Dependence Makes You Vulnerable: Differential Privacy Under Dependent Tuples

Changchang Liu[1], Supriyo Chakraborty[2], Prateek Mittal[1]
Email: cl12@princeton.edu, supriyo@us.ibm.com, pmittal@princeton.edu
[1]Department of Electrical Engineering, Princeton University
[2]IBM T.J. Watson Research Center

***Abstract*—Differential privacy (DP) is a widely accepted mathematical framework for protecting data privacy. Simply stated, it guarantees that the distribution of query results changes only slightly due to the modification of any one tuple in the database. This allows protection, even against powerful adversaries, who know the entire database except one tuple. For providing this guarantee, differential privacy mechanisms assume independence of tuples in the database – a vulnerable assumption that can lead to degradation in expected privacy levels especially when applied to real-world datasets that manifest natural dependence owing to various social, behavioral, and genetic relationships between users. In this paper, we make several contributions that not only demonstrate the feasibility of exploiting the above vulnerability but also provide steps towards mitigating it. First, we present an inference attack, using real datasets, where an adversary leverages the probabilistic dependence between tuples to extract users' sensitive information from differentially private query results (violating the DP guarantees). Second, we introduce the notion of dependent differential privacy (DDP) that accounts for the dependence that exists between tuples and propose a dependent perturbation mechanism (DPM) to achieve the privacy guarantees in DDP. Finally, using a combination of theoretical analysis and extensive experiments involving different classes of queries (e.g., machine learning queries, graph queries) issued over multiple large-scale real-world datasets, we show that our DPM consistently outperforms state-of-the-art approaches in managing the privacy-utility tradeoffs for dependent data.**

I. Introduction

Information sharing is key to realizing the vision of a data-driven customization of our environment. Data that were earlier locked up in private repositories are now being increasingly shared for enabling new context-aware applications, better monitoring of population statistics, and facilitating academic research in diverse fields. However, sharing personal data gives rise to serious privacy concerns as the data can contain sensitive information that a user might want to keep private. Thus, while on one hand, it is imperative to release utility-providing information, on the other hand, the privacy of users whose data is being shared also needs to be protected. Towards this end, the notion of Differential Privacy (DP), which provides a rigorous mathematical foundation for defining and preserving privacy, has received considerable attention [12]–[16]. Used for protecting the privacy of aggregate query results over statistical databases, DP guarantees that the distribution of query outputs changes only slightly with the modification of a single tuple in the database. Thus, the information that an adversary can infer through observing the query output is strictly bounded by a function of the privacy budget.

To provide its guarantees, DP mechanisms assume that the data tuples (or records) in the database, each from a different user, are all independent. This is a weak assumption, especially because tuple dependence occurs naturally in datasets due to social, behavioral and genetic interactions between users. For example, in a social network graph (with nodes representing users, and edges representing 'friendship' relations), the 'friendship' between two nodes, not explicitly connected in the graph, can be inferred from the existence of edges between other nodes [28]. Private attributes in a user's record can be inferred by exploiting the public attributes of other users sharing similar interests [6]. A user's susceptibility to a contagious disease can be easily inferred by an adversary who has access to noisy query results and is aware of the fact that the user's immediate family members are part of the database being queried [24]. Social and behavioral dependence have also been used to perform de-anonymization attacks on released datasets [22], [32], [33], [37].

The fact that dependence (or correlation) among tuples can degrade the expected privacy guarantees of DP mechanisms was first observed by Kifer et al. [24], and later in [8], [21], [25], [40]. Based on our own experiments with real-world datasets in Section IV, we attribute this degradation to a faster exhaustion of the privacy budget in DP. In prior work, the Pufferfish framework [25], proposed as a generalization of DP, incorporated adversarial belief about existing data relationships using a data generation model maintained as a distribution over all possible database instances. However, the framework did not propose any specific perturbation algorithm to handle the dependence. The Blowfish framework [21], which is a subclass of the Pufferfish framework, allowed users to specify adversarial knowledge about the database in the form of deterministic policy constraints and provided perturbation mechanisms to handle these constraints. Finally, to handle correlation in network data using DP, the authors in [8] multiplied the *sensitivity* of the query output with the number of correlated records. This technique resulted in excessive noise being added to the output severely degrading the utility of the shared data, which serves as the baseline approach in our experiments.

NDSS '16, 21-24 February 2016, San Diego, CA, USA
Copyright 2016 Internet Society, ISBN 1-891562-41-X
http://dx.doi.org/10.14722/ndss.2016.23279

In this paper, we formalize the notion of dependent differential privacy (DDP) to handle *probabilistic dependence constraints* between tuples while providing rigorous privacy guarantees. We further develop an effective dependent perturbation mechanism (DPM) to achieve the privacy guarantees in DDP. Our mechanism uses a carefully computed *dependence coefficient* that quantifies the probabilistic dependence between tuples in a fine-grained manner. We interpret this coefficient as the ratio of *dependent indistinguishability* of a tuple which is the maximum change in a query output due to the modification of another dependent tuple and *self indistinguishability* which is the maximum change in a query output due to modification of the tuple itself. In summary, our paper makes the following contributions:

- **Inference Attack:** Using real-world datasets we demonstrate the feasibility of an inference attack on differentially private query results by utilizing the dependence between tuples. We show that an adversary can infer sensitive location information about a user from private query outputs by exploiting her social relationships. Furthermore, this adversary, even with partial knowledge of both the user's social network and the tuple database, can extract more sensitive location information than the adversary in DP (that knows all the tuples in the database except one but is unaware of their dependence relationships), thus violating the DP guarantees.

- **Dependent Differential Privacy:** We formalize the notion of DDP, to defend against adversaries who have prior information about the probabilistic dependence between tuples in a statistical database. We then show that it is possible to achieve the DDP guarantees by augmenting the Laplace mechanism, used for achieving the DP guarantees, with a dependence coefficient. The coefficient allows accurate computation of the query sensitivity for dependent data, thus minimizing the noise that needs to be added providing better utility at the same privacy level. Furthermore, we prove that our dependent perturbation mechanism is also resilient to composition attacks [11], [18].

- **Evaluation:** Our proposed dependent perturbation mechanism applies to any class of query functions. Using extensive evaluation involving different query functions (e.g., machine learning queries such as clustering and classification, and graph queries such as degree distribution) over multiple large-scale real-world datasets we illustrate that our DPM outperforms state-of-the-art approaches in providing rigorous privacy and utility guarantees for dependent tuples.

II. Preliminaries

In this section, we introduce terms used in formalizing the notion of differential privacy.

A. Differential Privacy

Differential privacy is a rigorous mathematical framework aimed at protecting the privacy of users' sensitive information in a statistical database [12]–[16]. The threat to privacy arises from the release of aggregate query results computed over the statistical database. The goal of DP is to randomize the query results to ensure that the risk to a users' privacy does not increase substantially (bounded by a function of the privacy budget ϵ) as a result of participating in the statistical database. We represent a statistical database using a vector $D = [D_1, D_2, \cdots, D_n]$ drawn from domain $\mathcal{D}$, where $D_i \in \mathcal{R}^m$ denotes the data of the i^{th} user. The notion of ϵ-differential privacy is formally defined as:

Definition 1. (ϵ-differential privacy) [12] *A randomized algorithm $\mathcal{A}$ provides ϵ-differential privacy if for any two databases D, D' that differ in only a single entry, and for any output S,*

$$\max_{D,D'} \frac{P(\mathcal{A}(D) = S)}{P(\mathcal{A}(D') = S)} \leq \exp(\epsilon) \tag{1}$$

where $\mathcal{A}(D)$ (resp. $\mathcal{A}(D')$) is the output of $\mathcal{A}$ on input D (resp. D') and ϵ is the privacy budget. Smaller value of the privacy budget ϵ corresponds to a higher privacy level.

B. Achieving Differential Privacy

The Laplace Perturbation Mechanism (LPM), proposed in [12], achieves ϵ-differential privacy. The key idea is to use noise drawn from a suitable Laplace distribution to perturb the query results before their release. Let $Lap(\sigma)$ denote a zero mean Laplace distribution with scaling factor σ. The corresponding density function is given by $f(x) = \frac{1}{2\sigma} \exp\left(-\frac{|x|}{\sigma}\right)$. For a query output of dimension q, LPM uses a noise vector $Lap^q(\sigma)$ where each dimension of the vector is drawn independently from the distribution $Lap(\sigma)$.

Integral to the design of the LPM is the global sensitivity parameter ΔQ, computed for the issued query function Q, and is defined as follows:

Definition 2. (Global sensitivity) [12] *The global sensitivity of a query function $Q : D \to \mathbb{R}^q$, issued on database D, is the maximum difference between the outputs of the function when one input changes (i.e., D and D' differ in only a single entry). Formally,*

$$\Delta Q = \max_{D,D'} \|Q(D) - Q(D')\|_1 \tag{2}$$

Theorem 1. *ϵ-differential privacy is guaranteed if the scaling factor σ in the Laplace distribution is calibrated according to the global sensitivity ΔQ. For any query function Q over an arbitrary domain $\mathcal{D}$, the mechanism $\mathcal{A}$*

$$\mathcal{A}(D) = Q(D) + Lap(\Delta Q/\epsilon) \tag{3}$$

achieves ϵ-differential privacy (see [12] for detailed proof).

III. Adversarial Model

The popularity of DP as a privacy definition (recall Definition 1) stems from the fact that it makes no assumptions about the background knowledge available to an adversary. In other words, mechanisms such as LPM, that satisfy the DP definition, guarantee that users' sensitive data are protected regardless of adversarial knowledge. However, the privacy guarantees provided by the existing DP mechanisms are valid only under the assumption that the data tuples forming the database are pairwise independent (which is also implicitly assumed by the DP adversary model) [8], [21], [24], [25], [27], [40]. In reality, this assumption is a cause of vulnerability as data from different users can be dependent, where the dependence can

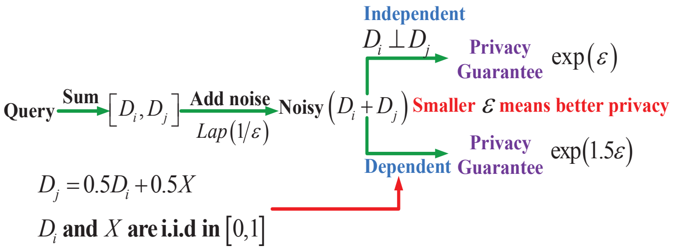


Fig. 1. Dependence between tuples can seriously degrade the privacy guarantees provided by the existing differential privacy mechanisms.

be due to various social, behavioral and genetic interactions that might exist between users. An active adversary can use auxiliary information channels to access these dependence and exploit the vulnerabilities in DP mechanisms as illustrated by the simple example below.

Example 1: Consider a database $D = [D_i, D_j]$ where D_i, D_j have a probabilistic dependence as $D_j = 0.5D_i + 0.5X$ and D_i, X are independently and uniformly distributed within $[0, 1]$ as shown in Fig. 1. Below we consider a simple inference attack in which an adversary issues a sum query $Q(D) = D_i + D_j$ and uses the query result to infer the value of D_i.

First, we consider a DP-adversary, one that assumes independence between D_i and D_j. Following the LPM mechanism, we add Laplace noise with parameter $1/\epsilon$ [1] which allows us to achieve $\epsilon-$differential privacy guarantee, i.e., $\max\limits_S \frac{P(\mathcal{A}([D_i=0,D_j])=S))}{P(\mathcal{A}([D_i=1,D_j])=S))} = \max\limits_S \frac{\exp(-\frac{|S-0-D_2|}{\epsilon})}{\exp(-\frac{|S-1-D_2|}{\epsilon})} \leq \exp(\epsilon)$.

Next, for the same inference attack, we consider a more powerful adversary, one that not only has all the properties of the DP-adversary but in addition also knows the dependence relation between D_i and D_j. Using the same LPM mechanism with Laplace noise of parameter $1/\epsilon$ for such an adversary results in a much weaker privacy guarantee, i.e., $\max\limits_S \frac{P(\mathcal{A}([D_i=0,D_j])=S))}{P(\mathcal{A}([D_i=1,D_j])=S))} = \max\limits_S \frac{\int_{x=0}^{1} \exp(-\frac{|S-0-0-0.5x|}{\epsilon})}{\int_{x=0}^{1} \exp(-\frac{|S-1-0.5-0.5x|}{\epsilon})} \leq \exp(1.5\epsilon)$.

The above example exposes the following vulnerabilities in DP: (1) The privacy Definition 1 does not account for the dependence relations between the tuples in a database; (2) Privacy mechanisms such as LPM rely on the independence of the data tuples for providing privacy guarantees; and motivates the DDP-adversary model which is formally defined below.

