IEEE Transactions on Kowledge and Data Engineering (Volume :26, Issue :7) , July 2014

A Supermodularity-Based Differential Privacy Preserving Algorithm for Data Anonymization

Mohamed R. Fouad, Khaled Elbassioni, *Member, IEEE,*
and Elisa Bertino, *Fellow, IEEE*

E. Bertino and M. Fouad are with Purdue Universaity.

K. Elbassioni is with Masdar Institute of Science and Technology.

Digital Object Indentifier 10.1109/TKDE.2013.107

Abstract

Maximizing data usage and minimizing privacy risk are two conflicting goals. Organizations always apply a set of transformations on their data before releasing it. While determining the best set of transformations has been the focus of extensive work in the database community, most of this work suffered from one or both of the following major problems: scalability and privacy guarantee. *Differential Privacy* provides a theoretical formulation for privacy that ensures that the system essentially behaves the same way regardless of whether any individual is included in the database.

In this paper, we address both scalability and privacy risk of data anonymization. We propose a scalable algorithm that meets differential privacy when applying a specific random sampling. The contribution of the paper is two-fold: (1) we propose a personalized anonymization technique based on an aggregate formulation and prove that it can be implemented in polynomial time, and (2) we show that combining the proposed aggregate formulation with specific sampling gives an anonymization algorithm that satisfies differential privacy. Our results rely heavily on exploring the supermodularity properties of the risk function, which allow us to employ techniques from convex optimization.

Through experimental studies we compare our proposed algorithm with other anonymization schemes in terms of both time and privacy risk.

Index Terms

Differential Privacy, security, risk management, data sharing, data utility, anonymity, scalability.

I. Introduction

Although data disclosure is advantageous for many reasons such as research purposes, it may incur some risk due to security breaches. Releasing health care information, for example, though useful in improving the quality of service that patients receive, raises the chances of identity exposure of the patients. Disclosing the minimum amount of

information (or no information at all) is compelling specially when organizations try to protect the privacy of individuals. To achieve such a goal, the organizations typically try to hide the identity of an individual to whom data pertains and apply a set of transformations to the microdata before releasing it. These transformations include (1) data suppression (disclosing the value $\perp$, instead), (2) data generalization (releasing a less specific variation of the original data such as in [32]), and (3) data perturbation (adding noise directly to the original data values such as in [25]). Studying the risk-utility tradeoff has been the focus of much research. Resolving this tradeoff by determining the optimal data transformation has suffered from two major problems, namely, scalability and privacy risk. To the best of our knowledge, most of the work in determining the optimal transformation to be performed on a database before it gets disclosed is inefficient in the sense that increasing the table dimension will substantially exacerbate the performance. Moreover, data anonymization techniques [31], [24], [27], [4] do not provide enough theoretical evidence that the disclosed table is immune from security breaches. Anonymization techniques include (1) hiding the identities by making each record indistinguishable from at least $k-1$ other records [31] (k-anonymity), (2) ensuring that the distance between the distribution of sensitive attributes in a class of records and the distribution of them in the whole table is no more than t [24] (t-closeness), and (3) ensuring that there are at least l distinct values for a given sensitive attribute in each indistinguishable group of records [27] (l-diversity). Indeed, these tehniques do not completely prevent re-identification [23]. It is shown in [1] that the k-anonymity [31] technique suffers from the curse of dimensionality: the level of information loss in k-anonymity may not be acceptable from a data mining point of view because the specifics of the inter-attribute behavior have a very powerful revealing effect in the high dimensional case.

A realization of t-closeness is proposed in [6], called SABRE. It partitions a table

into buckets of similar sensitive attribute values in a greedy fashion, then it redistributes tuples from each bucket into dynamically configured equivalence classes (EC). SABRE adopts the information loss measures [16], [5] for each EC as a unit rather than treating released records individually. Moreover, although experimental evaluation demonstrates that SABRE is superior to schemes that merely applied algorithms tailored for other models to t-closeness in terms of quality and speed, it lacks the theoretical foundations for privacy guarantees and efficiency.

In [13], an heuristic called ARUBA is proposed to address the tradeoff between data utility and data privacy. Although the proposed algorithm determines a personalized optimum data transformation based on predefined risk and utility models, it provides neither scalability nor theoretical foundations for privacy guarantees.

The notion of *Differential privacy* [7], [9] has become very popular in the database communities. It requires that the distribution of outcomes of a computation does not change significantly when one individual changes their input data. A randomized query satisfies differential privacy if the likelihood of obtaining a certain answer from a database x is not "too" different from the likelihood of obtaining the same answer from other databases which differ from x for only one individual.

Our Contribution: In this paper we address the problem of maximizing the utility of data disclosure while maintaining its risk below a certain acceptable threshold. The main focus of the paper is to provide a theoretical study of our prior work [13] and extend it to satisfy *Differential Privacy*. We propose a differential privacy preserving algorithm for data disclosure. The algorithm provides personalized transformation on individual data items based on the risk tolerance of the person to whom the data pertains. We first consider the problem of obtaining such a transformation for each record individually without taking the differential privacy constraint into consideration. We propose two different methods to deal with this hardness: (1) an approximation

algorithm that we prove (under some conditions) to produce a data transformation within constant guarantees of the optimum, and (2) a slightly modified variant of the formulation in [13] that can be used to get a polynomial-time algorithm for the data transformation. For achieving these two results, we explore the fact that the risk function is a ratio with a *supermodular* denominator. Thus, we get a fractional program whose solution can be reduced to a number of supermodular function maximization problems, each of which can be solved in polynomial time. In addition, we consider the problem of obtaining a set of data transformations, one for each record in the database, in such a way that satisfies differential privacy and at the same time maximizes (minimizes) the average utility (risk) per record. Towards this end, we adopt the *exponential mechanism* recently proposed in [28]. The main technical difference that distinguishes our application of this mechanism from the previous applications (e.g., in [28], [19]) is the fact that in our case *the output set is also a function of the input*, and hence it changes if a record is dropped from the database. In fact, a simple example is presented to show that it is not possible to obtain differential privacy without sacrificing utility maximization. To resolve this issue, we sample only from "frequent elements", that is, those records generalizing a large number of records in the database and show that, this way, differential privacy can be achieved with any desired success probability arbitrarily close to 1. Another technical difficulty that we need to overcome is how to perform the sampling needed by the exponential mechanism. Again, we explore the supermodularity of the (denominator of the) risk function to show that such sampling can be done efficiently even for a *large* number of attributes. Note that **the proofs of some Theorems, Lemmas and Propositions are omitted for lack of space and could be found in [12]**.

The rest of the paper is organized as follows. In Section II we formally describe our model for data generalization. We present the aggregate optimization model and the *approximation algorithm*, in Sections II-B and II-C, respectively. Differential Privacy

is investigated in Section III, where we present the exponential sampling algorithm in Section III-C, and show how to perform the sampling in Section III-D. Experimental results that show the superiority of our proposed algorithm over existing algorithms are reported in Section IV. Section V surveys the related work and, finally, Section VI presents some concluding remarks and future directions.

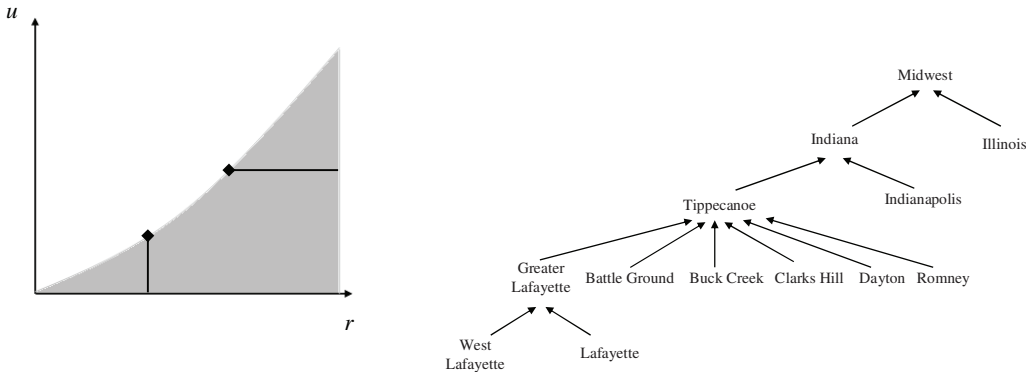
II. The Data Generalization Model

In this section, we recall the data transformation model proposed in [13]. For reasons that will become clear soon, we shall refer to this model as the *threshold model*. Due to lack of space, some details and proofs are omitted. The complete information could be found in the extended version of the paper [12]. Unfortunately, when the number of attributes k is part of the input, the underling optimization problem cannot be solved in polynomial time unless P=NP. In the next subsections, we propose two different methods to deal with such NP-hardness. Specifically, in Section II-B, we modify the model by bringing the constraint on the utility into the objective and show that this modified objective can be optimized in polynomial time. In section II-C, we develop an approximation algorithm for the threshold model which can be used to produce a solution within a constant factor of the optimal utility, yet violating the risk constraint by a constant factor.

A. The Threshold Formulation

The model described in this section is based on [13].

1) The Informal Model: The relationship between the risk and expected utility is schematically depicted in Fig.1(a) which displays different instances of a disclosed table by their 2-D coordinates (r, u) representing their risk and expected utility, respectively. In other words, different data generalization procedures pose different utility and risk which lead to different locations in the (r, u)-plane. The shaded region in the figure



(a) (b)

Fig. 1. (a) Space of disclosure rules and their risk and expected utility, (b) A partial VGH for the *city* attribute.

corresponds to the set of feasible points (r, u) (that is, the risk and utility are achievable by a certain disclosure policy) whereas the unshaded region corresponds to the infeasible points. The vertical line corresponds to all instances whose risk is fixed at a certain level. Similarly, the horizontal line corresponds to all instances whose expected utility is fixed at a certain level. Since the disclosure goal is to obtain both low risk and high expected utility, naturally we are most interested in these disclosure policies occupying the boundary of the shaded region. Policies in the interior of the shaded region can be improved upon by projecting them to the boundary.

