Differential Privacy: A Survey of Results

Cynthia Dwork

Microsoft Research
dwork@microsoft.com

Abstract. Over the past five years a new approach to privacy-preserving data analysis has born fruit [13, 18, 7, 19, 5, 37, 35, 8, 32]. This approach differs from much (but not all!) of the related literature in the statistics, databases, theory, and cryptography communities, in that a formal and *ad omnia* privacy guarantee is defined, and the data analysis techniques presented are rigorously proved to satisfy the guarantee. The key privacy guarantee that has emerged is *differential privacy*. Roughly speaking, this ensures that (almost, and quantifiably) no risk is incurred by joining a statistical database.

In this survey, we recall the definition of differential privacy and two basic techniques for achieving it. We then show some interesting applications of these techniques, presenting algorithms for three specific tasks and three general results on differentially private learning.

1 Introduction

Privacy-preserving data analysis is also known as statistical disclosure control, inference control, privacy-preserving datamining, and private data analysis. Our principal motivating scenario is a *statistical database*. A statistic is a quantity computed from a sample. Suppose a trusted and trustworthy curator gathers sensitive information from a large number of respondents (the sample), with the goal of learning (and releasing to the public) statistical facts about the underlying population. The problem is to release statistical information without compromising the privacy of the individual respondents. There are two settings: in the *noninteractive* setting the curator computes and publishes some statistics, and the data are not used further. Privacy concerns may affect the precise answers released by the curator, or even the set of statistics released. Note that since the data will never be used again the curator can destroy the data (and himself) once the statistics have been published.

In the *interactive* setting the curator sits between the users and the database. Queries posed by the users, and/or the responses to these queries, may be modified by the curator in order to protect the privacy of the respondents. The data cannot be destroyed, and the curator must remain present throughout the lifetime of the database. Of course, any interactive solution yields a non-interactive solution, provided the queries are known in advance: the curator can simulate an interaction in which these known queries are posed, and publish the resulting transcript.

There is a rich literature on this problem, principally from the satistics community (see, *e.g.*, [10, 14, 27, 28, 29, 38, 40, 26, 39] and the literature on controlled

M. Agrawal et al. (Eds.): TAMC 2008, LNCS 4978, pp. 1–19, 2008.
© Springer-Verlag Berlin Heidelberg 2008

release of tabular data, contingency tables, and cell suppression), and from such diverse branches of computer science as algorithms, database theory, and cryptography, for example as in [3, 4, 21, 22, 23, 33, 34, 41, 48], [1, 24, 25, 31], and [6, 9, 11, 12, 13, 18, 7, 19]; see also the survey [2] for a summary of the field prior to 1989.

This survey is about *differential privacy*. Roughly speaking, differential privacy ensures that the removal or addition of a single database item does not (substantially) affect the outcome of any analysis. It follows that no risk is incurred by joining the database, providing a mathematically rigorous means of coping with the fact that distributional information may be disclosive.

We will first describe three differentially private algorithms for specific, unrelated, data analysis tasks. We then present three general results about computational learning when privacy of individual data items is to be protected. This is not usually a concern in the learning theory literature, and signals the emergence of a new line of research.

2 Differential Privacy

In the sequel, the randomized function $\mathcal{K}$ is the algorithm applied by the curator when releasing information. So the input is the data set, and the output is the released information, or *transcript*. We do not need to distinguish between the interactive and non-interactive settings.

Think of a database as a set of rows. We say databases D_1 and D_2 *differ in at most one element* if one is a proper subset of the other and the larger database contains just one additional row.

Definition 1. *A randomized function $\mathcal{K}$ gives* ϵ-differential privacy *if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$,*

$$\Pr[\mathcal{K}(D_1) \in S] \leq \exp(\epsilon) \times \Pr[\mathcal{K}(D_2) \in S] \tag{1}$$

The probability is taken is over the coin tosses of $\mathcal{K}$.

A mechanism $\mathcal{K}$ satisfying this definition addresses concerns that any participant might have about the leakage of her personal information: even if the participant removed her data from the data set, no outputs (and thus consequences of outputs) would become significantly more or less likely. For example, if the database were to be consulted by an insurance provider before deciding whether or not to insure a given individual, then the presence or absence of that individual's data in the database will not significantly affect her chance of receiving coverage.

Differential privacy is therefore an *ad omnia* guarantee. It is also a very strong guarantee, since it is a statistical property about the behavior of the mechanism and therefore is independent of the computational power and auxiliary information available to the adversary/user.