DDP-adversary

We assume a setting, in which a trusted data curator maintains a statistical database $D = [D_1, D_2, \cdots, D_n]$ where D_i denotes the data from the i^{th} user. In response to a query, the curator computes a randomized query result $\mathcal{A}(D)$, with the goal of providing statistical information about the dataset while preserving the privacy of individual users. Both the users and the data curator are assumed to be honest. The data recipient, issuing the query, is our DDP-adversary. We associate the following properties to this adversary who wants to use the noisy query result to infer data D_i:

- **Access to $\mathbb{D}_{-i}$**: Data of all the other $n-1$ users (excluding the i^{th} user), denoted by $\mathbb{D}_{-i}$, is available to the adversary. This property makes the DDP-adversary as powerful as a DP-adversary.
- **Access to joint distribution** $P(D_1, \ldots, D_n)$: The adversary uses auxiliary channels (e.g., the Gowalla social network in our attack in Section IV) to estimate the joint probability distribution $P(D_1, \ldots, D_n)$, between the data tuples. This property together with access to $\mathbb{D}_{-i}$ makes a DDP-adversary more powerful than a DP-adversary.

In the remaining paper, unless otherwise specified, the privacy definitions and guarantees are all with respect to the DDP-adversary. In Section IV, we perform a real-world inference attack to demonstrate that a DDP-adversary can extract more private information than guaranteed by DP. In Section V, we develop a new privacy definition *dependent differential privacy* (DDP) that allows for dependence between data tuples in the database. In Section VI, we propose a privacy mechanism to satisfy the DDP definition. We establish formal guarantees for our privacy mechanism and illustrate its efficacy using experiments on large-scale real-world datasets.

IV. Inference Attack: Differential Privacy under Dependent Tuples

Real-world datasets are complex networks that exhibit strong dependence (correlations) and their release introduces various privacy challenges. Adversaries can combine the released obfuscated data (generated by applying the privacy mechanisms on the data), with knowledge of the existing dependence relations to infer users' sensitive information. There exist limited prior work that have outlined realistic inference attacks exposing the vulnerability of DP mechanisms under dependent data tuples [24], [25]. In this section, we demonstrate (1) a real-world inference attack on the LPM-based differential privacy mechanism, as a realistic confirmation of the feasibility of such attacks in practical scenarios; and (2) the capability of a DDP-adversary to use released data, satisfying DP definition, to build an inference attack which violates the security guarantees of DP mechanisms. Before outlining our real inference attack for DP we compare our work with existing related work [17], [18], [24] to highlight the importance of our attack.

- Ganta et al. in [18] explored how one can reason about privacy in the presence of independent anonymized releases of overlapping data. Compared with our inference attack, they do not consider the dependence between data tuples in their attack.
- Fredrikson et al. in [17] considered predicting a patient's genetic marker from the differentially private query results by utilizing demographic information about that patient. Thus, the auxiliary information used in this attack is additional information about a patient (single tuple) and not dependence between tuples.
- Kifer et al. in [24] investigated the inference about the participation of an edge in a social network through observing the number of inter-community edges. The inference performance varied with different network generation models. In contrast to the theoretical work of Kifer et al., we demonstrate inference attacks using real data on complex differentially private machine learning queries.

[1] The global sensitivity for *Example 1* computed according to Definition 2 is 1 since they only consider neighboring database which differ in one entry.

TABLE I. Statistics of the Gowalla location dataset in our selected regions.

	Mahattan,NY	Brooklyn,NY	Queens,NY	San Jose,SF	Oakland,SF	Pasadena,LA	Beverly Hill,LA	Long Beach,LA
# of Users	997	507	811	1228	862	656	1083	825
# of Check-ins	11277	7116	9344	16347	12647	13114	19848	9109
# of Locations	1641	1680	2392	2066	2319	1803	4383	1486

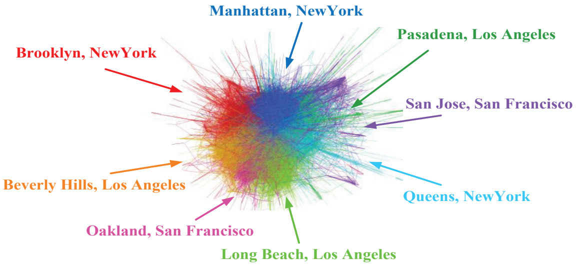


Fig. 2. Gowalla's social dataset colored according to the results from K-means clustering of the location dataset. We can see that the location dataset is inherently correlated and the social dataset well represents such relationships.

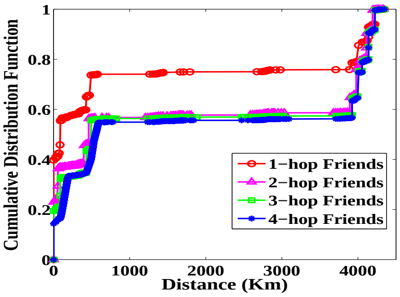


Fig. 3. The distance between the location vectors of users.

A. Dataset Description

Sharing of location information is often associated with serious privacy threats [3], [35], [36]. Location data (or mobility traces), can be easily linked with auxiliary information sources (such as maps) to not only infer places such as home and work location but also a user's political views, her medical conditions, etc.

We use the data collected from the location-based social networking service Gowalla [9] for mounting our attack to infer user's location information. The locations correspond to users' check-ins at places. We obtained a subset of their location dataset which had a total of $196,591$ users and $6,442,890$ check-ins of these users over the period from February 2009 to October 2010. Gowalla also maintains an associated social network connecting its users. In fact, it is this correlation that forms the basis of our inference attack. The network data we obtained consisted of $950,327$ edges connecting $196,591$ users. Considering the sparsity of the location information, we decided to restrict our analysis to users around three cities: New York, San Francisco and Los Angeles. We selected these cities since they had the highest number of active users. For our attack, we used data from users who performed at least 10 check-ins at locations within a 25km radius in any of the three cities. The resulting dataset contains $6,969$ users, $98,802$ check-ins and $17,770$ locations as shown in Table I. The corresponding selected social dataset contains $47,502$ edges connecting these $6,969$ users.

Constructing the Location Pattern Dataset: For each user i, we collect her check-ins in a set $\mathbf{C}(i) = \{\mathbf{c}_{i_0}, \mathbf{c}_{i_1}, \mathbf{c}_{i_2}, \cdots, \mathbf{c}_{i_{m_i}}\}$, where each check-in $\mathbf{c}_{i_k}$=[User ID, timestamp, GPS coordinates, place identifier]. The GPS coordinates $= [lat, lon]$ represents the latitude and longitude of the location shared by the user. For the inference attack, we only consider the GPS coordinates as effective check-in records, and use them to extract a location pattern vector for each user. To do so we compute the frequency of visits to each location, and only keep the latitude and longitude of those locations that correspond to the top-q frequencies. Formally, the location pattern vector for user i is defined as:

$$\mathbf{d}_i = \left(\mathrm{lat}_{i_1}, \mathrm{lon}_{i_1}, \mathrm{lat}_{i_2}, \mathrm{lon}_{i_2}, \mathrm{lat}_{i_3}, \mathrm{lon}_{i_3}, \cdots, \mathrm{lon}_{i_q}\right) \quad (4)$$

where $(\mathrm{lat}_{i_1}, \mathrm{lon}_{i_1})$ is the coordinate of the most frequently visited location of user i, $(\mathrm{lat}_{i_2}, \mathrm{lon}_{i_2})$ and $(\mathrm{lat}_{i_3}, \mathrm{lon}_{i_3})$ correspond to the second and the third most frequently visited location of user i, respectively. Without loss of generality, we normalize each attribute in the location pattern dataset such that its value lies within $[0, 1]$.

B. Differentially Private Data Release

The GPS coordinates describing a user's check-ins are generally considered to be distinct, but in reality they are typically clustered around a limited number of points. We consider a scenario where the data provider uses the classical K-means clustering approach [20] on the Gowalla location dataset to compute cluster centroids, applies DP mechanism on the centroids, and publishes the perturbed centroids to other applications or researchers. The input to the K-means algorithm are points $\mathbf{d}_1 \ldots \mathbf{d}_n$ in the $2q$-dimensional unit cube $[0, 1]^{2q}$. For our attack, we choose $q = 6$ while constructing the location pattern in Eq. 4. To preserve the privacy of users' sensitive data, the data provider perturbs the true centroids $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \widetilde{\boldsymbol{\mu}}_k)$ by using the LPM mechanism, and releases the perturbed centroids $\widetilde{\boldsymbol{\mu}} = (\widetilde{\boldsymbol{\mu}}_1, \widetilde{\boldsymbol{\mu}}_2, \cdots, \widetilde{\boldsymbol{\mu}}_k)$ [2] for preserving the privacy of each individual's location pattern.

Fig. 2 depicts the structure of the Gowalla social network dataset which is colored according to the K-means clustering results of the Gowalla location dataset (users belonging to the same community are giving the same color). We can see that users' location patterns are inherently correlated, and the social dataset embeds relationships contained in the location dataset. Fig. 3 further shows the cumulative distribution function of the location pattern distance between different user pairs, where we compute the distance for $\mathbf{d}_i$ and $\mathbf{d}_j$ as

[2] In this paper, we use $\tilde{a}$ to represent the perturbed version of a, and $\hat{a}$ to represent the estimated value of a.

$$\mathrm{Dist}(\mathbf{d}_i, \mathbf{d}_j) = \frac{1}{q}\sum_{l=1}^{q} dist\left((\mathrm{lat}_{il}, \mathrm{lon}_{il}), (\mathrm{lat}_{jl}, \mathrm{lon}_{jl})\right) \quad (5)$$

and $dist\left((\mathrm{lat}_{il}, \mathrm{lon}_{il}), (\mathrm{lat}_{il}, \mathrm{lon}_{il})\right)$ represents the earth's surface distance between two coordinates[3].

We find that the distance between the location patterns for closer friends is smaller. These observations from Fig. 2 and Fig. 3 not only imply that the location patterns of users are correlated with each other, but also that their social network can serve as an important external information source for an adversary to infer a user's sensitive records.

C. Inference Algorithm

The adversary can observe the differentially private community centroids $\widetilde{\boldsymbol{\mu}} = [\widetilde{\boldsymbol{\mu}}_1, \widetilde{\boldsymbol{\mu}}_2, \cdots, \widetilde{\boldsymbol{\mu}}_k]$, has access to auxiliary information $\mathbb{D}_{-i}$ (recall DDP-adversary in Section III) and also to the social relationships among the users. Let the adversary's estimated value of D_i be $\hat{D}_i$. Using Bayes' theorem, the posterior probability of $\hat{D}_i = \hat{\boldsymbol{d}}_i$ computed by the DDP-adversary can thus be written as

$$\begin{aligned}
&P(\hat{D}_i = \hat{\boldsymbol{d}}_i | \widetilde{\boldsymbol{\mu}}, \mathbb{D}_{-i}) \\
=&\frac{P(\widetilde{\boldsymbol{\mu}}, \mathbb{D}_{-i} | \hat{D}_i = \hat{\boldsymbol{d}}_i) P(\hat{D}_i = \hat{\boldsymbol{d}}_i)}{P(\widetilde{\boldsymbol{\mu}}, \mathbb{D}_{-i})} \\
=&\frac{P(\widetilde{\boldsymbol{\mu}} | \mathbb{D}_{-i}, \hat{D}_i = \hat{\boldsymbol{d}}_i) P(\mathbb{D}_{-i} | \hat{D}_i = \hat{\boldsymbol{d}}_i) P(\hat{D}_i = \hat{\boldsymbol{d}}_i)}{P(\widetilde{\boldsymbol{\mu}}, \mathbb{D}_{-i})} \\
=&\frac{P(\widetilde{\boldsymbol{\mu}} | \mathbb{D}_{-i}, \hat{D}_i = \hat{\boldsymbol{d}}_i) P(\hat{D}_i = \hat{\boldsymbol{d}}_i | \mathbb{D}_{-i})}{P(\widetilde{\boldsymbol{\mu}} | \mathbb{D}_{-i})} \\
\sim& \exp\{-|\widetilde{\boldsymbol{\mu}} - \hat{\boldsymbol{\mu}}|\epsilon\} \cdot P(\hat{D}_i = \hat{\boldsymbol{d}}_i | \mathbb{D}_{-i}) \quad (6)
\end{aligned}$$

where $\exp\{-|\widetilde{\boldsymbol{\mu}} - \hat{\boldsymbol{\mu}}|\epsilon\}$ in Eq. 6 represents the Laplace noise induced by the estimated centroids $\hat{\boldsymbol{\mu}}$. Such estimated centroids are computed using the auxiliary $\mathbb{D}_{-i}$ of other users and each potential value $\hat{D}_i = \hat{\mathbf{d}}_i$. $P(\hat{D}_i = \hat{\boldsymbol{d}}_i | \mathbb{D}_{-i})$ represents the prior information of D_i inferred from the auxiliary information $\mathbb{D}_{-i}$ of other users. Note that our inference attack can be mounted with any amount of auxiliary information. For an adversary with partial auxiliary information, the corresponding estimated centroids $\hat{\boldsymbol{\mu}}$ and prior information $P(\hat{D}_i = \hat{\boldsymbol{d}}_i | \mathbb{D}_{-i})$ are computed based on these partial auxiliary information.