The vertical and horizontal lines suggest the following way of resolving the risk-utility tradeoff. Assuming that it is imperative that the risk remains below a certain level c, the optimization problem becomes

$$\max\ u \quad \text{subject to} \quad r \leq c.$$

2) The Formal Model: More formally, we assume that we have k attributes, and let $\mathcal{L}_1, \ldots, \mathcal{L}_k$ be the corresponding value generalization hierarchies (VGH's). We will consider VGH's that allow for modeling taxonomies (see Fig.1(b) for an example of

the VGH for the *city* attribute). Each such $\mathcal{L}_i$, equipped with the hierarchical relation $\succeq_i$, defines a *join semi-lattice*, that is, for every pair $x, x' \in \mathcal{L}_i$, the least upper bound $x \vee x'$ exists, where the relation $x \succeq_i x'$ means that x is a generalization of x' in the corresponding VGH. Let $\mathcal{L} = \mathcal{L}_1 \times \ldots \times \mathcal{L}_k$ be the semi-lattice defined by the product such that for every $\mathbf{x} = (x_1, \ldots, x_k), \mathbf{x}' = (x'_1, \ldots, x'_k) \in \mathcal{L}$, $\mathbf{x} \succeq \mathbf{x}'$ if and only if $x_i \succeq_i x'_i$ for all $i \in [k] = \{1, \ldots, k\}$. The unique upper bound of $\mathcal{L}$ corresponds to the most general element and is denoted by $\perp = (\perp, \ldots, \perp)$. For $\mathbf{x} \in \mathcal{L}$ and $i \in [k]$, let us denote by ${x_i}^+ = \{y \in \mathcal{L}_i : \ y \succeq_i x_i\}$ the *chain* (that is, total order) of elements that generalize x_i, and let $\mathbf{x}^+ = x_1^+ \times \ldots \times x_k^+$ be the chain product that generalizes $\mathbf{x}$.

Fig. 2 shows an example of a generalization lattice formed on a two-attribute record. It is formed by the product of two chains: "City" chain with relations $\perp \succeq_1$ "Indiana" $\succeq_1$"Tippicanoe" $\succeq_1$ "Dayton"; and "Race" chain with relations $\perp \succeq_2$ "Asian" $\succeq_2$ "Chinese".

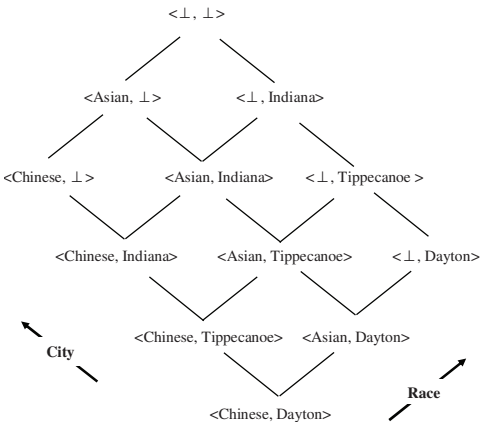


Fig. 2. A 2D lattice.

When considering a chain $\mathcal{C}_i$, we will assume, without loss of generality, that $\mathcal{C}_i = \{0, 1, 2, \ldots, h_i\}$, where h_i is the height of the VGH corresponding to attribute i and the ordering on $\mathcal{C}_i$ is given by the natural ordering on the integers.

The utility function: The utility is defined by non-negative monotonically non-increasing functions $d_1 : \mathcal{L}_1 \to \mathbb{R}_+, \ldots, d_k : \mathcal{L}_k \to \mathbb{R}_+$, (i.e., $d_i(x) \leq d_i(y)$ for $x, y \in \mathcal{L}_i$ such that $x \succeq_i y$). For $\mathbf{x} \in \mathcal{L}$, the utility is given by $u(\mathbf{x}) = \sum_{i=1}^{k} d_i(x_i)$. For instance, in [13, eq.(5)], $d_i(x_i) = \frac{1}{n_i(x_i)}$, and in [13, eq.(6)], $d_i(x_i) = \ln(\frac{n_i(\perp)}{n_i(x_i)})$; for $x_i \in \mathcal{L}_i$, where $n_i(x_i)$ is the number of leaf nodes of the VGH subtree rooted at x_i.

The risk function: We use the risk model proposed in [13]. For a record $\mathbf{a}$, given the side database Θ, the risk of a generalization $\mathbf{x} \in \mathbf{a}^+$ is given by $r^{\mathbf{a}}(\mathbf{x}) = r^{\mathbf{a}}(\mathbf{x}, \Theta) = \frac{\Phi^{\mathbf{a}}(\mathbf{x})}{|\rho(\mathbf{x},\Theta)|}$. The function $\Phi^{\mathbf{a}}(x) = \sum_{i=1}^{k} w_i^{\mathbf{a}}(x_i)$, where $w_i^{\mathbf{a}} : a_i^+ \to \mathbb{R}_+$ is a non-negative monotonically non-increasing function, represents the sensitivity of the ith attribute to the user owning $\mathbf{a}$, and $\rho(\mathbf{x}, \Theta) = \{\mathbf{t} \in \Theta \mid \mathbf{t} \preceq \mathbf{x}\}$ is the set of records in the external database Θ consistent with the disclosed generalization $\mathbf{x}$. In [13, Model I], $w_i^{\mathbf{a}}(x_i)$ is either 0 if $x_i = \perp$ or some fixed weight $w_i^{\mathbf{a}}$ if $x_i \neq \perp$; in [13, Model II], $w_i^{\mathbf{a}}(x_i) = \frac{1}{k}$ for all $x_i \in \mathbf{a}_i^+$.

Definition 1: The Threshold Model

In data privacy context, given a record $\mathbf{a} = (a_1, a_2, \cdots, a_i, \cdots, a_k)$, a utility measure $u(\mathbf{x})$, and a risk measure $r(\mathbf{x})$, the threshold model determines the generalization $\mathbf{x} \in \mathbf{a}^+$ that maximizes $u(\mathbf{x})$ subject to $r(\mathbf{x}) \leq c$, where $c \in \mathbb{R}_+$ is a given parameter and $\mathbf{a}^+$ is the set of all generalizations of the record $\mathbf{a}$.

B. A Polynomial-Time Solvable Optimization Model: The Aggregate Formulation

1) Preliminaries: Our results in Sections II-B.2, II-C, and III-D are mainly based on the fact that the risk function exhibits certain *submodularity* properties. The very desirable property of submodular (respectively, supermodular) functions is that they can be minimized (respectively, maximized) in polynomial time [18]. In this section we summarize the basic facts that we need about such functions.

Definition 2: A function $f : \mathcal{C} \to \mathbb{R}$ on a chain (or a lattice) product $\mathcal{C} = \mathcal{C}_1 \times \ldots \times \mathcal{C}_k$ is said to be *monotonically non-decreasing* (or simply monotone) if $f(\mathbf{x}) \geq f(\mathbf{x}')$ whenever

$\mathbf{x} \succeq \mathbf{x}'$, and *monotonically non-increasing* (or anti-monotone) if $f(\mathbf{x}) \leq f(\mathbf{x}')$ whenever $\mathbf{x} \succeq \mathbf{x}'$.

Definition 3: A function $f : \mathcal{C} \to \mathbb{R}$ is said to be *supermodular* if

$$f(\mathbf{x} \wedge \mathbf{x}') + f(\mathbf{x} \vee \mathbf{x}') \geq f(\mathbf{x}) + f(\mathbf{x}'), \tag{1}$$

for every pair $\mathbf{x}$ and $\mathbf{x}'$ in $\mathcal{C}$, where $\mathbf{x} \wedge \mathbf{x}'$ is the meet (the greatest lower bound) of $\mathbf{x}$ and $\mathbf{x}'$, and $\mathbf{x} \vee \mathbf{x}'$ is the join (the least upper bound). f is *submodular* if the reverse inequality in (1) holds for every pair $\mathbf{x}$ and $\mathbf{x}'$ in $\mathcal{C}$ (that is, if and only if $-f$ is supermodular).

To show that a given function is supermodular, the following proposition will be useful.

Proposition 1: A function $f : \mathcal{C} \to \mathbb{R}$ is supermodular if and only if, for any $i \in [k]$, for any $z \in \mathcal{C}_i$, and for any $\mathbf{x} \in \mathcal{C}_1 \times \ldots \times \mathcal{C}_{i-1} \times \{z\} \times \mathcal{C}_{i+1} \times \ldots \times \mathcal{C}_k$; the difference

$$\partial_f(\mathbf{x}, i, z) \stackrel{\text{def}}{=} f(\mathbf{x} + \mathbf{e}^i) - f(\mathbf{x})$$

as a function of $\mathbf{x}$ is monotonically non-decreasing in $\mathbf{x}$, where $\mathbf{e}^i$ is the i^{th} unit vector[1].

When restricted on the chain product $\mathbf{a}^+$, for $\mathbf{a} \in \mathcal{L}$, the utility function defined in Section II-A.2 is *modular*, that is, for $\mathbf{x}, \mathbf{x}' \in \mathbf{a}^+$, inequality (1) holds as an equality. Indeed, $u(\mathbf{x} \wedge \mathbf{x}') + u(\mathbf{x} \vee \mathbf{x}') = \sum_{i=1}^k d_i(\min\{x_i, x_i'\}) + \sum_{i=1}^k d_i(\max\{x_i, x_i'\}) = \sum_{i=1}^k d_i(x_i) + \sum_{i=1}^k d_i(x_i') = u(\mathbf{x}) + u(\mathbf{x}')$. The following proposition will be used to establish that a certain combination of risk and utility is supermodular.

