\mathcal{T}$. We also note the spike in error for $\mathcal{T}$ when f_C approaches 0.6, which we also observed for the median task (Figure 2(b)); again, this appears to be due to the fact that the input to $\mathcal{T}$ is only about 3% of the total records in that case. Since $\mathcal{M}$ does not take the parameter ε_M or f_C, we chose to focus on the other mechanisms in the respective plots. We note that the RMSE for $\mathcal{M}$ was > 0.77 and generally unstable, due to the high noise variance associated with a privacy setting of 0.01.

Finally, Figure 3(d) shows that the size of the dataset n has a big impact on all approaches, as expected. The larger the dataset, the more data that is left after sampling and the better the PDP approaches perform. Moreover, since the sensitivity of linear regression is independent of n, the signal to noise ratio improves with increasing input size. In Fig. 3(d), the RMSE of $\mathcal{M}$ remained above 1.4 until $n \geq 40{,}000$.

VI. Conclusions and Future Work

We have introduced PDP, a personalized privacy framework that combines the strength of differential privacy with the added flexibility of user-specific privacy guarantees. Mechanisms based on non-uniform sampling and extensions of the exponential mechanism can achieve PDP effectively and efficiently.

There are many avenues for future work. With respect to the Sampling mechanism, although we have shown that simple threshold heuristics work well in practice, it is likely that the error could be further reduced through a more careful tuning of the threshold for specific tasks. As with differential privacy, although the exponential mechanism is quite general, getting the best results require a careful choice of quality function, and the use of the seemingly "obvious" quality function can be beaten by tailored approaches in terms of accuracy and scalability. It will also be of interest to extend notions of personalized privacy to social networks, where the individuals are nodes, and edges represent connections between pairs.

Acknowledgment

The authors are grateful to the anonymous reviewers for their helpful feedback. This work is supported in part by the National Science Foundation under the awards CNS-0747247 and CNS-1314229, an NSA Science of Security Lablet grant at North Carolina State University, and European Commission Marie Curie Integration Grant PCIG13-GA-2013-618202.

References

[1] M. S. Ackerman, L. F. Cranor, and J. Reagle. Privacy in e-commerce: examining user scenarios and privacy preferences. In *ACM EC*, pages 1–8. 1999.

[2] A. Acquisti and J. Grossklags. Privacy and rationality in individual decision making. *IEEE Security & Privacy*, 2:24–30, 2005.

[3] M. Alaggan, S. Gambs, and A.-M. Kermarrec. Heterogeneous differential privacy. www.helwan.edu.eg/university/staff/Dr.MohamedNabil/hdp.pdf 2014.

[4] B. Berendt, O. Günther, and S. Spiekermann. Privacy in e-commerce: stated preferences vs. actual behavior. volume 48, pages 101–106. CACM, 2005.

[5] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the sulq framework. In *ACM PODS*, pages 128–138. 2005.

[6] C. Dwork. Differential privacy. In *ICALP*, pages 1–12, 2006.

[7] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, pages 486–503. 2006.

[8] C. Dwork and J. Lei. Differential privacy and robust statistics. In *ACM STOC*, pages 371–380. 2009.

[9] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284. 2006.

[10] J. Gehrke, M. Hay, E. Lui, and R. Pass. Crowd-blending privacy. In *CRYPTO*, pages 479–496. 2012.

[11] S. Kasiviswanathan, H. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *IEEE FOCS*, pages 531–540. 2008.

[12] G. Kellaris and S. Papadopoulos. Practical differential privacy via grouping and smoothing. In *PVLDB*, volume 6, 2013.

[13] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE ICDE*, pages 106–115, 2007.

[14] N. Li, W. Qardaji, and D. Su. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *ASIACCS*, pages 32–33. 2012.

[15] B.-R. Lin, Y. Wang, and S. Rane. On the benefits of sampling in privacy preserving statistical analysis on distributed databases. *arXiv:1304.4613*, 2013.

[16] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM TKDD*, 1(1):3, 2007.

[17] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *IEEE FOCS*, pages 94–103. 2007.

[18] S. J. Milberg, H. J. Smith, and S. J. Burke. Information privacy: Corporate management and national regulation. *Organization Science*, 11(1):35–57, 2000.

[19] G. R. Milne and A. J. Rohm. Consumer privacy and name removal across direct marketing channels: exploring opt-in and opt-out alternatives. *Journal of Public Policy & Marketing*, 19(2):238–249, 2000.

[20] P. Mohan, A. Thakurta, E. Shi, D. Song, and D. Culler. Gupt: privacy preserving data analysis made easy. In *ACM SIGMOD*, pages 349–360. 2012.

[21] M. M. Pai, and A. Roth Privacy and mechanism design. *ACM SIGecom Exchanges*, 12(1):8–29. 2013.

[22] Y. Shen, H. Shao, and Y. Li. Research on the personalized privacy preserving distributed data mining. In *IEEE FITME*, pages 436–439. 2009.

[23] S. Ruggles, J. T. Alexander, K. Genadek, R. Goeken, M. B. Schroeder, and M. Sobek Integrated public use microdata series: Version 5.0. *Minneapolis: University of Minnesota*, 2010.