To estimate $\hat{D}_i$, an adversary can discretize the potential region of D_i and compute $\exp\{-|\widetilde{\boldsymbol{\mu}} - \hat{\boldsymbol{\mu}}|\epsilon\} \cdot P(\hat{D}_i = \hat{\boldsymbol{d}}_i | \mathbb{D}_{-i})$ for each potential value $\hat{\boldsymbol{d}}_i$. The adversary can then estimate $\hat{D}_i$ which corresponds to the maximal posterior probability for all the potential values $\hat{\boldsymbol{d}}_i$, i.e.,

$$\hat{D}_i = \underset{\hat{\boldsymbol{d}}_i}{argmax} \exp\{-|\widetilde{\boldsymbol{\mu}} - \hat{\boldsymbol{\mu}}|\epsilon\} \cdot P(\hat{D}_i = \hat{\boldsymbol{d}}_i | \mathbb{D}_{-i}) \quad (7)$$

The key challenge is for the adversary to compute the prior information $P(\hat{D}_i = \hat{\boldsymbol{d}}_i | \mathbb{D}_{-i})$. We consider two different types of adversaries: one which assumes that the tuples are independent and the other which utilizes the social relationships between the users.

1) Attack 1 (Independent Tuple Assumption): First, we consider an adversary who assumes the tuples within the dataset are independent as in the standard differential privacy model. To simplify our analysis without loss of generality, we assume that D_i is independent of $\{D_j\}_{j=0, j\neq i}^{n-1}$ (i.e., $P(\hat{D}_i = \hat{\boldsymbol{d}}_i | \mathbb{D}_{-i}) = P(\hat{D}_i = \hat{\boldsymbol{d}}_i)$) with identical distributions.

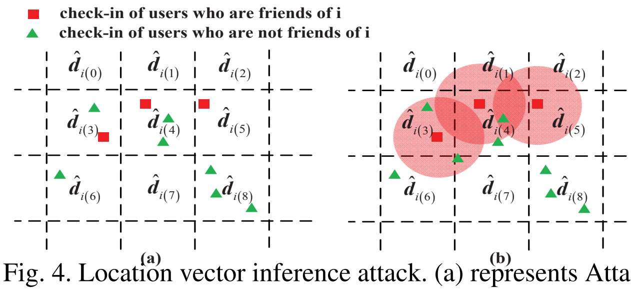


Fig. 4. Location vector inference attack. (a) represents Attack 1 under independent tuple assumption and (b) represents Attack 2 under dependent tuple assumption.

Therefore, the auxiliary information $\mathbb{D}_{-i}$ can serve as sampling values of D_i, which can be utilized to estimate the prior probability $P(\hat{D}_i = \hat{\boldsymbol{d}}_i)$.

Fig. 4 (a) shows the mechanism for inference attack under the independent assumption. We discretize the estimated region for $\hat{D}_i$ where each grid corresponds to a potential value $\hat{\mathbf{d}}_i$ of D_i. The red squares (friends of user i) and the green triangles (non-friends of user i) are location patterns of the other users which also represent the sampling values of D_i. Based on these sampling values, we estimate the prior probability of $\hat{D}_i = \hat{\boldsymbol{d}}_i$ by counting the number of values in $\mathbb{D}_{-i}$ that fall into the grid of $\hat{\boldsymbol{d}}_i$ as

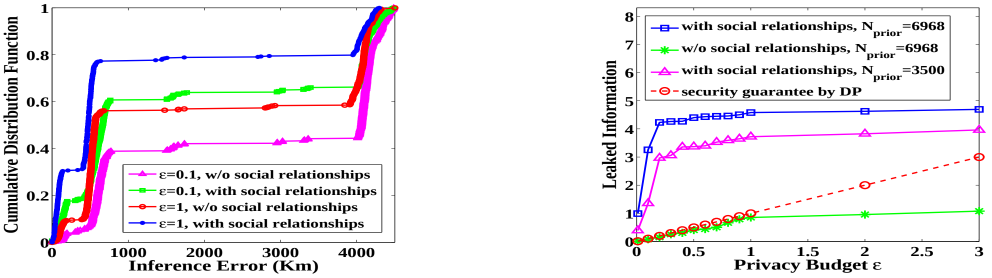
$$P_{inde}(\hat{D}_i = \hat{\boldsymbol{d}}_i | \mathbb{D}_{-i}) = \frac{|\mathbf{d}_\mathrm{j} : \mathbf{d}_\mathrm{j} \in \mathrm{grid}(\hat{\mathbf{d}}_\mathrm{i})|}{\sum\limits_{\hat{\mathbf{d}}_k} |\mathbf{d}_\mathrm{j} : \mathbf{d}_\mathrm{j} \in \mathrm{grid}(\hat{\mathbf{d}}_\mathrm{k})|} \quad (8)$$

2) Attack 2 (Dependent Tuple Assumption): Next, we consider a sophisticated adversary who assumes that tuples in the dataset are dependent on each other. Such an assumption is practical since the mobility traces from close friends are likely to be similar as shown in Fig.3. For an adversary who has access to the social relationships of the users, he can draw circles, shown in red, in Fig. 4(b), to represent the dependent relationships among users, and all the girds (corresponding to each potential value $\hat{\mathbf{d}}_i$) within the red circles would be given a higher weight. The prior probability for $\hat{D}_i = \hat{\mathbf{d}}_i$ would thus be weighted based on the relationships of the users, and the weighted prior probability under the dependent assumption would become

$$P_{de}(\hat{D}_i = \hat{\mathbf{d}}_i | \mathbb{D}_{-i}) = \frac{\mathrm{weight}(\hat{\mathbf{d}}_i) |\mathbf{d}_j : \mathbf{d}_j \in \mathrm{grid}(\hat{\mathbf{d}}_i)|}{\sum\limits_{\hat{\mathbf{d}}_k} \mathrm{weight}(\hat{\mathbf{d}}_k) |\mathbf{d}_j : \mathbf{d}_j \in \mathrm{grid}(\hat{\mathbf{d}}_k)|} \quad (9)$$

In Fig. 4(a), we can see that there are three sampling values that belong to the grids corresponding to $\hat{D}_i = \hat{\mathbf{d}}_{i(4)}$ and $\hat{D}_i = \hat{\mathbf{d}}_{i(8)}$. Therefore, we have $P_{inde}(\hat{D}_i = \hat{\mathbf{d}}_4 | \mathbb{D}_{-i}) = P_{inde}(\hat{D}_i = \hat{\mathbf{d}}_8 | \mathbb{D}_{-i})$. However, in Fig. 4(b), the grid for $\hat{D}_i = \hat{\mathbf{d}}_{i(4)}$ would have a much higher weight than the grid for $\hat{D}_i = \hat{\mathbf{d}}_{i(8)}$. Therefore, we have $P_{de}(\hat{D}_i = \hat{\mathbf{d}}_{i(4)} | \mathbb{D}_{-i}) > P_{de}(\hat{D}_i = \hat{\mathbf{d}}_{i(8)} | \mathbb{D}_{-i})$. As we know the location patterns of the user i's friends (shown as the red squares Fig. 4), it is more likely that the location pattern D_i of user i will be located closer to her friends based on our observations in Fig. 3.

[3] http://www.movable-type.co.uk/scripts/latlong.html



(a)

(b)

Fig. 5. Performance ((a) the inference error and (b) the leaked information) for the inference attack.

D. Experimental Evaluation

We evaluate the performance for these two inference attacks by measuring the following two metrics

$$Inference\ Error = \frac{1}{n}\sum\nolimits_{i=1}^{n} \text{Dist}(\mathbf{d}_i, \hat{D}_i) \qquad (10)$$

$$Leaked\ Information = \frac{1}{n}\sum\nolimits_{i=1}^{n} H(D_i) - H(D_i|\widetilde{\boldsymbol{\mu}}, \mathbb{D}_{-i}) \qquad (11)$$

$\text{Dist}(\cdot)$ is defined in Eq. 5 and $H(\cdot)$ denotes the entropy (information) of a random variable [10]. $H(D_i)$ evaluates the adversary's prior information for D_i without utilizing the social relationships and is the entropy of the prior probability in Eq. 8, $H(D_i|\widetilde{\boldsymbol{\mu}}, \mathbb{D}_{-i})$ evaluates the adversary's posterior information after the inference attack and is the entropy of the posterior probability in Eq. 6 (combined with Eq. 8 under the independent assumption or combined with Eq. 9 under the dependent assumption). By evaluating the *Leaked Information*, we can measure the privacy breaches in terms of change in an adversary's a-priori to a-posteriori beliefs.

We set the number of communities $K = 8$ in the K-means algorithm (shown in Fig. 2) and discretize each city (NY, SF, LA) into 20×20 grids. The prior information for the adversary before the inference attack can be computed as $H(D_i) = 8.38$ bits according to Eq. 8. From the results in Fig. 5, we can see that the attacker can exploit the social relationships between users to make better inferences (shown by smaller inference errors in Fig. 5(a) and more information leakage in Fig. 5(b)). Furthermore, in Fig. 5(b), larger ϵ (worse privacy guarantee) results in smaller inference errors and more information leakage, since the adversary has access to more accurate centroids.

DDP-adversary with Partial Information: We also consider a more realistic adversary that has access to partial information of other users, e.g., $N_{prior} = 3500$ (roughly half of all the other $6,968$ users) as in Fig. 5(b). By utilizing social relationships, an adversary who only has access to partial information of other users' location data can still infer more information than the DP adversary (recall Section III) who has access to location information of all the other users but ignores their dependence. Therefore, our advanced inference attack is still effective even for realistic adversaries with partial auxiliary information.

Violating DP Guarantees: We first prove that the maximum information leakage due to a DP-adversary, quantified by the metric in Eq. 11, is bounded by ϵ. This is the upper bound on the information leakage from a differentially private query output due to a DP-adversary.

Theorem 2. *The $Leaked\ Information$ (in Eq. 11) for an ϵ-differentially private mechanism is bounded by ϵ.*

Detailed proof for Theorem 2 is deferred to the appendix. As illustrated in Fig. 5(b), the information leakage due to a DDP-adversary, that exploits dependence relationships between tuples, exceeds the upper bound computed on a DP-adversary. This proves our claim that a DDP-adversary can violate the security guarantees provided by DP mechanisms.

From our analysis, we can see that a differential privacy technique performed on a dependent data set will disclose more information than expected, and this is a serious privacy violation which hinders its applications to real-world data that may be inherently dependent. Note that we used the location data just as an example, and our attack observations are broadly applicable to any dataset that exhibits probabilistic dependence between user records. Therefore, we have to take the dependent relationships into consideration when applying differential privacy to real-world dependent tuples.

V. Dependent Differential Privacy

As demonstrated in Section IV, DP underestimates the privacy risk in the presence of dependent tuples, resulting in degradation of expected privacy for existing DP mechanisms. Hence, for databases with dependent tuples, a stronger privacy notion is required.

Recent work has made attempts to capture and model this notion of tuple dependence and correlation in databases. The Pufferfish framework [25], proposed as a generalization of DP, incorporates adversarial belief about a database and its generation as a distribution over all possible database instances. The Blowfish framework [21], which is a subclass of the Pufferfish framework, allows a user to specify adversarial knowledge about the database in the form of deterministic policy constraints.

Motivated by the above frameworks, we formalize the notion of dependent differential privacy, as a subclass of the general Pufferfish framework, incorporating probabilistic dependence between the tuples in a statistical database. In addition, we also propose an effective perturbation mechanism (Section VI) that can provide rigorous privacy guarantees. In contrast, there are no general algorithms known for achieving Pufferfish privacy.

For any database $D = [D_1, D_2, \cdots, D_n]$, we define its *dependence size* to be L if any tuple in D is dependent on at most $L-1$ other tuples. We denote by $\mathcal{R}$ the *probabilistic dependence relationship* among the L dependent tuples. Relationship $\mathcal{R}$ could be due to the data generating process as specified in [24] or could be due to other social, behavioral and genetic relationships arising in real-world scenarios. We provide an instance of $\mathcal{R}$ in Section IV, where dependence in the Gowalla location dataset was introduced via the Gowalla social network dataset and such dependence is probabilistic instead of deterministic as in Blowfish framework [21]. The DDP framework is equivalent to the DP framework when $\mathcal{R}$ represents independence between data tuples. We begin by defining the *dependent neighboring databases* as follows:

Definition 3. *Two databases $D(L,\mathcal{R}), D'(L,\mathcal{R})$ are* dependent neighboring databases, *if the modification of a tuple value in database $D(L,\mathcal{R})$ (e.g., the change from D_i in $D(L,\mathcal{R})$ to D_i') causes change in atmost $L-1$ other tuple values in $D'(L,\mathcal{R})$ due to the probabilistic dependence relationship $\mathcal{R}$ between the data tuples.*

Based on the above *dependent neighboring databases*, we define our dependent differential privacy as follows.