Proposition 2:

(i) The function $g(\mathbf{x}) = |\rho(\mathbf{x}, \Theta)|$ over $\mathbf{x} \in \mathbf{a}^+$, is supermodular and monotonically non-decreasing.

(ii) Let $p : \mathbf{a}^+ \to \mathbb{R}_+$ be a monotonically non-increasing supermodular function and $q : \mathbf{a}^+ \to \mathbb{R}_+$ be a non-negative monotonically non-increasing modular function. Then, $h(\mathbf{x}) = q(\mathbf{x})p(\mathbf{x})$ over $\mathbf{x} \in \mathbf{a}^+$ is monotonically non-increasing supermodular.

[1]The notation $\mathbf{x} + \mathbf{e}^i$ here means a one-step generalization for $\mathbf{x}$ w.r.t. attribute i.

Repeated application of Proposition 2 yields the following.

Corollary 1: For any $\kappa \in \mathbb{Z}_+$, the function $h(\mathbf{x}) = \Phi^{\mathbf{a}}(\mathbf{x})(u(\mathbf{x}))^{\kappa}$ over $\mathbf{x} \in \mathbf{a}^+$ is supermodular.

2) The Modified Aggregate Model: One other way to deal with the NP-hardness of the threshold formulation is to use the following model which aggregates both risk and utility into one objective function: Given a record $\mathbf{a}$, it is required to find a generalization $\mathbf{x} \in \mathbf{a}^+$, that maximizes the "Lagrangian" relaxation

$$f^{\mathbf{a}}(\mathbf{x}) = \frac{\lambda}{r^{\mathbf{a}}(\mathbf{x})} + (u(\mathbf{x}))^{\kappa}, \tag{2}$$

where $\lambda \in \mathbb{R}_+$ and $\kappa \in \mathbb{Z}_+$ are given design parameters, which we use to control how much importance to give to utility maximization/risk minimization. It is worth noting that throughout the rest of this paper we will use the term "utility" interchangeably to denote the utility function $u(\mathbf{x})$ and the aggregate objective function (2).

Theorem 1: Assuming rational input, $\alpha^* = \max_{\mathbf{x} \in \mathbf{a}^+} f^{\mathbf{a}}(\mathbf{x})$ can be computed in polynomial time in $\sum_{i=1}^{k} |a_i^+|$, $|\Theta|$, and the bit length of the input weights.

3) Optimization On a Ring Family: Since it is easier to work on the 0/1-hypercube (and moreover there are available software for maximizing supermodular/minimizing submodular set-functions), we describe here how to reduce the optimization problem over a chain product to one over the cube.

By Birkhoff's representation theorem (for e.g., [17, Chapter II]), we may regard a chain product $\mathcal{C}$ as a sublattice of the Boolean lattice. More precisely, we consider the set of *join-irreducible elements*

$$\begin{aligned} \mathcal{J} \;=\; & \{(1,0,\ldots,0),(2,0,\ldots,0),\ldots,(h_1,0,\ldots,0), \\ & (0,1,\ldots,0),(0,2,\ldots,0),\ldots,(0,h_2,\ldots,0),\ldots, \\ & (0,0,\ldots,1),(0,0,\ldots,2),\ldots,(0,0,\ldots,h_k)\}, \end{aligned}$$

and, for $\mathbf{x} \in \mathcal{C}$, define $S(\mathbf{x}) = \{\mathbf{y} \in \mathcal{J} : \mathbf{y} \preceq \mathbf{x}\}$. Then a supermodular (respectively, submodular, or modular) function $f : \mathcal{C} \to \mathbb{R}$ gives rise to another supermodular (respectively, submodular, or modular) function $g : \mathcal{F} \to \mathbb{R}$, defined over the *ring family*[2] $\mathcal{F} = \{S(\mathbf{x}) : \mathbf{x} \in \mathcal{C}\}$ as $g(S(\mathbf{x})) = f(\mathbf{x})$.

Thus, we can maximize a supermodular function on $\mathcal{C}$ by solving a maximization problem for a supermodular set-function over a ring family.

Using known techniques (for e.g., [18, Chapter 10] and [30, Chapter 10]), the problem can be further reduced to maximizing a supermodular function over the hypercube $2^{\mathcal{J}}$. For completeness, let us sketch the reduction from [30] here. For $\mathbf{v} \in \mathcal{J}$, denote by $N_{\mathbf{v}}$ the largest member of $\mathcal{F}$ not containing $\mathbf{v}$. For $X \subseteq \mathcal{J}$, define the closure $\overline{X} = S(\vee_{\mathbf{x} \in X} \mathbf{x})$. Equivalently, $\overline{X}$ is the smallest member in $\mathcal{F}$ that contains X. Let us now extend the function $g : \mathcal{F} \to \mathbb{R}$ into the function $\bar{g} : 2^{\mathcal{J}} \to \mathbb{R}$ by setting

$$\bar{g}(X) = g(\overline{X}) + b(X) - b(\overline{X}) \qquad \text{for } X \subseteq \mathcal{J},$$

where $b : 2^{\mathcal{J}} \to \mathbb{R}$ is the modular function given by

$$b(\{\mathbf{v}\}) = \max\{0, g(N_{\mathbf{v}} \cup \{\mathbf{v}\}) - g(N_{\mathbf{v}})\} \qquad \text{for } \mathbf{v} \in \mathcal{J},$$

and $b(Y) = \sum_{\mathbf{v} \in Y} b(\mathbf{v})$, for any $Y \subseteq \mathcal{J}$. As shown in [30], the following holds: (1) $\bar{g}$ is supermodular, and (2) for all $X \subseteq \mathcal{J}$, $g(\overline{X}) \geq \bar{g}(X)$. In particular, $X \in \operatorname{argmax} \bar{g}$ implies $\overline{X} \in \operatorname{argmax} g$. Thus, we can maximize g over $\mathcal{F}$ by maximizing $\bar{g}$ over the hypercube. Alternatively [18], we may also use the extension $\bar{g}(X) = g(\overline{X}) - K|\overline{X} \setminus X|$, for sufficiently large $K > \max_{X \subseteq \mathcal{J},\ \mathbf{v} \in \mathcal{J}} g(X \cup \{\mathbf{v}\}) - g(X)$.

[2]A set family $\mathcal{F}$ is called a ring family if $X,\ Y \in \mathcal{F} \ \Rightarrow\ X \cap Y,\ X \cup Y \in \mathcal{F}$.

C. An Approximation Algorithm

When the risk threshold c is "small"[3], we can use convex optimization, as described in this section, to obtain a generalization of the given record $\mathbf{a}$ that approximately maximizes the utility and is only within a constant factor from the risk threshold. We need a few more preliminaries first.

The Lovász extension [18]: Let V be a finite set of size n, and $\mathcal{F} \subseteq 2^V$ be a ring family over V, such that $\emptyset, V \in \mathcal{F}$. We assume that the family $\mathcal{F}$ is defined by a *membership oracle*, that is, an algorithm that can decide for a given $X \subseteq V$ whether $X \in \mathcal{F}$ or not. For instance, such an oracle can be easily defined for the family $\mathcal{F}$ given in the previous section (where $V = \mathcal{J}$), since checking if $X \in \mathcal{F}$ is equivalent to checking if $S(\vee_{\mathbf{x} \in X} \mathbf{x}) = X$.

For $X \subseteq V$, denote by $\chi(X) \in \{0,1\}^V$ the characteristic vector of X, that is, $\chi_i(X) = 1$ if and only if $i \in X$. Let us denote by $P(\mathcal{F}) = \text{conv}\{\chi(X) : \ X \in \mathcal{F}\}$ the convex hull of the characteristic vectors of the sets in $\mathcal{F}$. Given $\mathbf{x} \in [0,1]^V$, and writing $U_i(\mathbf{x}) = \{j \in V : \ x_j \geq x_i\}$, for $i = 1, \ldots, n$, one can easily check that $\mathbf{x} \in P(\mathcal{F})$ if and only if $U_i(\mathbf{x}) \in \mathcal{F}$ for all $i \in [n]$. Thus, a membership oracle for $P(\mathcal{F})$ can be obtained from the given membership oracle for $\mathcal{F}$.

Given a set function $f : \mathcal{F} \to \mathbb{R}$ over $\mathcal{F}$, the Lovász extension $\hat{f} : P(\mathcal{F}) \to \mathbb{R}$ of f, is defined as follows: For any $\mathbf{x} \in P(\mathcal{F})$, assuming without loss of generality, that $x_1 \geq x_2 \geq \cdots \geq x_n$ and defining $x_{n+1} = 0$, then $\hat{f}(\mathbf{x}) = \sum_{i=1}^{n} (x_i - x_{i+1}) \big(f(\{1, \ldots, i\}) - f(\emptyset) \big) + f(\emptyset)$. Equivalently, $\hat{f} = \mathbb{E}[f(\{i : \ x_i > \lambda\})]$ for a randomly chosen $\lambda \in [0,1]$. It is known (for e.g., [18, Chapter 10], and [30, Chapter 10]) that f is supermodular (respectively, submodular) over $\mathcal{F}$, if and only if $\hat{f}$ is *concave* (respectively, *convex*)

[3]As we shall see, "small" here means a small multiple of the minimum possible risk $\nu'(k) = \min_{\mathbf{x} \in \mathbf{a}^+} r(\mathbf{x})$. Note that the result in Theorem 2 holds for $\theta \geq 1/(\sigma_1 + \sigma_2 - \sigma_1 \sigma_2)$. Thus, increasing the value of σ_2 allows us more flexibility in choosing c, but at the cost of worsening the approximation factor.

over $P(\mathcal{F})$. In particular, the extension of a modular function is *linear*.