[24] K. A. Stewart and A. H. Segars. An empirical examination of the concern for information privacy instrument. *Inf. Syst. Res.*, 13(1):36–49, 2002.

[25] E. F. Stone, H. G. Gueutal, D. G. Gardner, and S. McClure. A field experiment comparing information-privacy values, beliefs, and attitudes across several types of organizations. *J. Appl. Psychol.*, 68(3):459, 1983.

[26] L. Sweeney. k-anonymity: A model for protecting privacy. *IJUFKS*, 10(5):557–570, 2002.

[27] Y. Tao and X. Xiao. Personalized privacy preservation. In C. C. Aggarwal and P. S. Yu, editors, *Privacy-Preserving Data Mining*, pages 461–485. 2008.

[28] B. Wang and J. Yang. Personalized (α, k)-anonymity algorithm based on entropy classification. *JCIS*, 8(1):259–266, 2012.

[29] P. Wang. Personalized anonymity algorithm using clustering techniques. *JCIS*, 7(3):924–931, 2011.

[30] A. F. Westin. Privacy and freedom. *Wash. & Lee L. Rev.*, 25(1):166, 1968.

[31] X. Ye, Y. Zhang, and M. Liu. A personalized (a, k)-anonymity model. In *IEEE WAIM*, pages 341–348. 2008.

[32] M. Yuan, L. Chen, and P. S. Yu. Personalized privacy protection in social networks. In *PVLDB*, pages 141–150, 2010.

[33] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett. Functional mechanism: regression analysis under differential privacy. In *PVLDB*, pages 1364–1375. 2012.

APPENDIX

Theorem 6. *The $\mathcal{PE}$ mechanism satisfies Φ-PDP.*

Proof: Let $D \overset{x}{\sim} D' \subset \mathcal{D}$ be two arbitrary, neighboring datasets that differ in the value for tuple x. In the following, let $\varepsilon_x = \Phi^{x_\mathcal{U}}$, the privacy requirement for tuple x. First, observe that for any D and r, if $d_f(D,r,\Phi) = y$, then there is a dataset X such that $f(X) = r$ and $\sum_{i \in D \oplus X} \Phi^{i_\mathcal{U}} = -y$ (by Eq. 5). Then, since $D \oplus D' = \{x\}$, it follows that

$$\sum_{i \in D' \oplus X} \Phi^{i_\mathcal{U}} \geq \sum_{i \in D \oplus X} \Phi^{i_\mathcal{U}} - \sum_{i \in D' \oplus D} \Phi^{i_\mathcal{U}} \geq -y - \varepsilon_x.$$

Since $f(X) = r$, it follows that

$$d_f\left(D', r, \Phi\right) \leq y + \varepsilon_x = d_f(D,r,\Phi) + \varepsilon_x \tag{6}$$

We will use this fact below. To prove that $\mathcal{PE}$ satisfies Φ-PDP, we need to show that,

$$\frac{\Pr[\mathcal{PE}^f_\Phi(D) = r]}{\Pr[\mathcal{PE}^f_\Phi(D') = r]} \leq \exp(\varepsilon_x).$$

By equation (4), we have

$$\frac{\Pr[\mathcal{PE}^f_\Phi(D) = r]}{\Pr[\mathcal{PE}^f_\Phi(D') = r]} = \frac{\left(\frac{\exp\left(\frac{d_f(D,r,\Phi)}{2}\right)}{\sum_{q \in R} \exp\left(\frac{d_f(D,q,\Phi)}{2}\right)}\right)}{\left(\frac{\exp\left(\frac{d_f(D',r,\Phi)}{2}\right)}{\sum_{q \in R} \exp\left(\frac{d_f(D',q,\Phi)}{2}\right)}\right)}$$

$$= \left(\frac{\exp\left(\frac{d_f(D,r,\Phi)}{2}\right)}{\exp\left(\frac{d_f(D',r,\Phi)}{2}\right)}\right)\left(\frac{\sum_{q \in R} \exp\left(\frac{d_f(D',q,\Phi)}{2}\right)}{\sum_{q \in R} \exp\left(\frac{d_f(D,q,\Phi)}{2}\right)}\right) = A \cdot B$$

Rewriting A using equation (6), we have

$$A = \frac{\exp\left(\frac{d_f(D,r,\Phi)}{2}\right)}{\exp\left(\frac{d_f(D',r,\Phi)}{2}\right)} = \exp\left(\frac{d_f(D,r,\Phi) - d_f\left(D',r,\Phi\right)}{2}\right) \leq \exp\left(\frac{\varepsilon_x}{2}\right)$$