Definition 4. (ϵ-Dependent Differential Privacy) *A randomized algorithm $\mathcal{A}$ provides ϵ-dependent differential privacy, if for any pair of* dependent neighboring databases *$D(L,\mathcal{R})$ and $D'(L,\mathcal{R})$ and any possible output S, we have*

$$\max_{D(L,\mathcal{R}),D'(L,\mathcal{R})} \frac{P(\mathcal{A}(D(L,\mathcal{R}))=S)}{P(\mathcal{A}(D'(L,\mathcal{R}))=S)} \leq \exp(\epsilon) \qquad (12)$$

where L denotes the dependence size and $\mathcal{R}$ is the probabilistic dependence relationship between the data tuples.

From Definition 4, we see that dependent differential privacy restricts an adversary's ability to infer the sensitive information of an individual tuple, even if the adversary has complete knowledge of the probabilistic dependence relationship $\mathcal{R}$ between the tuples.

A. Security Analysis

Dinur et al. [11] proved that unless a particular amount of noise is added to the query responses, an adversary can use a polynomial number of queries to completely reconstruct the database. Therefore, any privacy framework must provide privacy guarantees for multiple queries, in order to defend against such *composition attacks* [11], [18]. In the following, we show that DDP is secure against these composition attacks. Here, 'secure' means that the algorithms that provide strict DDP also provide meaningful privacy in the presence of auxiliary information. To this end, we propose both the *sequential composition theorem* and the *parallel composition theorem* for DDP by extending the previous results on composition for DP in [30]. Our analysis show that the composition properties for DDP provide privacy guarantees in a well-controlled manner, rather than collapsing rapidly as other approaches in [18]. The proofs for Theorems 3 and 4 follow directly from the ones presented in [30] for differential privacy and are deferred to the appendix to improve readability.

Sequential Composition Theorem Multiple queries that each provides dependent differential privacy in isolation provide dependent differential privacy in sequence.

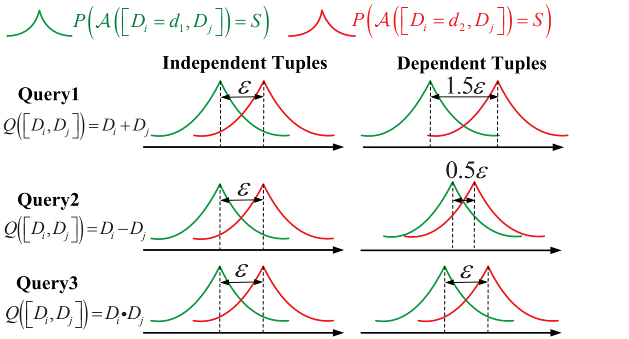


Fig. 6. *Separation* for different queries under dependent tuples.

Theorem 3. *Let randomized algorithm $\mathcal{A}_t$ each provide ϵ_t-dependent differential privacy under the dependence size L and probabilistic dependence relationship $\mathcal{R}$ over the same input data D. The sequence of these algorithms $\mathcal{A}_t$ provides $\sum_t \epsilon_t$-dependent differential privacy under the same $L, \mathcal{R}$.*

Parallel Composition Theorem When the queries are applied to disjoint subsets of the data, we have the parallel composition theorem as

Theorem 4. *Let randomized algorithms $\mathcal{A}_t$ provide ϵ_t-dependent differential privacy under the dependence size L and probabilistic dependence relationship $\mathcal{R}$. We denote by D_t the arbitrary disjoint subsets of the input domain $\mathcal{D}$. The sequence of these randomized algorithm $\mathcal{A}_t$ provides $\max\limits_t \epsilon_t$-dependent differential privacy under the same $L, \mathcal{R}$.*

B. Privacy Axioms

Kifer et al. in [23] suggested two privacy axioms: *transformation invariance* and *convexity* that should be satisfied by any consistent privacy definition. The following theorems show that our DDP satisfies both the axioms.

Theorem 5. Transformation Invariance Property*: For a randomization algorithm $\mathcal{A}$ that satisfies ϵ-dependent differential privacy under the dependence size L and probabilistic dependence relationship $\mathcal{R}$ and any other randomization algorithm $\mathcal{B}$, $\mathcal{B}_{\mathcal{A}}(\cdot) = \mathcal{B}(\mathcal{A}(\cdot))$ also satisfies ϵ-dependent differential privacy under the same $L, \mathcal{R}$.*

Theorem 6. Convexity Property*: For two randomization algorithms $\mathcal{A}_1, \mathcal{A}_2$ that both satisfy ϵ-dependent differential privacy under the dependence size L and probabilistic dependence relationship $\mathcal{R}$, let $\mathcal{A}^p$ represent an algorithm that runs $\mathcal{A}_1$ with probability p and runs $\mathcal{A}_2$ with probability $1-p$, then $\mathcal{A}^p$ also satisfies ϵ-dependent differential privacy under the same $L, \mathcal{R}$.*

Proofs for the above two theorems are also deferred to appendix to improve readability.

VI. Mechanism Design for DDP

In this section, we design an effective mechanism to achieve ϵ-dependent differential privacy and support private query results over dependent tuples. We also describe extensions to the existing LPM-based differential privacy scheme that allows it to be used in the DDP setting.

To provide more insights into our privacy mechanism design, we take a further look at *Example 1* in Section III. Recall that the probabilistic dependence relationship $\mathcal{R}$ was specified

$$
\begin{aligned}
&\max_{\mathbf{d}_{i_1},\mathbf{d}_{i_2}} \frac{P\left(A([D_i=\mathbf{d}_{i_1},D_j])=[\widetilde{\mathbf{d}}_i,\widetilde{\mathbf{d}}_j]\right)}{P\left(A(D_i=[\mathbf{d}_{i_2},D_j])=[\widetilde{\mathbf{d}}_i,\widetilde{\mathbf{d}}_j]\right)}\\
&=\max_{\mathbf{d}_{i_1},\mathbf{d}_{i_2}} \frac{P(\widetilde{D}_i=\widetilde{\mathbf{d}}_i|D_i=\mathbf{d}_{i_1})\sum_{\mathbf{d}_j}P(D_j=\mathbf{d}_j|D_i=\mathbf{d}_{i_1})P(\widetilde{D}_j=\widetilde{\mathbf{d}}_j|D_j=\mathbf{d}_j)}{P(\widetilde{D}_i=\widetilde{\mathbf{d}}_i|D_i=\mathbf{d}_{i_2})\sum_{\mathbf{d}_j}P(D_j=\mathbf{d}_j|D_i=\mathbf{d}_{i_2})P(\widetilde{D}_j=\widetilde{\mathbf{d}}_j|D_j=\mathbf{d}_j)}\\
&\le\max_{\mathbf{d}_{i_1},\mathbf{d}_{i_2}} \frac{P(\widetilde{D}_i=\widetilde{\mathbf{d}}_i|D_i=\mathbf{d}_{i_1})}{P(\widetilde{D}_i=\widetilde{\mathbf{d}}_i|D_i=\mathbf{d}_{i_2})}\max_{\mathbf{d}_{i_1},\mathbf{d}_{i_2}} \frac{\sum_{\mathbf{d}_j}P(D_j=\mathbf{d}_j|D_i=\mathbf{d}_{i_1})P(\widetilde{D}_j=\widetilde{\mathbf{d}}_j|D_j=\mathbf{d}_j)}{\sum_{\mathbf{d}_j}P(D_j=\mathbf{d}_j|D_i=\mathbf{d}_{i_2})P(\widetilde{D}_j=\widetilde{\mathbf{d}}_j|D_j=\mathbf{d}_j)}
\end{aligned}
\tag{12}
$$

as $D_j = 0.5D_i + 0.5X$ where D_i, X are independently and uniformly distributed within $[0, 1]$.

Quantifying the performance of LPM: We use *separation* as a metric to analyze the performance of the LPM-based differential privacy scheme under dependent tuples. *Separation* measures the maximum difference between two Laplace distributions of $P(\mathcal{A}([D_i = d_1, D_j]) = S)$ and $P(\mathcal{A}([D_i = d_2, D_j]) = S)$. Smaller *separation* implies better privacy performance. We further consider three query functions *sum, subtraction, multiplication* over the same dependent database. To achieve DP, an Laplace noise with parameter $1/\epsilon$ is added to each of the three query results. Assuming independent tuples, the *separation* for each noisy query output is the same ϵ as guaranteed by DP (shown in Fig. 6). In comparison, under the probabilistic dependence between tuples, we have the following interesting observations: 1) the *separation* may become larger for the dependent tuples than that under the independent assumption (see the *sum* query in Fig. 6); and 2) the change of *separation* caused by the same dependent database may vary for different queries. In this section, we aim to develop a principled perturbation mechanism for supporting arbitrary query functions, by introducing an extra parameter *dependence coefficient* to measure the fine-grained dependence relationship between tuples.

A. Baseline Approach

A database $D(L, \mathcal{R})$ with dependence size L would result in a quicker exhaustion of the privacy budget ϵ in DP by a factor of L. This observation provides the baseline approach for achieving the ϵ-dependent differential privacy as stated in the theorem below:

Theorem 7. *An ϵ/L-differentially private mechanism $\mathcal{A}(D) = Q(D) + Lap(L\Delta Q/\epsilon)$ over a database D with the dependence size L achieves ϵ-dependent differential privacy, for a query function Q with global sensitivity ΔQ.*

While the above theorem follows directly from the definition of DDP, the baseline approach is not optimal as it implicitly assumes that all the dependent tuples in the database are *completely dependent* on each other. By *completely dependent*, we mean that the change in one tuple would cause a dependent tuple to change by the maximum domain value, thus making the sensitivity of the query over the two tuples twice the sensitivity under the independent assumption. As we can see from Fig. 6, the sensitivity for the *sum* query ΔQ, under the independent tuple assumption, is 1 as $D_i \in [0, 1]$. Under the dependent tuples, the maximum change in D_j caused by the change of D_i is 0.5, which is only half of the maximum domain value for D_j. Therefore, the sensitivity of the *sum* query over the two dependent tuples is 1.5, which is smaller than $2 \times \Delta Q = 2$ as considered in the baseline approach.

This conservative assumption of *completely dependent* tuples results in the addition of a lot of unnecessary noise to the query output rendering it unusable. In real-world datasets, although the tuples are related, only a few of them are *completely dependent* on each other. This insight motivates us to explore mechanisms that can use less amount of noise but still satisfy all the guarantees provided by ϵ-dependent differential privacy.

B. Our Dependent Perturbation Mechanism

To minimize the amount of added noise we want to identify the fine-grained dependence relationship between tuples and use it to design the mechanism. We begin with a simple query function (e.g., an identity query) over a dataset with only two tuples $D = [D_i, D_j]$. The privacy objective is to publish a *sanitized version of the dataset* i.e., $\widetilde{D} = [\widetilde{D}_i, \widetilde{D}_j]$ as query output. We later generalize our analysis to scenarios involving arbitrary query functions over databases with more than two tuples, i.e., $D = [D_i, D_j, D_k, \cdots]$. According to Definition 4, to satisfy ϵ-dependent differentially privacy requires

$$
\max_{\mathbf{d}_{i_1},\mathbf{d}_{i_2}} \frac{P\left(\mathcal{A}([D_i=\mathbf{d}_{i_1},D_j])=[\widetilde{\mathbf{d}}_i,\widetilde{\mathbf{d}}_j]\right)}{P\left(\mathcal{A}([D_i=\mathbf{d}_{i_2},D_j])=[\widetilde{\mathbf{d}}_i,\widetilde{\mathbf{d}}_j]\right)} \le \exp(\epsilon) \tag{11}
$$

where the output distributions of $\mathcal{A}$, due to the change in D_i from $\mathbf{d}_{i_1}$ to $\mathbf{d}_{i_2}$, would be bounded.