Randomized rounding of a vector in the extension: Let $f : \mathcal{F} \to \mathbb{R}$ be a set function and $\hat{f}$ be its Lovász extension. Given a vector $\hat{\mathbf{x}}$ from $P(\mathcal{F})$, we can get back a point in the discrete domain $\mathcal{F}$ as follows. Assuming $\hat{x}_1 \geq \hat{x}_2 \geq \cdots \geq \hat{x}_n$, for $i = 1, \ldots, n-1$, we return the characteristic vector of the set $\{1, \ldots, i\}$ with probability $\hat{x}_i - \hat{x}_{i+1}$, return the vector $\underline{1}$ of all ones with probability $\hat{x}_n$, and return the vector $\underline{0}$ of all zeros with the remaining probability $1 - \hat{x}_1$. Let $RR(\hat{\mathbf{x}})$ be the random set returned by this procedure. It is easy to see that if $X = RR(\hat{\mathbf{x}})$, then $\mathbb{E}[f(X)] = \hat{f}(\hat{\mathbf{x}})$.

Example 1: Consider the 2D lattice in Fig. 2. Let us assign numbers $0, 1, 2, 3$ respectively to "Dayton", "Tippecanoe", "Indiana", $\perp$, and $0, 1, 2$ respectively to "Chinese", "Asian", $\perp$. So the set $\mathcal{J}$ of join-irreducibles is $\{(1,0), (2,0), (3,0), (0,1), (0,2)\}$; for convenience, let us rename elements of $\mathcal{J}$ (in order) as a, b, c, d, e. The corresponding sets in the ring family $\mathcal{F}$ are:

$$S(0,0) = \emptyset,\ S(1,0) = \{a\},\ S(2,0) = \{a,b\},\ S(3,0) = \{a,b,c\},$$
$$S(0,1) = \{d\},\ S(1,1) = \{a,d\},\ S(2,1) = \{a,b,d\},\ S(3,1) = \{a,b,c,d\},$$
$$S(0,2) = \{d,e\},\ S(1,2) = \{a,d,e\},\ S(2,2) = \{a,b,d,e\},\ S(3,2) = \{a,b,c,d,e\}.$$

Let us pick a point $\mathbf{x} \in P(\mathcal{F}) \subseteq [0,1]^{\mathcal{J}}$: for instance $\mathbf{x} = \frac{1}{3}(1,0,0,0,0) + \frac{1}{6}(1,1,0,1,0) + \frac{1}{2}(1,0,0,1,1) = (1, \frac{1}{6}, 0, \frac{2}{3}, \frac{1}{2})$ (is in the convex hull of the characteristic vectors of the sets $S(1,0)$, $S(2,1)$ and $S(1,2)$). Then after ordering the coordinates we have $\mathbf{x} = (1, \frac{2}{3}, \frac{1}{2}, \frac{1}{6}, 0)$ (corresponding to the order a, d, e, b, c). The sets that can be returned by randomized rounding are $\{a\}$, $\{a,d\}$, $\{a,d,e\}$, $\{a,b,d,e\}$ and $\{a,b,c,d,e\}$. Suppose that $\{a,d,e\}$ was returned. Then since $S(1,2) = \{a,d,e\}$, the corresponding vector returned from the lattice is $(1,2) =$ (Tippecanoe, $\perp$).

Now we can state our result for this section. The proof of the dual problem could be found in [12].

Theorem 2: Consider a record $\mathbf{a}$ in the database. Let $\nu'(k) = \min_{\mathbf{x}\in\mathbf{a}^+} r(\mathbf{x})$ and suppose that the utility threshold $c = \nu'(k)/\theta$, for some constant $\theta \in (0,1)$. Then there is an algorithm that outputs in expected polynomial time an element $\mathbf{x} \in \mathbf{a}^+$ such that

$$\mathbb{E}\left[u^{\mathbf{a}}(\mathbf{x})\right] \geq z^* \text{ and } r^{\mathbf{a}}(\mathbf{x}) \leq \frac{\sigma_2(1+\epsilon)}{\sigma_1}c,$$

for any constants $\epsilon > 0$, $\sigma_1 \in (0,1)$, and $\sigma_2 > 1$ such that $\frac{1-\theta}{1-\theta\sigma_1} + \frac{1}{\sigma_2} < 1$, where $z^* = \max_{\mathbf{x}'\in\mathbf{a}^+,\ r^{\mathbf{a}}(\mathbf{x}')\leq c} u^{\mathbf{a}}(\mathbf{x})$ is the value of an optimal solution in the threshold model.

Algorithm 1 Approx$(\mathbf{a}, \epsilon, \theta, \sigma_1, \sigma_2)$

Input: a record $\mathbf{a} \in \mathcal{D}$, real numbers $\epsilon, \theta, \sigma_1 \in (0,1)$, and $\sigma_2 > 1$ s.t. $\frac{1-\theta}{1-\theta\sigma_1} + \frac{1}{\sigma_2} < 1$

Output: a generalization $\mathbf{x} \in \mathbf{a}^+$

1. define $\phi_l(k) = \min_{\mathbf{x}\in\mathbf{a}^+:\Phi^{\mathbf{a}}(\mathbf{x})>0} \Phi^{\mathbf{a}}(\mathbf{x})$ and $\phi_u(k) = \max_{\mathbf{x}\in\mathbf{a}^+} \Phi^{\mathbf{a}}(\mathbf{x})$; for $i = 0,1,2,\ldots,U = \lceil \log_{(1+\epsilon)} \frac{\phi_u(k)}{\phi_l(k)} \rceil$, define $\tau_i = \phi_l(k)(1+\epsilon)^i$; define $T(\cdot) = |\rho(\cdot,\Theta)|$
2. **for** $i \in \{1,\ldots,U\}$ **do**
3. solve the maximization problem (over a convex set)

$$z_i^* = \max\ \hat{u}^{\mathbf{a}}(\mathbf{x}) \text{ subject to } \hat{T}(\mathbf{x}) \geq \frac{\tau_{i-1}}{c},\ \hat{\Phi}^{\mathbf{a}}(\mathbf{x}) \leq \tau_i$$

4. let $\hat{\mathbf{x}}^i$ be an optimal solution to the problem in step 3
5. **repeat**
6. $X^i = RR(\hat{\mathbf{x}}^i)$ and let $\mathbf{x}^i = \vee_{\mathbf{x}\in X^i}\mathbf{x}$ be the corresponding element in $\mathbf{a}^+$
7. **until** $T(\mathbf{x}^i) \geq \sigma_1 \frac{\tau_{i-1}}{c}$ and $\Phi^{\mathbf{a}}(\mathbf{x}^i) \leq \sigma_2\tau_i$
8. **return** $\mathbf{x} \in \text{argmax}_i\, u^{\mathbf{a}}(\mathbf{x}^i)$

III. Satisfying Differential Privacy

Differential privacy [8], [9] provides a mathematical way to model and bound the information gain when an individual is added (removed) to (from) a data set $\mathcal{D} \subseteq \mathcal{L}$.

It is natural that privacy degrades when multiple operations are performed on the same set since more information is exposed. However differential privacy has the advantage that privacy degrades in a well controlled manner.

Definition 4: Differential Privacy [10]

A randomized algorithm $\mathcal{A} : \mathcal{D} \to 2^{\mathcal{L}}$ is said to satisfy the (ϵ, δ)-differential privacy if

$$\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{G}] \leq e^{\epsilon} \Pr[\mathcal{A}(\mathcal{D}') \in \mathcal{G}] + \delta, \tag{3}$$

for any two data sets $\mathcal{D}$ and $\mathcal{D}'$ that differ by at most one record, and any subset of outputs $\mathcal{G} \subseteq \mathrm{Range}(\mathcal{A})$.

A. Challenges

For every record $\mathbf{a}$ in the database $\mathcal{D}$, we define an "aggregate utility" function $f^{\mathbf{a}}$ as in (2). Our ultimate goal is to design a (randomized) mechanism $\mathcal{A} : \mathcal{D} \to 2^{\mathcal{L}}$ that outputs a set $\mathcal{G} \subseteq \mathcal{L}$ that satisfies the following 3 conditions:

(C1) *Complete cover*: for each $\mathbf{a} \in \mathcal{D}$, there is a $\mathbf{g}^{\mathbf{a}} \in \mathcal{A}(\mathcal{D})$ such that $\mathbf{g}^{\mathbf{a}}$ generalizes $\mathbf{a}$, that is, $\mathbf{g}^{\mathbf{a}} \succeq \mathbf{a}$ (with probability 1);

(C2) *Differential privacy*: $\mathcal{A}(\mathcal{D})$ satisfies the (ϵ, δ)-differential privacy, for some given constants ϵ and δ;

(C3) *Utility maximization*: the expected average utility

$$\mathcal{M}(f, \mathcal{D}) = \mathbb{E}\left[\frac{1}{|\mathcal{D}|} \sum_{\mathbf{a} \in \mathcal{D}} f^{\mathbf{a}}(\mathbf{g}^{\mathbf{a}})\right] \tag{4}$$

is maximized.