Similarly, rewriting B we get

$$B = \frac{\sum_{q \in R} \exp\left(\frac{d_f(D',q,\Phi)}{2}\right)}{\sum_{q \in R} \exp\left(\frac{d_f(D,q,\Phi)}{2}\right)} \leq \frac{\sum_{q \in R} \exp\left(\frac{d_f(D,q,\Phi)+\varepsilon_x}{2}\right)}{\sum_{q \in R} \exp\left(\frac{d_f(D,q,\Phi)}{2}\right)}$$

$$= \frac{\exp\left(\frac{\varepsilon_x}{2}\right) \sum_{q \in R} \exp\left(\frac{d_f(D,q,\Phi)}{2}\right)}{\sum_{q \in R} \exp\left(\frac{d_f(D,q,\Phi)}{2}\right)} = \exp\left(\frac{\varepsilon_x}{2}\right)$$

Thus, we have $A \cdot B \leq \exp\left(\frac{\varepsilon_x}{2} + \frac{\varepsilon_x}{2}\right) = \exp(\varepsilon_x)$. It can be similarly shown that the ratio is also $\geq \exp(-\varepsilon_x)$. Therefore, the $\mathcal{PE}$ mechanism satisfies Φ-PDP. ■

Next, we prove the composition theorem (Theorem 4) stated in Section III-A.

Theorem 4 (Composition)**.** *Let $\mathcal{M}_1 : \mathcal{D}_1 \to R$ and $\mathcal{M}_2 : \mathcal{D}_2 \to R$ denote two mechanisms that satisfy PDP for Φ_1 and Φ_2, respectively. Let $\mathcal{U}_1$ and $\mathcal{U}_2$ denote the associated universes of users. Finally, let $\mathcal{D}_3 = \mathcal{D}_1 \cup \mathcal{D}_2$. Then, for any $D \subset \mathcal{D}_3$, the mechanism $\mathcal{M}_3(D) = g(\mathcal{M}_1(D \cap \mathcal{D}_1), \mathcal{M}_2(D \cap \mathcal{D}_2))$ satisfies Φ_3-PDP, where $\Phi_3 = (\{(u, \Phi_1^u + \Phi_2^u) | u \in \mathcal{U}_1 \cap \mathcal{U}_2\} \cup \{(v, \Phi_1^v) | v \in \mathcal{U}_1 \setminus \mathcal{U}_2\} \cup \{(w, \Phi_2^w) | w \in \mathcal{U}_2 \setminus \mathcal{U}_1\})$, and g is an arbitrary function of the outputs of $\mathcal{M}_1$ and $\mathcal{M}_2$.*

Proof: Let $D \overset{t}{\sim} D'$, with $D, D' \subset \mathcal{D}_{1,2}$ be an arbitrary pair of neighboring datasets. First, let us consider the case where $t_\mathcal{U} \in \mathcal{U}_1 \cap \mathcal{U}_2$. We have that $t \in D' \cap \mathcal{D}_1$, and $t \in D' \cap \mathcal{D}_2$. To simplify notation a bit, let $\varepsilon_1 = \Phi_1^{t_\mathcal{U}}$ and $\varepsilon_2 = \Phi_2^{t_\mathcal{U}}$. For any $O \subseteq \text{Range}(\mathcal{M}_3)$, we can write

$$\Pr[M_3(D) \in O] = \sum_{(r_1,r_2) \in O} \Pr[\mathcal{M}_1(D \cap \mathcal{D}_1) = r_1] \cdot \Pr[\mathcal{M}_2(D \cap \mathcal{D}_2) = r_2].$$

Applying Def. 6 for both A_1 and A_2, we have that for any $O \subseteq \text{Range}(A_3)$,

$$\begin{aligned}\Pr[A_3(D) \in O] &\leq \sum_{(r_1,r_2) \in O} \left(e^{\varepsilon_1} \Pr[A_1(D') = r_1]\right)\left(e^{\varepsilon_2} \Pr[A_2(D') = r_2]\right) \\ &= e^{(\varepsilon_1+\varepsilon_2)} \sum_{(r_1,r_2) \in O} \Pr[A_1(D') = r_1] \Pr[A_2(D') = r_2] \\ &= e^{(\varepsilon_1+\varepsilon_2)} \Pr[A_3(D') \in O]\end{aligned}$$

Thus, $\Phi_3^t = (\varepsilon_1 + \varepsilon_2) = (\Phi_1^t + \Phi_2^t)$ as claimed. Next we consider the case in which $t_\mathcal{U} \in \mathcal{U}_1 \setminus \mathcal{U}_2$. Observe that $\Pr[A_2(D) = r_2] = \Pr[A_2(D') = r_2]$, since $D \cap \mathcal{U}_2 = D' \cap \mathcal{U}_2$. Thus, we have

$$\begin{aligned}\Pr[A_3(D) \in O] &\leq \sum_{(r_1,r_2) \in O} \left(e^{\varepsilon_1} \Pr[A_1(D') = r_1]\right) \Pr[A_2(D') = r_2] \\ &= e^{\varepsilon_1} \sum_{(r_1,r_2) \in O} \Pr[A_1(D') = r_1] \Pr[A_2(D') = r_2] \\ &= e^{\varepsilon_1} \Pr[A_3(D') \in O]\end{aligned}$$

and $\Phi_3^t = \varepsilon_1 = \Phi_1^t$, as claimed. An analogous argument can be made for the case in which $t_\mathcal{U} \in \mathcal{U}_2 \setminus \mathcal{U}_1$. ■