Motivated by the LPM in Section II-B, we continue to use Laplace noise for perturbing the true query output to satisfy ϵ-dependent differential privacy. Our objective thus reduces to finding a proper scaling factor $\sigma(\epsilon)$ for the required Laplace distribution. According to the law of total probability[4], we further transform the left-handside (LHS) of Eq. 11 to Eq. 12. For the first term of the right-handside (RHS) of Eq. 12, we have

$$
\begin{aligned}
\max_{\mathbf{d}_{i_1},\mathbf{d}_{i_2}} \frac{P(\widetilde{D}_i=\widetilde{\mathbf{d}}_i|D_i=\mathbf{d}_{i_1})}{P(\widetilde{D}_i=\widetilde{\mathbf{d}}_i|D_i=\mathbf{d}_{i_2})} &= \max_{\mathbf{d}_{i_1},\mathbf{d}_{i_2}} \frac{\exp\left(\frac{\|\widetilde{\mathbf{d}}_i-\mathbf{d}_{i_1}\|_1}{\sigma(\epsilon)}\right)}{\exp\left(-\frac{\|\widetilde{\mathbf{d}}_i-\mathbf{d}_{i_2}\|_1}{\sigma(\epsilon)}\right)}\\
&\le \max_{\mathbf{d}_{i_1},\mathbf{d}_{i_2}} \exp\left(\frac{\|\mathbf{d}_{i_1}-\mathbf{d}_{i_2}\|_1}{\sigma(\epsilon)}\right)\\
&\le \exp\left(\frac{\Delta D_i}{\sigma(\epsilon)}\right)
\end{aligned}
\tag{13}
$$

[4] We restrict ourselves to discrete variables for simplicity, but all the results will also apply to the continuous case as in [1].

where ΔD_i is the maximal difference due to the change in D_i. If we ignore the second term in the RHS of Eq. 12 and combine the remaining terms with Eq. 11 and Eq. 13, we obtain the scaling factor of the Laplace noise as $\sigma(\epsilon) = \frac{\Delta D_i}{\epsilon}$, which is exactly the same form as in traditional DP [12]. Therefore, the LPM that satisfies DP is only a special case for our mechanism. The second term in the RHS of Eq. 12 incorporates the dependence relationship between D_i, D_j and we will focus our study on this term.

To evaluate the extent of dependence induced in D_j by the modification of D_i, we define the dependence coefficient ρ_{ij} as

$$\begin{aligned} &\exp\left(\frac{\rho_{ij}\Delta D_j}{\sigma(\epsilon)}\right) \\ &= \frac{\sum\limits_{\mathbf{d}_j} P(D_j = \mathbf{d}_j | D_i = \mathbf{d}_{i_1}) P(\widetilde{D}_j = \widetilde{\mathbf{d}}_j | D_j = \mathbf{d}_j)}{\sum\limits_{\mathbf{d}_j} P(D_j = \mathbf{d}_j | D_i = \mathbf{d}_{i_2}) P(\widetilde{D}_j = \widetilde{\mathbf{d}}_j | D_j = \mathbf{d}_j)} \end{aligned} \tag{14}$$

Next, we aim to prove that $0 \leq \rho_{ij} \leq 1$. We first have,

$$\begin{aligned} &\frac{\sum_{\mathbf{d}_j} P(D_j = \mathbf{d}_j | D_i = \mathbf{d}_{i_1}) P(\widetilde{D}_j = \widetilde{\mathbf{d}}_j | D_j = \mathbf{d}_j)}{\sum_{\mathbf{d}_j} P(D_j = \mathbf{d}_j | D_i = \mathbf{d}_{i_2}) P(\widetilde{D}_j = \widetilde{\mathbf{d}}_j | D_j = \mathbf{d}_j)} \\ &= \max_{\mathbf{d}_{i_1}, \mathbf{d}_{i_2}} \frac{\sum_{\mathbf{d}_j} P(D_j = \mathbf{d}_j | D_i = \mathbf{d}_{i_1}) \exp\left(-\frac{\|\widetilde{\mathbf{d}}_j - \mathbf{d}_j\|_1}{\sigma(\epsilon)}\right)}{\sum_{\mathbf{d}_j} P(D_j = \mathbf{d}_j | D_i = \mathbf{d}_{i_2}) \exp\left(-\frac{\|\widetilde{\mathbf{d}}_j - \mathbf{d}_j\|_1}{\sigma(\epsilon)}\right)} \\ &\leq \max_{\mathbf{d}_{i_1}, \mathbf{d}_{i_2}} \frac{\sum_{\mathbf{d}_j} P(D_j = \mathbf{d}_j | D_i = \mathbf{d}_{i_1}) \exp\left(-\frac{\|\widetilde{\mathbf{d}}_j - \mathbf{d}_j\|_1}{\sigma(\epsilon)}\right)}{\exp\left(-\frac{\|\widetilde{\mathbf{d}}_j - \mathbf{d}_j^{\min}\|_1}{\sigma(\epsilon)}\right)} \\ &\leq \max_{\mathbf{d}_j} \exp\left(\frac{\|\mathbf{d}_j - \mathbf{d}_j^{min}\|_1}{\sigma(\epsilon)}\right) \\ &\leq \exp\left(\frac{\Delta D_j}{\sigma(\epsilon)}\right) \end{aligned} \tag{15}$$

where $\mathbf{d}_j^{\min}$ is the value of $\boldsymbol{d}_j$ that minimizes $\exp\left(\frac{\|\widetilde{\mathbf{d}}_j - \mathbf{d}_j\|_1}{\sigma(\epsilon)}\right)$. Comparing Eq. 14 and Eq. 15, we have $\rho_{ij} \leq 1$. Furthermore, it is obvious that

$$\frac{\sum_{\mathbf{d}_j} P(D_j = \mathbf{d}_j | D_i = \mathbf{d}_{i_1}) P(\widetilde{D}_j = \widetilde{\mathbf{d}}_j | D_j = \mathbf{d}_j)}{\sum_{\mathbf{d}_j} P(D_j = \mathbf{d}_j | D_i = \mathbf{d}_{i_2}) P(\widetilde{D}_j = \widetilde{\mathbf{d}}_j | D_j = \mathbf{d}_j)} \geq 1 \tag{16}$$

Comparing Eq. 14 and Eq. 15, we have $\rho_{ij} \geq 0$. Finally, combining Eq. 11–14, we have

$$\begin{aligned} &\max_{\mathbf{d}_{i_1}, \mathbf{d}_{i_2}} \frac{P(\mathcal{A}([D_i = \mathbf{d}_{i_1}, D_j]) = [\widetilde{\mathbf{d}}_i, \widetilde{\mathbf{d}}_j])}{P(\mathcal{A}([D_i = \mathbf{d}_{i_2}, D_j]) = [\widetilde{\mathbf{d}}_i, \widetilde{\mathbf{d}}_j])} \\ &\leq \exp\left(\frac{\Delta D_i}{\sigma(\epsilon)}\right) \exp\left(\frac{\rho_{ij}\Delta D_j}{\sigma(\epsilon)}\right) \\ &= \exp\left(\frac{(\Delta D_i + \rho_{ij}\Delta D_j)}{\sigma(\epsilon)}\right) \end{aligned} \tag{17}$$

Therefore, the sensitivity under the dependence relationship between D_i and D_j can be computed as $\Delta D_i + \rho_{ij}\Delta D_j$.

The dependence coefficient $\rho_{ij} \in [0, 1]$ serves as an effective metric to evaluate the dependence relationship between two tuples in a fine-grained manner. We make the following observations about ρ_{ij}:

- ρ_{ij} evaluates the dependence relationship between D_i and D_j from the privacy perspective.
- $\rho_{ij} = 0$ corresponds to the setting where $P(D_j = \mathbf{d}_j | D_i = \mathbf{d}_i)$ is independent of $\mathbf{d}_i$. Therefore, the mechanism that satisfies DP is just a special case of our analysis that takes arbitrary dependence relationship between tuples into consideration. In addition, using the sensitivity definition $\Delta D_i + \rho_{ij}\Delta D_j$, we observe that more noise needs to be added than under the independent assumption that computes the sensitivity as ΔD_i.
- $\rho_{ij} = 1$ corresponds to the *completely dependent* setting where D_j can be uniquely determined by D_i. The baseline approach in Section VI-A is just a special case of our analysis where all the dependent L tuples are completely dependent on each other. As all practical privacy notions require some assumptions on the allowed distributions, it makes sense to analyze the fine-grained dependence relationship in order to maximize utility under the same privacy requirement. Compared with the baseline approach, less noise would be added for our dependent perturbation mechanism since we consider fine-grained dependence relationship. In real-world scenarios, tuples are related but few of them are completely dependent i.e., $\rho_{ij} < 1$. Therefore, our proposed dependent perturbation mechanism can significantly decrease the added noise compared with the baseline approach.
- ρ_{ij} is asymmetric, i.e., $\rho_{ij} \neq \rho_{ji}$. The reason is that the dependence coefficient evaluates the extent of dependence in D_j induced by D_i, which is causal and directional. For example, a celebrity's participation in a social network is likely to result in the participation of her fans. However, it may not be the case the other way around.

To generalize and derive ρ_{ij} for any output $\widetilde{\mathbf{d}}_j$, we reformulate ρ_{ij} to avoid the appearance of $\widetilde{\mathbf{d}}_j$. After some manipulations (details are deferred to the appendix to improve readability), we have

$$\rho_{ij} = \frac{\max\limits_{\mathbf{d}_i} \log\left\{\sum\limits_{\mathbf{d}_j} P(D_j = \mathbf{d}_j | D_i = \mathbf{d}_i) \exp\left(\frac{\|\mathbf{d}_j - \mathbf{d}_j^*\|_1}{\sigma(\epsilon)}\right)\right\} \sigma(\epsilon)}{\Delta D_j} \tag{18}$$

where $\mathbf{d}_j^*$ is the optimal solution to $\operatorname*{argmax}\limits_{\mathbf{d}_{j1}} \|\mathbf{d}_j - \mathbf{d}_{j1}\|_1$.

Interpreting ρ_{ij}: To further understand the dependence coefficient in Eq. 18, we define the *Self* and *Dependent Indistinguishability* terms [5] as follows:

$$\begin{aligned} Self\ \ Indistinguishability &= \max_{\mathbf{d}_{j_1}, \mathbf{d}_{j_2}} \frac{P(\widetilde{D}_j = \widetilde{\mathbf{d}}_j | D_j = \mathbf{d}_{j_1})}{P(\widetilde{D}_j = \widetilde{\mathbf{d}}_j | D_i = \mathbf{d}_{j_2})} \\ &= \max_{\mathbf{d}_{j1}, \mathbf{d}_{j2}} \log\left\{\exp\left(\frac{\|\mathbf{d}_{j2} - \mathbf{d}_{j1}\|_1}{\sigma(\epsilon)}\right)\right\} \\ &= \frac{\Delta D_j}{\sigma(\epsilon)} \end{aligned} \tag{19}$$

[5] Dwork et al. in [15] defined Eq. 19 as *Indistinguishability*, and here we name it as *Self Indistingushiability* in order to compare with the *Dependent Indistinguishability* of D_j.

Self Indistinguishability represents the maximal difference of D_j caused by the modification of D_j itself. We further define the *Dependent Indistinguishability* of D_j induced by D_i as

$$Dependent\ Indistinguishability$$
$$= \max_{\mathbf{d}_i} \log \left\{ \sum_{\mathbf{d}_j} P\left(D_j = \mathbf{d}_j | D_i = \mathbf{d}_i\right) \exp\left(\frac{\|\mathbf{d}_j - \mathbf{d}_j^*\|_1}{\sigma(\epsilon)} \right) \right\} \tag{20}$$

Dependent Indistinguishability evaluates the maximal *expected* difference in D_j caused by the modification of D_i. Therefore,

$$\rho_{ij} = \frac{Dependent\ Indistinguishability}{Self\ Indistinguishability} \tag{21}$$

or in other words, ρ_{ij} evaluates the ratio of *dependent indistinguishability* of D_j induced by D_i and the *self indistinguishability* of D_j.

To generalize our dependent perturbation mechanism, we consider an arbitrary query function Q and compute the *dependent sensitivity* of Q over D_j induced by the modification of D_i as

$$DS_{ij}^Q = \rho_{ij} \Delta Q_j \tag{22}$$

where ΔQ_j is the sensitivity of Q with respect to the modification of D_j itself, i.e., $\Delta Q_j = \max_{\mathbf{d}_{j1}, \mathbf{d}_{j2}} \|Q(\cdots, \mathbf{d}_{j1}, \cdots) - Q(\cdots, \mathbf{d}_{j2}, \cdots)\|_1$. We defer the detailed proof for Eq. 22 to appendix to improve readability.

Furthermore, we can generalize the dependent sensitivity to multiple users as

$$DS_i^Q = \sum_{j=C_{i1}}^{C_{iL}} \rho_{ij} \Delta Q_j \tag{23}$$

where $C_{i1}, \cdots, C_{iL}$ represent the L tuples that are dependent with i-th tuple and $\rho_{ii} = 1$. DS_i^Q measures the dependent sensitivity of Q over all tuples in D caused by the modification of one individual tuple D_i. We further derive the dependent sensitivity for the whole dataset as

Theorem 8. *The dependent sensitivity for publishing any query Q over a dependent (correlated) dataset is*

$$DS^Q = \max_i DS_i^Q \tag{24}$$

Finally, the dependent perturbation mechanism (DPM) for achieving ϵ-dependent differential privacy is formalized as

Theorem 9. *For any query function Q over an arbitrary domain $\mathcal{D}$ with dependent tuples, the mechanism $\mathcal{A}$*

$$\mathcal{A}(D) = Q(D) + Lap(DS^Q/\epsilon) \tag{25}$$

gives ϵ-dependent differential privacy.