We may also consider the threshold version wherein the function $f^{\mathbf{a}}$ above is replaced by $r^{\mathbf{a}}$, and the generalizations $\mathbf{g}^{\mathbf{a}}$ satisfy $u^{\mathbf{a}}(\mathbf{g}^{\mathbf{a}}) \geq c$. In this case, the conditions (C1) and (C3) are replaced by:

(C1′) *Complete cover*: for each $\mathbf{a} \in \mathcal{D}$, there is a $\mathbf{g}^{\mathbf{a}} \in \mathcal{A}(\mathcal{D})$ such that $\mathbf{g}^{\mathbf{a}} \succeq \mathbf{a}$ and $u^{\mathbf{a}}(\mathbf{g}^{\mathbf{a}}) \geq c$ (with probability 1);

(C3′) *Risk minimization*: the expected average risk

$$\mathcal{M}(r,\mathcal{D}) = \mathbb{E}\left[\frac{1}{|\mathcal{D}|}\sum_{\mathbf{a}\in\mathcal{D}} r^{\mathbf{a}}(\mathbf{g}^{\mathbf{a}})\right]$$

is minimized. For lack of space we focus here mostly on the first variant and give the details of the threshold version in [12].

Some further notation: We define h to be the maximum possible height of the k VGH's. As before, we assume that $\phi_l(k) \leq \Phi^{\mathbf{a}}(\mathbf{x}) \leq \phi_u(k)$ and $u(\mathbf{x}) \leq \nu(k)$ for all $\mathbf{a} \in \mathcal{D}$ and all $\mathbf{x} \in \mathbf{a}^+$, and some functions $\phi_l(k)$, $\phi_u(k)$ and $\nu(k)$ that depend only on the dimension k. We assume also that the database is large enough: $|\mathcal{D}| \geq \nu(k)^\kappa$, where κ is the constant used in (2). For $\mathcal{L}' \subseteq \mathcal{L}$, we denote by $\textsc{Optimum}(\mathcal{D},\mathcal{L}')$ the maximum average utility when each generalization $\mathbf{g}$ is chosen from the sublattice $\mathcal{L}'$. We define $f_{max} = \max_{\mathbf{a}\in\mathcal{D},\ \mathbf{x}\in\mathbf{a}^+} f^{\mathbf{a}}(\mathbf{x})$, and $r_{max} = \max_{\mathbf{a}\in\mathcal{D},\ \mathbf{x}\in\mathbf{a}^+} r^{\mathbf{a}}(\mathbf{x})$. By our assumptions, $f_{max} \leq \frac{\lambda|\mathcal{D}|}{\phi_l(k)} + \nu(k)^\kappa$, $r_{max} \leq \phi_u(k)$, and hence, $\frac{f_{max}}{|\mathcal{D}|} \leq t_f(k)$ and $r_{max} \leq t_r(k)$ are bounded constants that depend on the dimension, but not on the size of the database. Furthermore, let $\mathcal{D}_{-\mathbf{a}}$ denote the dataset $\mathcal{D}$ after removing the record $\mathbf{a}$.

B. t-Frequent Elements

Ideally, one would like to generalize the database records with two goals in mind: (1) satisfy differential privacy, and (2) maximize the average utility obtained from the generalization. Unfortunately, the following example shows that it is not possible in general to achieve the two objectives (C2) and (C3) at the same time.

Example 2: Consider a database $\mathcal{D}$ whose attributes are generalized through k VGH's. The i^{th} VGH is of the form: $\mathcal{L}_i = \{\perp, a_i, b_i^1, b_i^2, \ldots, b_i^h\}$ with only the relations $\perp \succeq_i a_i$ and $\perp \succeq_i b_i^1 \succeq_i b_i^2 \succeq_i \cdot \succeq_i b_i^h$. Suppose that there is only one record $\mathbf{a}_0$ in $\mathcal{D}$ whose attributes are $a_1, \ldots, a_k$, while all other records have the ith attribute belonging to the chain $\{b_i^1, b_i^2, \ldots, b_i^h\}$ for all i.

Let $\mathcal{G} = \{\gamma^{\mathbf{a}} : \mathbf{a} \in \mathcal{D}\}$ be a set of generalizations such that $\gamma^{\mathbf{a}_0} \in \{(a_i, \mathbf{x}_{-i}) : \mathbf{x}_{-i} \in \prod_{j \neq i} \mathcal{L}_j\}$. Then, for any mechanism $\mathcal{A}$, $\Pr[\mathcal{A}(\mathcal{D}_{-\mathbf{a}_0}) \in \mathcal{G}] = 0$ since, for all i, none of the records in $\mathcal{D}_{-\mathbf{a}_0}$ have attribute a_i. Thus, in order to satisfy (3), we must have $\Pr[\mathcal{A}(\mathcal{D}) \in \mathcal{G}] \leq \delta$, implying that the "trivial" generalization $\gamma^{\mathbf{a}_0} = \perp$ must be chosen for $\mathbf{a}_0$ with probability at least $1 - \delta$. In particular, if the utility of $\mathbf{a}_0$ is very large compared to the maximum average utilities of all other records, then only a fraction δ of this utility can be achieved by any differentially private mechanism.

Examining the above example, we observe that the main obstacle for obtaining differential privacy is that some of the elements in $\mathcal{L}$ (such as $\mathbf{a}_0$ in the example) are not generalizing "enough" number of records. This motivates us to consider only those elements in $\mathcal{L}$ which are generalizing many records in $\mathcal{D}$. More formally, following [2], [11], we say that an element $\mathbf{x} \in \mathcal{L}$ is *t-frequent* for a given integer t with respect to the given database $\mathcal{D}$, if it generalizes at least t records in $\mathcal{D}$: $|\rho(\mathbf{x}, \mathcal{D})| \geq t$. In the sequel, we denote by $\mathcal{L}_t(\mathcal{D})$ the set of t-frequent elements in $\mathcal{L}$ with respect $\mathcal{D}$.

C. The Mechanism

We will apply the framework of McSherry and Talwar [28]. For $\mathbf{a} \in \mathcal{D}$ and $\mathbf{x} \in \mathbf{a}^+$, define

$$q^{\epsilon'}_{f^{\mathbf{a}}}(\mathbf{x}) = \frac{e^{\epsilon' f^{\mathbf{a}}(\mathbf{x})/|\mathcal{D}|}}{\sum_{\mathbf{x}' \in \mathbf{a}^+} e^{\epsilon' f^{\mathbf{a}}(\mathbf{x}')/|\mathcal{D}|}}, \text{ or } q^{\epsilon'}_{r^{\mathbf{a}}}(\mathbf{x}) = \frac{e^{-\epsilon' r^{\mathbf{a}}(\mathbf{x})/|\mathcal{D}|}}{\sum_{\mathbf{x}' \in \mathbf{a}^+} e^{-\epsilon' r^{\mathbf{a}}(\mathbf{x}')/|\mathcal{D}|}}, \tag{5}$$

for some constant $\epsilon' \in (0, 1)$. This distribution has the property that it tends to give preference to elements with larger utility (hence, approximately maximizing the utility), but in such a smooth way that the output of the mechanism does not change much if the size of the database changes by a constant (hence, satisfying differential privacy). Note that, since we assume below that the external database $\Theta = \mathcal{D}$, $f^{\mathbf{a}}(\cdot)$ and $r^{\mathbf{a}}(\cdot)$ are functions of $\mathcal{D}$, therefore we sometimes refer to them as $f^{\mathbf{a},\mathcal{D}}(\cdot)$ and $r^{\mathbf{a},\mathcal{D}}(\cdot)$.

We introduce a parameter β s.t. $\beta \geq e^{-\epsilon}$. We define $\eta(k) = 2\left(\frac{\lambda}{\phi(k)} + 1\right)$ and choose $t' = \frac{\beta^2 h^k e^{\epsilon' t_f(k)}}{1-\beta}$. In case of risk minimization (conditions (C1′) and (C3′)), we define $\eta(k) = \phi_u(k)$.

Algorithm 2 $\mathcal{A}(\mathcal{D}, \beta, \epsilon, \theta_0)$

Input: a database $\mathcal{D} \subseteq \mathcal{L}$, a number $\beta \in (0,1)$, an accuracy ϵ, and a constant $\theta_0 \in (0,1)$

Output: a subset $\mathcal{G} \subseteq \mathcal{L}$ satisfying (C1)

1. let $\epsilon' = \frac{\epsilon + \ln \beta}{3\eta(k)(1-\beta)}$
2. let $t = \theta|D|$ where θ is chosen randomly in $(\theta_0, 1)$
3. find the sublattice $\mathcal{L}_t(\mathcal{D}) \subseteq \mathcal{L}$ of t-frequent elements
4. sample a set $\mathcal{I}_s \subseteq \mathcal{D}$ such that $\Pr[\mathbf{a} \in \mathcal{I}_s] = 1 - \beta$ for all $\mathbf{a} \in \mathcal{D}$ (independently)
5. **for all** $\mathbf{a} \in \mathcal{I}_s$ **do**
6. sample $\mathbf{x} \in \mathbf{a}^+ \cap \mathcal{L}_t(\mathcal{D})$ with prob. $q^{\epsilon'}_{f^{\mathbf{a}}}(\mathbf{x})$; (or sample $\mathbf{x} \in \mathbf{a}^+ \cap \{\mathbf{g} \in \mathcal{L}_{t_{\mathbf{a}}}(\mathcal{D}) : u(\mathbf{g}) \geq c\}$ with prob. $q^{\epsilon'}_{r^{\mathbf{a}}}(\mathbf{x})$ in case of the threshold version)
7. set $g^{\mathbf{a}} = \mathbf{x}$
8. **return** the (multiset) $\{\perp\} \cup \{g^{\mathbf{a}} : \mathbf{a} \in \mathcal{I}_s\}$

Algorithm 2 shows the mechanism which initially samples each record with probability $1 - \beta$ (step 4). Then for each sampled record $\mathbf{a} \in \mathcal{D}$, it outputs an element from the generalization $\mathbf{a}^+ \cap \mathcal{L}_t(\mathcal{D})$ according to the exponential distribution (5) defined by the utility. Note that the sampling step 4 is necessary, or otherwise the outputs on two databases with different sizes will be different with probability 1. Note also that the threshold frequency t is chosen at random; otherwise, differential privacy cannot be guaranteed to hold since an element can be frequent in in $\mathcal{D}$ but not in $\mathcal{D}_{-\mathbf{a}}$. Clearly, the output of the algorithm satisfies (C1). (or (C1′) for the threshold version). We show that it satisfies approximately (C2) and (in some cases) (C3) (or (C3′) for the threshold version). In the next section, we show how the sampling step 6 can be performed in

polynomial time, when the dimension is not fixed (i.e., it is part of the input).

Theorem 3:

(i) $\mathcal{A}(\mathcal{D})$ satisfies $(\epsilon, \delta + o(1))$-differential privacy[4];

(ii) $\mathcal{A}(\mathcal{D})$ satisfies (C3) (respectively, (C3′)) approximately: The expected average utility obtained is at least $(1-\beta)(1-\frac{3}{\ell})\textsc{Optimum}(\mathcal{D}, \mathcal{L}_t(\mathcal{D}))$ whenever the optimum average utility satisfies $\textsc{Optimum}(\mathcal{D}, \mathcal{L}_t(\mathcal{D})) \geq \frac{\ell k |\mathcal{D}|}{\epsilon'} \ln(h\ell)$ for some constant ℓ.

D. Sampling

In this section we consider the problem of sampling from an exponential distribution defined by (5). We start with a few preliminaries.

Sampling from a log-concave distribution over a convex body: Let $\mathcal{B}$ be a convex set, and $q : \mathcal{B} \to \mathbb{R}_+$ be a *log-concave* density function, that is, $\log q$ is concave over $\mathcal{B}$. For instance, the density function $q^{\epsilon'}_{f^{\mathbf{a}}}(\mathbf{x})$ defined in (5) is log-concave. It known [3], [26], [14] that we can sample from $\mathcal{B}$ according to such a distribution q approximately in polynomial time. More precisely, there is a polynomial-time algorithm that samples a point $\mathbf{x} \in \mathcal{B}$ with density $\tilde{q} : \mathcal{B} \to \mathbb{R}_+$, such that

$$\sup_{\mathcal{B}' \subseteq \mathcal{B}} \left| \frac{\tilde{q}(\mathcal{B}')}{\tilde{q}(\mathcal{B})} - \frac{q(\mathcal{B}')}{q(\mathcal{B})} \right| \leq \delta', \tag{6}$$

where $\tilde{q}(\mathcal{B}') = \int_{\mathbf{x} \in \mathcal{B}'} \tilde{q}(\mathbf{x}) d\mathbf{x}$, and δ' is a given desired accuracy ($q(\mathcal{B}')$ is defined similarly). We will ignore the issue of representation of $\mathcal{B}$ and q, since for our purposes, both are given explicitly. We only require that q has a polynomial bit-length representation, that is, $\log(\max_{\mathbf{x} \in \mathcal{B}} f^{\mathbf{a}}(\mathbf{x}) / \min_{\mathbf{x} \in \mathcal{B}} f^{\mathbf{a}}(\mathbf{x}))$ is bounded by a polynomial in the input size. Note that the running time of the sampling algorithm depends polynomially on $\log \frac{1}{\delta'}$, so δ' can be set exponentially small in $|\mathcal{D}|$.

[4]Here, $o(1)$ hides a factor of the form $\frac{h^k}{|\mathcal{D}|^2}$, which is negligible for a sufficiently large database, assuming that $h^k = o(|\mathcal{D}|)$. In particular, our bounds on differential privacy apply when $k = o(\log_h |\mathcal{D}|)$.

Recall that for Theorem 3 to hold, it is enough to be able to sample $\mathbf{x} \in \mathbf{a}^+$ with probability proportional to $e^{\epsilon' f^{\mathbf{a}}(\mathbf{x})/|\mathcal{D}|}$ for each record $\mathbf{a} \in \mathcal{D}$. If the dimension (number of attributes in $\mathcal{D}$) is sufficiently small, then the sampling is trivial. Therefore, we assume in this section that the dimension k is part of the input. Due to the nature of the sampling procedure described below, we will have to extend the function $f^{\mathbf{a}}(\mathbf{x})$ over the hypercube (recall the Lovász extension and the randomized rounding procedure RR in Section II-C), and then sample from the exponential distribution over the hypercube. Once we get a point sampled from the hypercube, we apply randomized rounding to get back a point in $\mathbf{a}^+$. While the resulting distribution over $\mathbf{a}^+$ might not be exponential[5], we will prove that it is still sufficient for proving differential privacy.

Let us consider a single function $f^{\mathbf{a}} : \mathcal{C}^{\mathbf{a}} \to \mathbb{R}_+$, and assume that $f^{\mathbf{a}}(\mathbf{x}) = \frac{\xi^{\mathbf{a}}(\mathbf{x})}{\Phi^{\mathbf{a}}(\mathbf{x})}$, where $\xi^{\mathbf{a}} : \mathcal{C}^{\mathbf{a}} \to \mathbb{R}_+$ is supermodular, $\Phi^{\mathbf{a}} : \mathcal{C}^{\mathbf{a}} \to \mathbb{R}_+$ is modular, and $\mathcal{C}^{\mathbf{a}}$ is the chain product $\mathbf{a}^+$ (note that the function $f^{\mathbf{a}}(\cdot)$ defined in (2) satisfies these conditions). The function $f^{\mathbf{a}}$ is not necessarily supermodular, and hence its extension is not generally concave. To deal with this issue, we will divide the lattice into layers according to the value of $\Phi^{\mathbf{a}}$, and sample from each layer independently (this slicing is somewhat similar to what we did in the proof of Theorem 2 in Section II-C). More precisely, let $\epsilon'' \in (0,1)$ be a constant. For $i = 0, 1, 2, \ldots, U = \log_{1+\epsilon''}\left(\frac{\phi_u(k)}{\phi_l(k)}\right)$, define the layer $\mathcal{C}^{\mathbf{a},i}(\epsilon'') = \{\mathbf{x} \in \mathcal{C}^{\mathbf{a}} : \ (1+\epsilon'')^i \leq \Phi^{\mathbf{a}}(\mathbf{x}) \leq (1+\epsilon'')^{i+1}\}$. Let $\mathcal{J}^{\mathbf{a}}$ and $\mathcal{F}^{\mathbf{a}}$ be the set of join-irreducible elements of $\mathcal{C}^{\mathbf{a}}$ and the corresponding ring family defined in Section II-B.3. For $X \subseteq \mathcal{J}^{\mathbf{a}}$, define

$$\begin{aligned} \Psi^{\mathbf{a},i}(X) &= \frac{\epsilon' \xi^{\mathbf{a}}(\vee_{\mathbf{x}\in X}\mathbf{x})}{|\mathcal{D}|(1+\epsilon'')^i}, \ \Phi_1^{\mathbf{a}}(X) = \Phi^{\mathbf{a}}(\vee_{\mathbf{x}\in X}\mathbf{x}), \\ T(X) &= |\{S(\mathbf{a}) : \ \mathbf{a} \in \mathcal{D} \text{ and } S(\mathbf{a}) \supseteq X\}|, \end{aligned}$$

where $S(\cdot)$ is the operator defined in Section II-B.3, and ϵ' is the parameter defined

[5]At least we are not able to prove it.

in Algorithm 2. Since $\Psi^{\mathbf{a},i}$ and T are supermodular, their Lovász extensions $\hat{\Psi}^{\mathbf{a},i}, \hat{T} : P(\mathcal{F}^{\mathbf{a}}) \to \mathbb{R}$ are concave. Likewise, $\Phi_1^{\mathbf{a}}$ is modular and hence its Lovász extension $\hat{\Phi}_1^{\mathbf{a}} : P(\mathcal{F}^{\mathbf{a}}) \to \mathbb{R}$ is linear. It follows that the set

$$\mathcal{B}^{\mathbf{a},i}(\epsilon'') = \{\mathbf{x} \in P(\mathcal{F}^{\mathbf{a}}) : \ \hat{T}(\mathbf{x}) \geq t, \ (1+\epsilon'')^i \leq \hat{\Phi}_1^{\mathbf{a}}(\mathbf{x}) \leq (1+\epsilon'')^{i+1}\}$$

is convex. Note that the constraint $\hat{T}(\mathbf{x}) \geq t$ is added to enforce sampling only from t-frequent elements (which will be only achieved approximately).

The details of the sampling procedure are shown in Algorithm 3. It works by first picking a layer at random from $0, 1, \ldots, U$. Then a point $\hat{\mathbf{x}}$ is picked from (the continuous extension of) this layer according to the log-concave density

$$q(\mathbf{x}) = \frac{e^{\hat{\Psi}^{\mathbf{a},i}(\mathbf{x})}}{\int_{\mathbf{x}' \in P(\mathcal{F}^{\mathbf{a}})} e^{\hat{\Psi}^{\mathbf{a},i}(\mathbf{x}')}}. \tag{7}$$

We then round $\hat{\mathbf{x}}$ by procedure RR to a set X in the family $\mathcal{F}^{\mathbf{a}}$, which corresponds to a point $\vee_{\mathbf{x} \in X} \mathbf{x}$ in the lattice $\mathcal{C}^{\mathbf{a}}$. If X is not approximately t-frequent, we apply RR again to $\hat{\mathbf{x}}$. If t is large enough, we can argue that X is σt-frequent with constant probability for some constant σ.