C. Utility and Privacy Guarantees

While the privacy guarantees of ϵ-differential privacy are well understood, the resulting utility due to the privacy mechanisms is often on a best-effort basis. In the following, we analyze the utility provided by our DPM. To do so, we consider a well known utility definition suggested by Blum et al. in [4].

Definition 5. *((α, β)-Accuracy): A randomization algorithm $\mathcal{A}$ satisfies (α, β) accuracy for a query function Q, if $\max_D |\mathcal{A}(D) - Q(D)| < \alpha$ with probability $1 - \beta$.*

Based on the definition of (α, β)-Accuracy, we have the utility guarantee for DPM as

Theorem 10. *A DPM $\mathcal{A}$ that satisfies ϵ-dependent differential privacy would achieve $\max_D |\mathcal{A}(D) - Q(D)| < \alpha$ with probability $1 - \exp(-\frac{\epsilon\alpha}{DS^Q})$.*

We defer the proof of Theorem 10 to the appendix to improve readability. Furthermore, we theoretically demonstrate the utility and privacy superiority of DPM over the baseline approach in Section VI-A.

Lemma 1. *Under the same privacy budget ϵ, DPM achieves better utility performance than the baseline approach.*

Proof: Given $\epsilon_{DPM} = \epsilon_{base}$, we have $\beta_{DPM} = 1 - \exp(-\frac{\epsilon\alpha}{DS^Q}) > 1 - \exp(-\frac{\epsilon\alpha}{L\Delta Q}) = \beta_{base}$ (since $DS^Q = \max_i \sum_j \rho_{ij} \Delta Q_j \leq L\Delta Q$). Therefore, DPM achieves smaller query errors and thus better utility performance. ■

Lemma 2. *Under the same (α, β)-accuracy, DPM achieves better privacy performance than the baseline approach.*

Proof: Given $\beta_{DPM} = \beta_{base}$, we have $\epsilon_{DPM} = -\frac{DS^Q \log(\beta)}{\alpha} < -\frac{L \log(\beta)}{\alpha} = \epsilon_{base}$ (since $DS^Q = \max_i \sum_j \rho_{ij} \Delta Q_j \leq L\Delta Q$). Therefore, our DPM results in better privacy performance. ■

D. Implications of Dependence Coefficient in System Design

We now discuss a practical challenge regarding the computation of the dependence coefficient. The dependence coefficient ρ_{ij} between two tuples D_i, D_j relies on the probabilistic models of the statistical data. Thus, it is difficult to compute ρ_{ij} reliably unless the probabilistic models are known. Here, we provide several effective strategies to compute ρ_{ij}, as guidelines for a data publisher to select a proper privacy model for her own setting.

1) Complete Knowledge of Dependence Relationship: The first type of analysis assumes that the data publisher has access to the complete knowledge of the dependence relationship between tuples in advance. Sen et al. [34] computed the dependence relationship among tuples using an appropriately constructed probabilistic graphical model. Their method relies on a fully known probabilistic database in which the dependent tuples have associated probabilities. Using the dependent probabilities, we can compute the dependence coefficient according to Eq. 18.

2) Knowledge About Data Generation: However, the entire dependence information between tuples is not always available to the data publisher. Under certain scenarios where the data generation process is known, the data publisher can estimate the dependence relationship by carefully analyzing the data generating process. For example, in [24], assuming that the social network generation model is known, extensive experiments and analysis were described to estimate the dependence relationship of the tuples.

Even in the absence of direct dependence information, analysis can still be carried out to estimate an upper bound on the dependence coefficient based on auxiliary information regarding the data (e.g., by using the Gowalla social datasets in Section IV).

Here, we consider to utilize the friend-based model in [2] to compute the probabilistic dependence relationship, where a user's location can be estimated by her friend's location based on the distance between their locations. Specifically, the probability of a user j locating at $\mathbf{d}_j$ when her friend i is locating at $\mathbf{d}_i$ is

$$P(D_j = \mathbf{d}_j | D_i = \mathbf{d}_i) = a(\|\mathbf{d}_j - \mathbf{d}_i\|_1 + b)^{-c} \tag{26}$$

where $a > 0, b > 0, c > 0$. The effectiveness of the probabilistic dependence relationship in Eq. 26 will be verified on multiple real-world datasets in Section VII. We believe that alternate potential strategies for dependence relationship analysis will be an impactful direction for future work. But no matter which method is applied, our goal is to evaluate fine-grained probabilistic dependence relationship among tuples for designing data sharing algorithms that satisfy ϵ-dependent differential privacy.

3) Challenges in Realistic Scenario: Furthermore, we carefully analyze the influence of inaccurate computation in ρ_{ij} on the overall performance of our DPM. We believe that designers are well-placed to compute ρ_{ij}. If ρ_{ij} is overestimated, DPM is conservative and continues to provide rigorous DDP privacy guarantees. In case of underestimation of ρ_{ij}, there are two cases. In the first case, if our estimated ρ_{ij}^e is larger than the expectation of the adversary who has access to certain auxiliary information, our DPM can still continue to provide rigorous DDP guarantees. However, in second case, in which the underestimation of ρ_{ij} is smaller than the adversary's expectation, we may not achieve the DDP guarantees, but would still provide better privacy than the traditional DP mechanism. To demonstrate the performance degradation due to underestimation of ρ_{ij}, we launch the inference attack in Section IV-C2 to DDP clustering query results which are obtained by utilizing underestimated dependence coefficients $\rho_{ij}^e = 0.8\rho_{ij}, 0.9\rho_{ij}$ in our DPM (ρ_{ij} is computed according to Eqs. 26,18,24). Fig. 7 demonstrates that even if $0.8, 0.9$ ratio of underestimation for ρ_{ij} is utilized in DPM, the leaked information is still well-bounded without uncontrolled collapsing. Therefore, our DPM suffers little degradation for the slight underestimation of ρ_{ij} and is likely to be acceptable for most realistic settings, thus making our DPM robust in real-world scenarios.

Furthermore, we consider a natural relaxation of dependent differential privacy to incorporate such imperfect estimation of ρ_{ij}.

Definition 6. ((ϵ, δ)-Dependent Differential Privacy) *A randomized algorithm $\mathcal{A}$ provides (ϵ, δ)-dependent differential privacy, if for any pair of* dependent neighboring databases *$D(L, \mathcal{R})$ and $D'(L, \mathcal{R})$ and any possible output S, we have*

$$P(\mathcal{A}(D(L, \mathcal{R})) = S) \leq \exp(\epsilon) P(\mathcal{A}(D'(L, \mathcal{R})) = S) + \delta \tag{27}$$

where $D(L, \mathcal{R}), D'(L, \mathcal{R})$ are dependent neighboring databases (recall Definition 3), based on the dependence size L and their probabilistic dependence relationship $\mathcal{R}$.

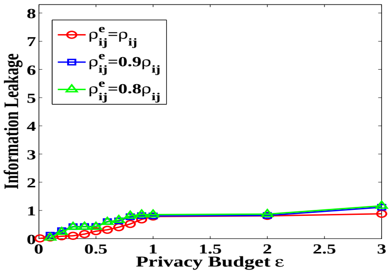


Fig. 7. Information leakage for underestimated ρ_{ij}.

Better accuracy (a smaller magnitude of added noise owing to the underestimation of ρ_{ij}) and generally more flexibility can often be achieved by relaxing the definition of DDP in Eq. 4. Exploring such relaxations of DDP would be an interesting direction for future work.

VII. Experimental Results

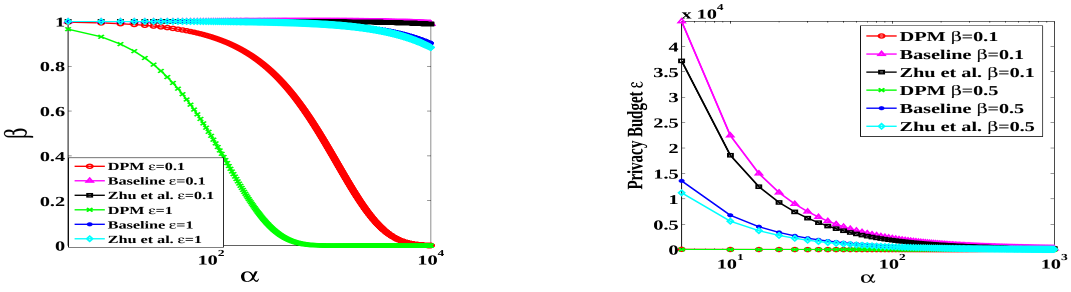
This section evaluates the performance of our proposed dependent data release algorithm on multiple real-world datasets (including Gowalla data in Section IV-A, the *adult* data in UCI Machine Learning Repository, and the large-scale Google+ data [19]). Our objectives are: 1) to show the privacy and utility superiority of our DPM over the state-of-the-art approaches, 2) to study the impact of enforcing DDP on the data in terms of machine learning queries and graph queries, and 3) to analyze the resistance of DPM to inference attacks described in Section IV-C.

A. Privacy and Utility Guarantees

Consider the application scenario in Fig. IV-B, where the data provider publishes the perturbed K-means centroids of the Gowalla location dataset while preserving the privacy of each individual data. Since the Gowalla dataset contains no associated probabilistic distributions or data generating process, we use the general dependent model in Eq. 26 to compute the dependence coefficient ρ_{ij} in Eq. 18 by setting $a = 0.0019, b = 0.196, c = 1.05$ as empirically determined according to [2]. Then, the global sensitivity DS^Q can be computed according to Eq. 23 and Eq. 24.

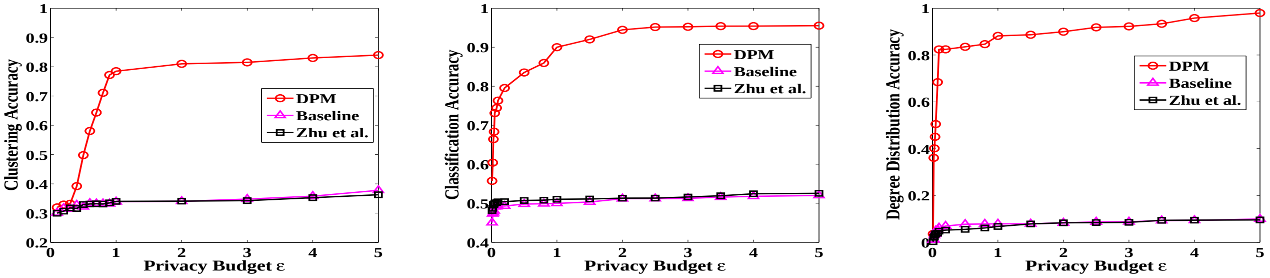
Fig. 8(a) analyzes the (α, β)-accuracy in Definition 5 under various privacy-preserving level ϵ. We can see that under the same α and ϵ, our DPM has much lower β than the baseline approach (where the dependence size L is set to be equal to the number of tuples) and the approach of Zhu et al. in [40][6], i.e., $\|\mathcal{A}(D) - Q(D)\|_1 \leq \alpha$ with higher probability $1 - \beta$. Therefore, DPM achieves much better accuracy than the existing approaches, and such advantage increases with a larger privacy preserving level ϵ. When $\alpha = 1000, \epsilon = 1$, the probability of $\|\mathcal{A}(D) - Q(D)\|_1 < \alpha$ for DPM reaches nearly 1 while in comparison this probability is approximately 0 for the other methods. Similarly, Fig. 8(b) demonstrates that DPM also provides significantly better privacy performance than the existing approaches under the same utility constraint. Therefore, DPM shows significant privacy and utility superiority

[6] The approach of Zhu et al. [40] utilized the linear relationships among tuples for correlated data publishing, which does not satisfy any rigorous privacy metric.



(a) (b)

Fig. 8. Comparison of (a) utility and (b) privacy performance of different perturbation mechanisms.



(a) (b) (c)

Fig. 9. (a) Clustering accuracy, (b) classification accuracy, (c) degree distribution accuracy of different perturbation methods.

over the state-of-the-art approaches as theoretically analyzed in Lemmas 1, 2.

B. Application Quality of Service

1) Clustering: In addition to the (α, β)-accuracy, we further evaluate the utility performance of DPM by sharing the perturbed query results with real-world applications that use machine learning algorithms and analyzing the quality of service for these applications.