Examining the proof of Theorem 3, we notice that the properties of the exponential distribution for satisfying differential privacy are not used much. In fact, ignoring small constant factors in the exponents, it is enough to show the following.

Lemma 1: With some $\delta' = O(\delta 2^{-|\mathcal{D}|^2})$,

$$e^{-2\epsilon'(1+\epsilon'')\frac{\eta(k)}{|\mathcal{D}|}} \leq \frac{\Pr[\mathbf{g}^{\mathbf{a}}(\mathcal{D}) = \gamma^{\mathbf{a}}] - \delta'}{\Pr[\mathbf{g}^{\mathbf{a}}(\mathcal{D}_{-\mathbf{a}_0}) = \gamma^{\mathbf{a}}] + \delta'} \leq e^{2\epsilon'(1+\epsilon'')\frac{\eta(k)}{|\mathcal{D}|}},$$

for every $\mathbf{a}, \mathbf{a}_0 \in \mathcal{D}$ and any output $\gamma^{\mathbf{a}} \in \mathbf{a}^+$, when $\mathbf{g}^{\mathbf{a}}(\mathcal{D})$ is sampled according to Algorithm Sample-Point$(\mathbf{a}, \epsilon', \epsilon'', \theta, \sigma)$.

Running time: To show that the expected running time is polynomial, it is enough to bound the probability of the event that $T(X) < \sigma t$ in step 6. Let $\hat{\mathbf{x}}$ be the point sampled in step 3 and $X = RR(\hat{\mathbf{x}})$. Then, $\mathbb{E}[T(X)] = \hat{T}(\hat{\mathbf{x}}) \geq t$ since $\hat{\mathbf{x}} \in \mathcal{B}^{\mathbf{a},i}(\epsilon'')$. By Markov's

Algorithm 3 Sample-Point$(\mathbf{a}, \epsilon', \epsilon'', \theta, \sigma)$

Input: a record $\mathbf{a} \in \mathcal{D}$, and real numbers $\epsilon', \epsilon'', \theta, \sigma \in (0, 1)$

Output: a point $\mathbf{x} \in \mathcal{C}^{\mathbf{a}}$

1. let $t = \theta|\mathcal{D}|$
2. pick $i \in \{0, 1, \ldots, U\}$ at random
3. sample $\hat{\mathbf{x}} \in \mathcal{B}^{\mathbf{a},i}(\epsilon'')$ with density $\tilde{q}$ satisfying (6), where $q(\mathbf{x})$ is given by (7) for $\mathbf{x} \in [0, 1]^{\mathcal{J}^{\mathbf{a}}}$
4. **repeat**
5. $\quad X = RR(\hat{\mathbf{x}})$
6. **until** $T(X) \geq \sigma t$
7. **return** $\vee_{\mathbf{x} \in X} \mathbf{x}$

Inequality[6] , $\Pr[T(X) < \sigma t] \leq \frac{1-\theta}{1-\theta\sigma}$. Thus, the expected number of calls to $RR(\hat{\mathbf{x}})$ until we get $T(X) \geq \sigma t$ is at most $\frac{1-\theta\sigma}{1-\theta}$.

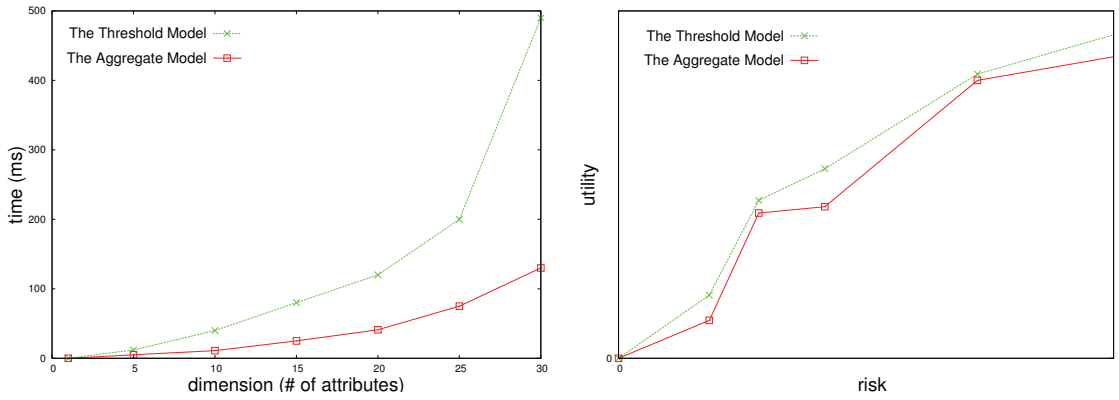
IV. Experimental Analysis

We conducted a number of experiments to evaluate the proposed algorithms. In the next subsection, we explain our experimental setup. The results for the aggregate utility function, and the sampling algorithm are given in Sections IV-B, and IV-C, respectively.

A. Experimental Setup

We use an experimental setup similar to that described in [13]. Specifically, we conducted our experiments on the `item description` table of Wal-Mart database. The table contains more than 400,000 records each with 30 attributes. The risk components

[6]Let Y be a random variable taking non-negative values. Then, Markov's inequality states that for any $y > 0$, $\Pr[Y \geq y] \leq \frac{\mathbb{E}[Y]}{y}$. In particular, if Y' is a random variable taking values bounded by M, then $\Pr[Y < y] \leq \frac{M - \mathbb{E}[Y]}{M - y}$.



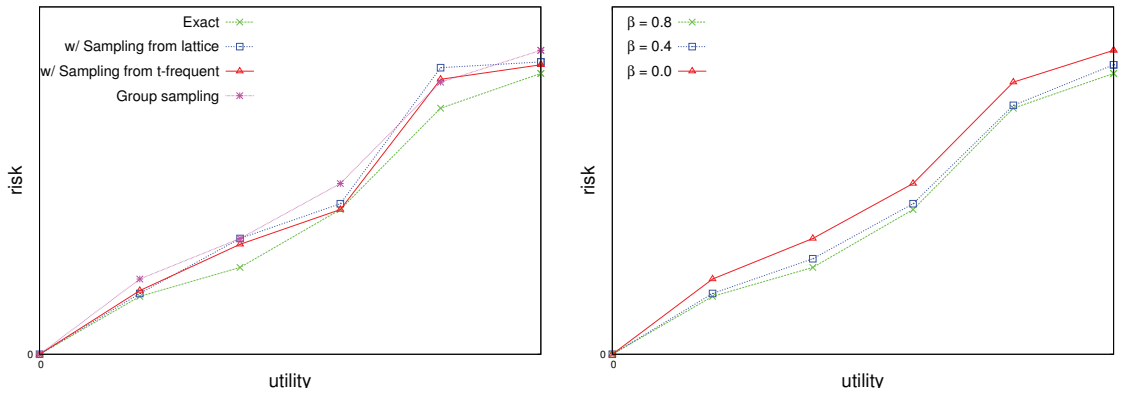
(a) (b)

Fig. 3. The impact of imposing supermodularity on the optimization objective function (a) Efficiency, and (b) Accuracy.

are computed based on both identifiability and sensitivity as described in [22]. We use a modified harmonic mean to compute the sensitivity of a parent node w_p with l immediate children given the sensitivities of these children w_i: $w_p = \frac{1}{\sum_{1 \leq i \leq l} \frac{1}{w_i}}$ with the exception that the root node (corresponding to suppressed data) has a sensitivity weight of 0. Moreover, we use a simplified utility function $u(\mathbf{a})$ to capture the information benefit of releasing a record $\mathbf{a}$: $u(\mathbf{a}) = \sum_{i=1}^{k} depth(a_i)$ where $depth(a_i)$ represents the distance between the attribute value a_i and the greatest value $\perp$.

B. The Modified Aggregate Algorithm

We compare the performance of both the threshold optimization algorithm and the modified (with supermodular objective function) aggregate algorithm. We implement the supermodular minimization using [21]. We run both algorithms with various (1) number of attributes, and (2) risk thresholds. Fig. 3 depicts the impact of imposing supermodularity on the optimization objective function. It is clear that while both algorithms have comparable utilities, the modified aggregate algorithm significantly outperforms



(a) (b)

Fig. 4. The impact of sampling from (a) all items, and (b) subset of the items.

the threshold algorithm in terms of running time.

C. *The Sampling Algorithm* $\mathcal{A}(\mathcal{D}, \beta, \epsilon, t)$

In this section, we study the impact of sampling on the utility of the generated database. We evaluate both the risk and utility functions using the models presented in Section IV-A and use equation (2) to compute the overall utility of a database $\mathcal{D}$. Each data point is obtained by running Algorithm 2 several times with random sampling and taking the average of the obtained results. We consider exponential sampling over both the whole lattice as well as the t-frequent items. We apply exponential sampling on the whole lattice with the value of β set to 1. Fig. 4(a) compares the results obtained from sampling with the optimal results. Indeed, the results are consistent with our claim that differential privacy occurs at the cost of sacrificing (a little) data utility. The figure also shows that our results are superior in terms of risk and utility to those obtained from the group sampling proposed in [29]. The same conclusion is reached when decreasing the value of β, and therefore increasing the sampling set size. Fig. 4(b) depicts the impact of varying the sampling set size on utility.

V. Related Work

In [13], an algorithm (ARUBA) to address the tradeoff between data utility and data privacy is proposed. The proposed algorithm determines a personalized optimum data transformations based on predefined risk and utility models. However, ARUBA provides no scalability guarantees and lacks the necessary theoretical foundations for privacy risk.

A top-down specialization algorithm is developed by Fung et al. [15] that iteratively specializes the data by taking into account both data utility and privacy constraints. A genetic algorithm solution for the same problem is proposed by Iyengar [20]. Both approaches consider classification quality as a metric for data utility. However, to preserve classification quality, they measure privacy as how uniquely an individual can be identified by collapsing every subset of records into one record. The per-record customization nature of our algorithms makes them superior over other algorithms.

A personalized generalization technique is proposed by Xiao and Tao [33]. Under such approach users define maximum allowable specialization levels for their different attributes. That is, sensitivity of different attribute values are binary (either released or not released). In contrast, our proposed scheme provides users with the ability to specify sensitivity weights for their attribute values.

Perhaps the most related work is the differentially private data release proposed in [29]. In that paper, the authors also consider a product of taxonomies for data generalization, assume some utility function quantifying the information content of the released generalizations, then apply the exponential mechanism to obtain a differentially private mechanism. Their application of the exponential mechanism is done in a somewhat restrictive way in the sense that they do not sample from the space of all generalizations as we do. Rather, the sampling is performed in a heuristic way as follows. All the records are put in one group and generalized by the top element $(\perp, \ldots, \perp)$. Then one of the top elements in the different taxonomies is chosen according to an exponential distribution

defined in terms of some utility function. The chosen element is replaced by its children in the corresponding taxonomy. This splits the current group into a number of subgroups, each generalized by an element in the product of the taxonomies. The process is repeated in each of the subgroups. After a predefined number of splits, the count of the number of elements in each of the obtained groups is perturbed by a Laplacian noise. One main restriction of this approach is that the utility function has to be *record-independent*. On the contrary, in our formulation we allow the utility function to be different for each record in the database.

VI. Conclusion and Future Directions

In this paper we addressed both scalability and privacy risk when identifying the optimal set of transformations which, when carried out on a given table, generate a resulting table that satisfies a set of optimality constraints. Since the problem is NP-hard, we suggested several methods to deal this hardness by utilizing the supermodularity properties of the risk function. In particular, we gave an approximation algorithm that computes a nearly optimal solution when the risk threshold is low enough. We also proposed a scalable algorithm that meets differential privacy (with acceptable probability) by applying a specific random sampling.

There are several open problems that deserve investigation in relation to our work. Can the approximation algorithm be extended to the cases when the risk threshold is small? Examining the NP-hardness reduction [12], one observes the connection to the *notoriously hard* densest subgraph problem. While this might shed some light on the difficulty of obtaining an optimal solution for the threshold model, it may be also possible to extend some of the techniques used for the densest subgraph problem to our problem.

One also notes the limitation of the exponential mechanism with respect to the theoretically proved bound on the expected utility (Theorem 3-(ii)). A very interesting point would be to modify the mechanism such that better utility bounds can be obtained.

REFERENCES

[1] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Procededings of the International Conference on Very Large Data Bases (VLDB)*, pages 901–909, 2005.

[2] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM Special Interest Group on Management of Data (SIGMOD)*, pages 207–216, 1993.

[3] D. Applegate and R. Kannan. Sampling and integration of near log-concave functions. In *Proceedings of the Annual ACM Symposium on Theory of Computing (STOC)*, pages 156–163, 1991.

[4] R. J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pages 217–228, 2005.

[5] J. Cao, B. Carminati, E. Ferrari, and K.-L. Tan. CASTLE: Continuously anonymizing data streams. *IEEE Transactions on Dependable and Secure Computing*, 8(3):337–352, 2011.

[6] J. Cao, P. Karras, P. Kalnis, and K.-L. Tan. SABRE: A sensitive attribute bucketization and redistribution framework for *t*-closeness. *Journal on Very Large Data Bases (VLDB)*, 20(1):59–81, 2011.

[7] C. Dwork. Differential privacy. In *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, pages 1–12, 2006.

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

[9] C. Dwork. Differential privacy: A survey of results. In *Proceedings of the International Conference on Theory and Applications of Models of Computation (TAMC)*, pages 1–19, 2008.

[10] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *Proceedings of the 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2006.

[11] K. M. Elbassioni. Algorithms for dualization over products of partially ordered sets. *SIAM Journal on Discrete Mathematics*, 23(1):487–510, 2009.

[12] M. R. Fouad, K. Elbassioni, and E. Bertino. Towards a differentially private data anonymization. Technical Report CERIAS 2012-1, Purdue University, 2012.

[13] M. R. Fouad, G. Lebanon, and E. Bertino. ARUBA: A risk-utility-based algorithm for data disclosure. In *Proceedings of the VLDB Workshop on Secure Data Management (SDM)*, pages 32–49, 2008.

[14] A. Frieze, R. Kannan, and N. Polson. Sampling from log-concave distributions. *Annals of Applied Probability*, 4:812–837, 1994.

[15] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pages 205–216, 2005.

[16] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis. Fast data anonymization with low information loss. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, pages 758–769, 2007.

[17] G. A. Grätzer. *General Lattice Theory*. Birkhauser, second edition, 2003.

[18] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, second corrected edition, 1993.

[19] A. Gupta, K. Ligett, F. McSherry, A. Roth, and K. Talwar. Differentially private combinatorial optimization. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1106–1125, 2010.

[20] V. S. Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 279–288, 2002.

[21] A. Krause. *The UCI Repository of Machine Learning Databases*. `http://users.cms.caltech.edu/home/~krausea/sfo/`.

[22] G. Lebanon, M. Scannapieco, M. R. Fouad, and E. Bertino. Beyond k-anonymity: A decision theoretic framework for assessing privacy risk. *Privacy in Statistical Databases, Springer Lecture Notes in Computer Science*, 4302:217–232, 2006.

[23] N. Li, W. H. Qardaji, and D. Su. Provably private data anonymization: Or, k-anonymity meets differential privacy. *Computing Research Repository (CoRR)*, abs/1101.2604, 2011.

[24] T. Li and N. Li. t-closeness: Privacy beyond k-anonymity and l-diversity. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, pages 106–115, 2007.

[25] L. Liu, M. Kantarcioglu, and B. Thuraisingham. The applicability of the perturbation based privacy preserving data mining for real-world data. *Data and Knowledge Engineering (DKE)*, 65(1):5–21, 2008.

[26] L. Lovász and S. Vempala. Fast algorithms for log-concave functions: Sampling, rounding, integration and optimization. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 57–68, 2006.

[27] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, page 24, 2006.

[28] F. McSherry and K. Tawler. Mechanism design via differential privacy. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pages 156–163, 2007.

[29] N. Mohammed, R. Chen, B. C. Fung, and P. S. Yu. Differentially private data release for data mining. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 493–501. ACM, 2011.

[30] K. Murota. *Discrete Convex Analysis*. The Society for Industrial and Applied Mathematics (SIAM), 2003.

[31] P. Samarati and L. Sweeney. Data to provide anonymity when disclosing information. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, page 188, 1998.

[32] L. Sweeney. Privacy-enhanced linking. *Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) Explorations*, 7(2):72–75, 2005.

[33] X. Xiao and Y. Tao. Personalized privacy preservation. In *Proceedings of the ACM Special Interest Group on Management of Data (SIGMOD)*, pages 229–240, 2006.

Mohamed Fouad received the B.Sc. degree in Computer Science and M.Sc. degree in Mathematics from Alexandria University, Egypt, in 1992 and 1998, respectively. He also received the M.Sc. and Ph.D. degrees in Computer Science from Purdue University, USA, in 2003 and 2012, respectively, and is currently working as a Software Engineer at Google, Inc., Mountain View, USA. Mr. Fouad authored a few papers on privacy risk modeling and optimization for both data disclosure and social network information sharing. His main research interests include statistical modeling and data analysis, privacy, security, risk management, data mining, anonymity, and social networks.

Khaled Elbassioni received the B.E. and M.S. degrees in Computer Science from Alexandria University, Egypt in 1992 and 1995, respectively, and the Ph.D. degree in Computer Science from Rutgers University, USA, in 2002. He spent one year, from 2003 to 2004, as a postdoctoral fellow at Rutgers Center for Operations Research (RUTCOR), Piscataway, NJ, USA, and two years, from 2004 to 2006 as a postdoctoral fellow at Max-Planck Institute for Informatics, Saarbruecken, Germany. From 2006 to 2012, he was a senior researcher at Max-Planck Institute for Informatics, Saarbruecken. He is currently an associate professor of Computing and Information Science at Masdar Institute of Science and Technology, Abu Dhabi, UAE. His main research interests are in Theoretical Computer Science, in particular, in the complexity of enumeration problems, approximation algorithms, combinatorial optimization, and game theory.

Elisa Bertino is Professor of Computer Science at Purdue University, and serves as research director of the Center for Education and Research in Information Assurance and Security (CERIAS) and Interim Director of Cyber Center (Discovery Park). Previously, she was a faculty member and department head at the Department of Computer Science and Communication of the University of Milan. Her main research interests include security, privacy, digital identity management systems, database systems, distributed systems, and multimedia systems. She is currently serving as chair of the ACM SIGSAC and as a member of the editorial board of the following international journals: IEEE Security & Privacy, IEEE Transactions on Service Computing, ACM Transactions on Web. She also served as editor in chief of the VLDB Journal and editorial board member of ACM TISSEC and IEEE TDSC. She co-authored the book Identity Management - Concepts, Technologies, and Systems. She is a fellow of the IEEE and a fellow of the ACM. She received the 2002 IEEE Computer Society Technical Achievement Award for outstanding contributions to database systems and database security and advanced data management systems and the 2005 IEEE Computer Society Tsutomu Kanai Award for pioneering and innovative research contributions to secure distributed systems.