We evaluate the clustering accuracy of the dependent differentially private K-means centroids based on the cross-validation mechanism. We randomly select $4/5$ data from the Gowalla location dataset for training the dependent differentially private centroids $\widetilde{\boldsymbol{\mu}} = [\widetilde{\boldsymbol{\mu}}_1, \cdots, \widetilde{\boldsymbol{\mu}}_k]$ and then apply the perturbed centroids to cluster the remaining $1/5$ location data. We repeat the cross-validation process for 1000 times and compare the average clustering performance of DPM with the state-of-the-art approaches. For a more comprehensive investigation, we evaluate the clustering accuracy of DPM under various privacy budget ϵ.

From Fig. 9 (a), we observe that DPM has significantly better clustering accuracy over the baseline approach and that proposed by Zhu et al. in [40]. The reason is that our DPM adds less noise to the K-means centroids by incorporating finer-grained dependence relationship among tuples. Therefore, for dependent datasets, DPM outperforms the state-of-the-art approaches in preserving the quality of service for real-world applications.

For $\epsilon = 0.9$, which corresponds to a fairly strong privacy guarantee, DPM achieves an acceptable clustering performance with nearly 80% accuracy, which is more than twice that of the other approaches. This indicates that DPM is capable of retaining the application quality of service while satisfying a suitable privacy preserving requirement.

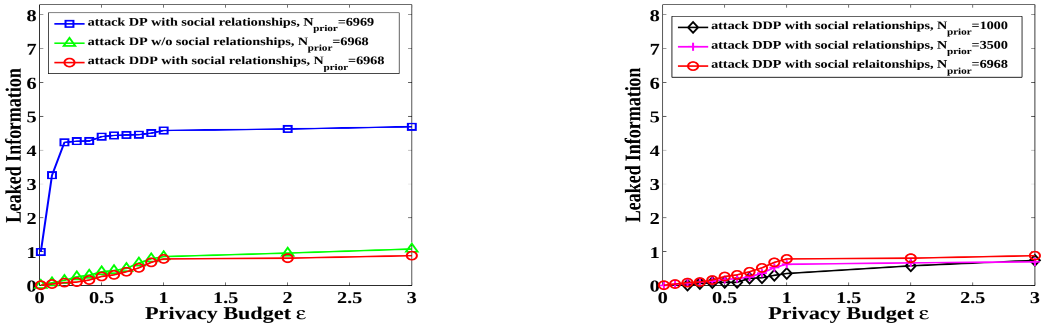
2) Classification: We also apply our DPM to the widely used classification query in machine learning, by designing the dependent differentially private support vector machine (SVM) [7] to the *Adult* dataset in UCI Machine Learning Repository [8]. This dataset contains multiple users' profiles and are labeled according to the users' salaries. By deleting those records with missing attributes, we extract a new dataset with $30,269$ tuples and each tuple has 14 attributes.

To compute the dependence coefficient, we first construct an affinity graph based on the similarities between the users' profiles, where an edge exists for a pair of users i and j if $\frac{\|\mathbf{d}_i^T\mathbf{d}_j\|_1}{\|\mathbf{d}_i\|_1\|\mathbf{d}_j\|_1} > 0.8$ ($\mathbf{d}_i, \mathbf{d}_j$ are the profiles of tuple i, j respectively). Similarly, we compute the dependence coefficient ρ_{ij} for users i and j according to Eqs. 26, 18, and 24. Fig. 9 (b) shows that our DPM has much better classification accuracy than the other methods by considering fine-grained dependence relationship. For $\epsilon = 0.9$, which represents a strong privacy level, DPM achieves an accurate classification performance with 85% accuracy, which is more than twice that of the other approaches. Therefore, DPM could provide an acceptable application quality of service while providing rigorous privacy guarantees.

3) Degree Distribution: We further consider a graph query whose result is to publish the degree distribution of a large-scale Google+ dataset [19]. The Google+ dataset is crawled from July 2011 to October 2011, which consists of

[7] Detailed process for applying DP to SVM classification can be found in [7].

[8] https://archive.ics.uci.edu/ml/datasets/Adult/



(a)

(b)

Fig. 10. (a) Comparison of information leakage due to LPM (for achieving DP) and DPM (for achieving DDP) under the same inference attack. (b) Information leakage due to DPM under various levels of prior information available to an adversary.

28,942,911 users and 947,776,172 edges and thus contains a broad degree distribution. The degree distribution of a graph is a histogram partitioning the nodes in the graph by their degrees [39], and it is often used to describe the underlying structure of social networks for the purposes of developing graph models and making similarity comparisons between graphs[9]. In addition to the social graph, an auxiliary data is also provided in this dataset with users' attributes such as *Employment* and *Education*. To compute the dependence coefficient, we construct an affinity graph based on the similarities between the users' profiles, where an edge is added for a pair of users i and j if $\frac{\|\mathbf{d}_i^T\mathbf{d}_j\|_1}{\|\mathbf{d}_i\|_1\|\mathbf{d}_j\|_1} > 0.8$ and $\mathbf{d}_i, \mathbf{d}_j$ represent the profiles of tuple i, j respectively in the auxiliary data. Similarly, we compute the dependence coefficient ρ_{ij} for users i and j according to Eqs. 26,18,24). Denoting $\mathbf{C}(D)$ and $\mathbf{C}'(D)$ as the true degree distribution and the perturbed degree distribution respectively, we define the accuracy for publishing $\mathbf{C}'(D)$ as $1 - \frac{\|\mathbf{C}(D)-\mathbf{C}'(D)\|_1}{\|\mathbf{C}(D)+\mathbf{C}'(D)\|_1}$. By considering fine-grained dependence relationship, our DPM has significantly higher accuracy for publishing dependent differentially private degree distribution of the social graph than the other methods, with almost 10x improvement as shown in Fig. 9 (c).

C. Resistance to the Inference Attack

To further demonstrate the privacy advantages of DPM, we analyze the resistance of DPM to real-world inference attacks as discussed in Section IV-C2. Fig. 10 (a) shows that the information leakage for DPM is much smaller than that for the traditional DP under the advanced inference attack (corresponding to the dependent scenario in Section IV-C2), and is under similar level as the scenario when the adversary has no access to the social relationships (corresponding to the independent scenario in Section IV-C1). That is to say, the leaked information caused by the dependent tuples has been largely offset by our incorporating dependence relationship to DPM. These results show that DPM can rigorously achieve the expected privacy guarantees for dependent data where traditional DP mechanisms fail, and also validate the effectiveness of our general dependent model in Section VI-D2.

We further investigate the influence of different prior information available to the adversary on the inference attack performance. Fig. 10 (b) shows that the increase in prior information would be beneficial for the adversary to infer more information of the users' location information. Comparing Fig. 10 (a) and Fig. 10 (b), we can also see that DPM shows strong resistance to the adversarial inference attack even under the case where an adversary has access to a large amount of auxiliary information. Therefore, DDP offers a rigorous and provable privacy guarantee for dependent tuples, which demonstrates the necessity of generalizing the standard DP to our DDP.

D. Summary for the Experimental Analysis

- DPM provides significant privacy and utility gains compared to the state-of-the-art approaches. Therefore, we can select a suitable privacy budget ϵ to achieve an optimal privacy and utility balance for DPM.
- DPM is more than 2x accurate in computing the K-means clustering centroids and the SVM classifier, and more than 10x accurate in publishing degree distribution of large-scale social network, compared with existing approaches (which may not even provide rigorous privacy guarantees). These results demonstrate the effectiveness of DPM in real-world query answering for network data.
- DPM is resilient to adversarial inference attack and provides rigorous privacy guarantees for dependent tuples that are not possible using LPM-based DP schemes.

VIII. Related Work

Data privacy is an issue of critical importance, motivating perturbation of query results over sensitive datasets for protecting users' privacy [5], [12], [26], [29], [31], [38]. However, the existing privacy-preserving mechanisms are fraught with pitfalls. A significant challenge is the auxiliary information, which the adversary gleans from other channels. Chaabane et al. [6] inferred users' private attributes by exploiting the public attributes of other users sharing similar interests. Narayaran et al. [32] re-identified users in the anonymous Twitter graph by utilizing information from their Flickr accounts. Srivatsa et al. [37] identified a set of location traces by another social network graph. Other interesting work can be found in [22], [33]. Our inference attack in Section IV demonstrates that the auxiliary information would also be useful to infer an individual's information from differentially private query results.

Differential privacy is one of the most popular privacy

[9] Detailed process for applying differential privacy on degree distribution can be found in [39].

frameworks [12]–[16]. Query answering algorithms that satisfy differential privacy produce noisy query answers such that the distribution of the query answers changes only slightly with the addition, deletion or modification of any tuple. Kifer and Machanavajjhala [24] were the first to criticize that the inherent assumption (limitation) for differential privacy is that the tuples within the dataset are independent of each other. They further argue that the dependence (correlation) among tuples would significantly degrade the privacy guarantees provided by differential privacy.

Tuples in real-world data often exhibit inherent dependence or correlations. Handling dependent tuples is a significant problem. Kifer et al. proposed the Pufferfish framework [25] to provide rigorous privacy guarantees against adversaries who may have access to any auxiliary background information and side information of the database. Blowfish [21] is a subclass of Pufferfish which only considered the data correlations introduced by the deterministic constraints. Our proposed dependent differential privacy is highly motivated by these privacy frameworks and is a subclass of the Pufferfish framework that takes the probabilistic dependence relationships into consideration. We further propose our dependent perturbation mechanism to rigorously achieve dependent differential privacy for general query functions.

Membership Privacy [27] is also applicable for dependent data, however limited anonymization algorithms have been proposed for this framework. Chen et al. [8] dealt with the correlated data by multiplying the original sensitivity with the number of correlated records, which is similar to our baseline approach in Section VI-A. We have shown, both theoretically and experimentally, that the baseline approach would introduce a large amount of noise and thus deteriorate the utility performance of query answers. Zhu et al. [40] exploited the linear relationships among tuples which does not satisfy any rigorous privacy metric. Furthermore, their method has been verified (in Fig. 8) to have significantly worse privacy and utility performance compared to our DPM.

IX. Discussions and Limitations

- Our dependent differential privacy can also accommodate other dependent or correlated relationships such as temporal correlations across a time series of dataset, which opens up an interesting future research direction.
- To form a deeper understanding of our dependent differential privacy, we will also explore the application of standard concepts in differential privacy to our framework, such as local sensitivity, smooth sensitivity [16].
- One limitation of our work is that the dependence coefficient ρ_{ij} is exactly known to both the adversary and the DPM designer. The effectiveness of DPM depends on how well the dependence among data can be modeled and computed. How to accurately compute the dependence coefficient and deal with the underestimation of ρ_{ij} (as we discussed in Section VI-D3) would be an interesting future work (note that the overestimation of ρ_{ij} continues to provide rigorous DDP guarantees).

X. Conclusion

Differential privacy provides a formal basis for expressing and quantifying privacy goals. For these reasons there is an emerging consensus in the privacy community around its use and various extensions are being proposed. However, there remain several limiting assumptions in the original framework that can severely weaken the privacy guarantees expected of a differentially private mechanism. In this paper, we used an inference attack to demonstrate the vulnerability of existing differential privacy mechanisms under data dependence. We show that social networks that exist between users can be used to extract more sensitive location information from differentially private query results than expected when standard DP mechanisms are applied. To defend against such attacks, we introduced a generalized dependent differential privacy framework that incorporates probabilistic dependence relationship between data and provides rigorous privacy guarantees. We further propose a dependent perturbation mechanism and rigorously prove that it can achieve the privacy guarantees. Our evaluations over multiple large-scale real datasets and multiple query classes show that the dependent perturbation scheme performs significantly better than state-of-the-art approaches used for providing differential privacy.

Acknowledgement

We would like to thank the anonymous reviewers at NDSS 2016 for helpful feedback, and we are especially grateful to Reza Shokri for his guidance as our shepherd. This work was supported in part by NSF awards number CNS-1409415 and CNS-1423139. This work was also partially sponsored by US Army Research laboratory and the UK Ministry of Defense under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *CCS*, 2013.

[2] L. Backstrom, E. Sun, and C. Marlow, "Find me if you can: improving geographical prediction with social and spatial proximity," in *WWW*, 2010.

[3] I. Bilogrevic, K. Huguenin, S. Mihaila, R. Shokri, and J.-P. Hubaux, "Predicting users" motivations behind location check-ins and utility implications of privacy protection mechanisms," in *NDSS*, 2015.

[4] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to noninteractive database privacy," *Journal of the ACM*, 2013.

[5] A. Campan and T. M. Truta, "Data and structural k-anonymity in social networks," in *Privacy, Security, and Trust in KDD*, 2009.

[6] A. Chaabane, G. Acs, M. A. Kaafar *et al.*, "You are what you like! information leakage through users' interests," in *NDSS*, 2012.

[7] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *The Journal of Machine Learning Research*, 2011.

[8] R. Chen, B. C. Fung, P. S. Yu, and B. C. Desai, "Correlated network data publication via differential privacy," *The International Journal on Very Large Data Bases*, 2014.

[9] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *SIGKDD*, 2011.

[10] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[11] I. Dinur and K. Nissim, "Revealing information while preserving privacy," in *PODS*, 2003.

[12] C. Dwork, “Differential privacy,” in *Automata, languages and programming*, 2006.

[13] ——, “Differential privacy: A survey of results,” in *Theory and Applications of Models of Computation*, 2008.

[14] ——, “A firm foundation for private data analysis,” *Communications of the ACM*, 2011.

[15] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Springer Theory of cryptography*, 2006.

[16] C. Dwork and A. Smith, “Differential privacy for statistics: What we know and what we want to learn,” *Journal of Privacy and Confidentiality*, 2010.

[17] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, “Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing,” in *USENIX Security*, 2014.

[18] S. R. Ganta, S. P. Kasiviswanathan, and A. Smith, “Composition attacks and auxiliary information in data privacy,” in *SIGKDD*, 2008.

[19] N. Z. Gong, W. Xu, L. Huang, P. Mittal, E. Stefanov, V. Sekar, and D. Song, “Evolution of social-attribute networks: measurements, modeling, and implications using google+,” in *IMC*, 2012.

[20] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *Applied statistics*, 1979.

[21] X. He, A. Machanavajjhala, and B. Ding, “Blowfish privacy: Tuning privacy-utility trade-offs using policies,” in *SIGMOD*, 2014.

[22] S. Ji, W. Li, M. Srivatsa, and R. Beyah, “Structural data de-anonymization: Quantification, practice, and implications,” in *CCS*, 2014.

[23] D. Kifer and B.-R. Lin, “Towards an axiomatization of statistical privacy and utility,” in *PODS*, 2010.

[24] D. Kifer and A. Machanavajjhala, “No free lunch in data privacy,” in *SIGMOD*, 2011.

[25] ——, “A rigorous and customizable framework for privacy,” in *PODS*, 2012.

[26] N. Li, T. Li, and S. Venkatasubramanian, “t-closeness: Privacy beyond k-anonymity and l-diversity,” in *ICDE*, 2007.

[27] N. Li, W. Qardaji, D. Su, Y. Wu, and W. Yang, “Membership privacy: a unifying framework for privacy definitions,” in *CCS*, 2013.

[28] D. Liben-Nowell and J. Kleinberg, “The link-prediction problem for social networks,” *Journal of the American society for information science and technology*, 2007.

[29] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, “l-diversity: Privacy beyond k-anonymity,” *ACM Transactions on Knowledge Discovery from Data=*, 2007.

[30] F. D. McSherry, “Privacy integrated queries: an extensible platform for privacy-preserving data analysis,” in *SIGMOD*, 2009.

[31] P. Mittal, C. Papamanthou, and D. Song, “Preserving link privacy in social network based systems,” in *NDSS*, 2013.

[32] A. Narayanan and V. Shmatikov, “De-anonymizing social networks,” in *IEEE S&P*, 2009.

[33] S. Nilizadeh, A. Kapadia, and Y.-Y. Ahn, “Community-enhanced de-anonymization of online social networks,” in *CCS*, 2014.

[34] P. Sen and A. Deshpande, “Representing and querying correlated tuples in probabilistic databases,” in *ICDE*, 2007.

[35] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, “Quantifying location privacy,” in *IEEE S&P*, 2011.

[36] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, “Protecting location privacy: optimal strategy against localization attacks,” in *CCS*, 2012.

[37] M. Srivatsa and M. Hicks, “Deanonymizing mobility traces: Using social network as a side-channel,” in *CCS*, 2012.

[38] L. Sweeney, “k-anonymity: A model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2002.

[39] C. Task and C. Clifton, “A guide to differential privacy theory in social network analysis,” in *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining*, 2012.

[40] T. Zhu, P. Xiong, G. Li, and W. Zhou, “Correlated differential privacy: Hiding information in non-iid dataset,” *Information Forensics and Security, IEEE Transactions on*, 2013.

APPENDIX

A. Security Guarantees of DP

Integrating Eq. 1 for ϵ-differential privacy with $\mathcal{P}(D')$, we have $\sum_{D'} \mathcal{P}(D')\mathcal{P}(\mathcal{A}(D) = S) \leq e^{\epsilon} \sum_{D'} \mathcal{P}(D')\mathcal{P}(\mathcal{A}(D') = S)$, i.e., $\mathcal{P}(\mathcal{A}(D) = S) \leq e^{\epsilon}\mathcal{P}(\mathcal{A}(\cdot) = S)$. Combining with the definition of Leaked Information in Eq. 11, we obtain $\mathit{Leaked\ Information} = \sum_D P(D)P(\mathcal{A}(D) = S) \log \frac{P(\mathcal{A}(D)=S)}{P(\mathcal{A}(\cdot)=S)}\} \leq \epsilon$.

B. Formulation for ρ_{ij} in Eq. 18

We consider the general dependent relationships of tuples to analyze the second term of the RHS of Eq. 12 as

$$\max_{\mathbf{d}_{i_1},\mathbf{d}_{i_2}} \frac{\sum_{\mathbf{d}_j} P(D_j{=}\mathbf{d}_j|D_i{=}\mathbf{d}_{i_1}) \exp(-\frac{\|\widetilde{\mathbf{d}}_j - \mathbf{d}_j\|_1}{\sigma(\epsilon)})}{\sum_{\mathbf{d}_j} P(D_j{=}\mathbf{d}_j|D_i{=}\mathbf{d}_{i_2}) \exp(-\frac{\|\widetilde{\mathbf{d}}_j - \mathbf{d}_j\|_1}{\sigma(\epsilon)})} \leq$$

$$\max_{\mathbf{d}_{i_1},\mathbf{d}_{i_2}} \frac{\sum_{\mathbf{d}_j} P(D_j{=}\mathbf{d}_j|D_i{=}\mathbf{d}_{i_1}) \exp(-\frac{\|\widetilde{\mathbf{d}}_j - \mathbf{d}_j\|_1}{\sigma(\epsilon)})}{\exp(-\frac{\|\widetilde{\mathbf{d}}_j - \mathbf{d}_j^{\min}\|_1}{\sigma(\epsilon)})} \leq$$

$\max_{\mathbf{d}_{i_1}} \sum_{\mathbf{d}_j} P(D_j = \mathbf{d}_j|D_i = \mathbf{d}_{i_1}) \exp(\frac{\|\mathbf{d}_j - \mathbf{d}_j^{\min}\|_1}{\sigma(\epsilon)})$, where $\mathbf{d}_j^{\min}$ is the value of $\boldsymbol{d}_j$ that minimizes $\exp(\frac{\|\widetilde{\mathbf{d}}_j - \mathbf{d}_j\|_1}{\sigma(\epsilon)})$. In order to quantify the dependence coefficient which is applicable for any output value of $\widetilde{\mathbf{d}}_j$, we further have $\exp(\frac{\|\mathbf{d}_j - \mathbf{d}_j^{\min}\|_1}{\sigma(\epsilon)}) \leq \exp(\frac{\|\mathbf{d}_j - \mathbf{d}_j^*\|_1}{\sigma(\epsilon)})$, where $\mathbf{d}_j^*$ maximizes $\|\mathbf{d}_j - \mathbf{d}_j^*\|_1$. Substituting $\mathbf{d}_j^{\min}$ with $\mathbf{d}_j^*$, we obtain Eq. 18.

C. Proof for Sequential Composition Theorem for DDP For any sequence r of outcomes $r_t \in \mathrm{Region}(\mathcal{A}_t)$ with the same dependence relationship $\mathcal{R}$, the probability of output r from the sequence of $\mathcal{A}_t(D)$ is $Pr(\mathcal{A}(D) = r) = \prod_t Pr(\mathcal{A}_t(D) = r_t)$. Applying the definition of DDP for each $\mathcal{A}_t$, we have $\prod_t Pr(\mathcal{A}_t(D) = r_t) \leq \prod_t Pr(\mathcal{A}_t(D') = r_t) \times \prod_t \exp\left(\frac{\epsilon_t}{DS^Q} \times |D - D'|\right) \leq Pr(\mathcal{A}(D') = r) \times \exp\left(\sum_t \epsilon_t\right)$.

D. Proof for Parallel Composition Theorem for DDP

For D and D', let $D_t : D \cap D_t$ and $D'_t = D \cap D_t$ with the same dependence relationship $\mathcal{R}$, for any sequence r of outcomes $r_t \in \mathrm{R}(\mathcal{A}_t)$, the probability of output r from the sequence of $\mathcal{A}_t(D)$ is $Pr(\mathcal{A}(D) = r) = \prod_t Pr(\mathcal{A}_t(D_i) = r_t)$ Applying the definition of DDP for each $\mathcal{A}_t$, we have $\prod_t Pr(\mathcal{A}_t(D) = r_t) \leq \prod_t Pr(\mathcal{A}_t(D'_t) = r_t) \times \prod_t \exp\left(\frac{\epsilon_t}{DS^Q} \times |D_t - D'_t|\right) \leq Pr(\mathcal{A}(D') = r) \times \exp\left(\frac{\max\limits_t \epsilon_t}{DS^Q} \times |D - D'|\right) \leq Pr(\mathcal{A}(D') = r) \times \exp\left(\max\limits_t \epsilon_t\right)$.

E. Dependent Sensitivity for Any Query Q

As ρ_{ij} evaluates the extent of dependence between D_i and D_j, the modification of D_i would imply modification of D_j as $\rho_{ij}\Delta D_j$. Therefore, for any query function Q, we have the corresponding sensitivity for D_j as $\rho_{ij}\Delta Q_j$. Furthermore, we can prove $DS_i^Q = \max_{d_{i1},d_{i2}} \|Q([D_1, \cdots, d_{i1}, \cdots]) - Q([D_1, \cdots, d_{i2}, \cdots])\|_1 = \max_{d_{i1},d_{i2}} \int_{d_{i1}}^{d_{i2}} \frac{\partial Q(D)}{\partial D_i} dD_i + \int_{d_{j(i)}^{\min}}^{d_{j(i)}^{\max}} \frac{\partial Q(D)}{\partial D_j} dD_j + \cdots \leq \Delta Q_i + \frac{\Delta D_{j(i)}}{\Delta D_j}\Delta Q_j + \cdots \leq \Delta Q_i + \rho_{ij}\Delta Q_j = \sum_{j=C_{i1}}^{C_{iL}} \rho_{ij}\Delta Q_j$. Therefore, the global sensitivity for publishing any query function Q on a dependent dataset is $DS^Q = \max_i DS_i^Q = \sum_j \rho_{ij}\Delta Q_j$.

F. (α, β)-Accuracy Guarantee for DDP

$P(\max|\mathcal{A}(D) - Q(D)| > \alpha) \leq \beta \Longrightarrow P(\max|Lap(\frac{DS^Q}{\epsilon})| > \alpha) \leq \beta \Longrightarrow P(Lap(\frac{DS^Q}{\epsilon}) > \alpha) + P(Lap(\frac{DS^Q}{\epsilon}) < -\alpha) \leq \beta \Longrightarrow 2\int_{\alpha}^{\infty} t\exp(-\frac{\epsilon t}{DS^Q})dt \leq \beta \Longrightarrow \exp(-\frac{\epsilon\alpha}{DS^Q}) \leq \beta$.

G. Proof for the Transform Invariance Axiom

$P(\mathcal{B}(\mathcal{A}(\mathcal{D})) = O|\mathbf{d}_{i1}) = \sum_D P(\mathcal{B}(\mathcal{A}(D)) = O)P(\mathcal{D} = D|\mathbf{d}_{i1}) = \sum_D \sum_S P(\mathcal{B}(S) = O)P(\mathcal{A}(D = S|\mathbf{d}_{i1}) \leq e^{\epsilon} \sum_D P(\mathcal{B}(\mathcal{A}(D)) = O)P(\mathcal{D} = D|\mathbf{d}_{i2}) = e^{\epsilon} P(\mathcal{B}(\mathcal{A}(\mathcal{D})) = O|\mathbf{d}_{i2})$.

H. Proof for the Convexity Axiom

$P(\mathcal{A}^p(\mathcal{D}) = S|\mathbf{d}_{i1}) = pP(\mathcal{A}_1(\mathcal{D}) = S|\mathbf{d}_{i1}) + (1-p)P(\mathcal{A}_2(\mathcal{D}) = S|\mathbf{d}_{i1}) \leq e^{\epsilon} pP(\mathcal{A}_1(\mathcal{D}) = S|\mathbf{d}_{i1}) + e^{\epsilon}(1-p)P(\mathcal{A}_2(\mathcal{D}) = S|\mathbf{d}_{i1}) = e^{\epsilon} P(\mathcal{A}^p(\mathcal{D}) = S|\mathbf{d}_{i2})$.