Differential privacy is not an absolute guarantee of privacy. In fact, Dwork and Naor have shown that any statistical database with any non-trivial utility

compromises a natural definition of privacy [15]. However, in a society that has decided that the benefits of certain databases outweigh the costs, differential privacy ensures that only a limited amount of additional risk is incurred by participating in the socially beneficial databases.

Remark 1. 1. The parameter ϵ in Definition 1 is public. The choice of ϵ is essentially a social question and is beyond the scope of this paper. That said, we tend to think of ϵ as, say, 0.01, 0.1, or in some cases, $\ln 2$ or $\ln\ 3$. If the probability that some bad event will occur is very small, it might be tolerable to increase it by such factors as 2 or 3, while if the probability is already felt to be close to unacceptable, then an increase by a factor of $e^{0.01} \approx 1.01$ might be tolerable, while an increase of e, or even only $e^{0.1}$, would be intolerable.

2. Definition 1 discusses the behavior of the mechanism $\mathcal{K}$, and is independent of any auxiliary knowledge the adversary, or user, may have about the database. Thus, a mechanism satisfying the definition protects the privacy of an individual row in the database even if the adversary knows every other row in the database.
3. Definition 1 extends to group privacy as well (and to the case in which an individual contributes more than a single row to the database). A collection of c participants might be concerned that their collective data might leak information, even when a single participant's does not. Using this definition, we can bound the dilation of any probability by at most $\exp(\epsilon c)$, which may be tolerable for small c. Of course, the point of the statistical database is to disclose aggregate information about large groups (while simultaneously protecting individuals), so we should expect privacy bounds to disintegrate with increasing group size.

3 Achieving Differential Privacy in Statistical Databases

We will presently describe an interactive mechanism, $\mathcal{K}$, due to Dwork, McSherry, Nissim, and Smith [19], for the case of continuous-valued queries. Specifically, in this section a *query* is a function mapping databases to (vectors of) real numbers. For example, the query "Count P" counts the number of rows in the database having property P.

When the query is a function f, and the database is X, the *true answer* is the value $f(X)$. The mechanism $\mathcal{K}$ adds appropriately chosen random noise to the true answer to produce what we call the *response*. The idea of preserving privacy by responding with a noisy version of the true answer is not new, but this approach is delicate. For example, if the noise is symmetric about the origin and the same question is asked many times, the responses may be averaged, cancelling out the noise[1]. We must take such factors into account.

[1] We do not recommend having the curator record queries and their responses so that if a query is issued more than once the response can be replayed: If the query language is sufficiently rich, then semantic equivalence of two syntactically different queries is undecidable; even if the query language is not so rich, the devastating attacks demonstrated by Dinur and Nissim [13] pose completely random and unrelated queries.

Definition 2. *For $f : \mathcal{D} \to R^k$, the* sensitivity *of f is*

$$\Delta f = \max_{D_1, D_2} \|f(D_1) - f(D_2)\|_1 \tag{2}$$

for all D_1, D_2 differing in at most one element.

In particular, when $k = 1$ the sensitivity of f is the maximum difference in the values that the function f may take on a pair of databases that differ in only one element.

For many types of queries Δf will be quite small. In particular, the simple counting queries discussed above ("How many rows have property P?") have $\Delta f = 1$. Our techniques work best – introduce the least noise – when Δf is small. Note that sensitivity is a property of the function alone, and is independent of the database. The sensitivity essentially captures how great a difference (between the value of f on two databases differing in a single element) must be hidden by the additive noise generated by the curator.

The scaled symmetric exponential distribution with standard deviation $\sqrt{2}\Delta f/\epsilon$ denoted $\mathrm{Lap}(\Delta f/\epsilon)$, has mass at x proportional to $\exp(-|x|(\epsilon/\Delta f))$. More precisely, let $b = \Delta f/\epsilon$. The probability density function is $p(x) = \exp(-|x|/b)/2b$ and the cumulative distribution function is $D(x) = (1/2)(1 + \mathrm{sgn}(x)(1 - \exp(|x|/b)))$.

On query function f the privacy mechanism $\mathcal{K}$ responds with

$$f(X) + (\mathrm{Lap}(\Delta f/\epsilon))^k$$

adding noise with distribution $\mathrm{Lap}(\Delta f/\epsilon)$ independently to each of the k components of $f(X)$. Note that decreasing ϵ, a publicly known parameter, flattens out the $\mathrm{Lap}(\Delta f/\epsilon)$ curve, yielding larger expected noise magnitude. When ϵ is fixed, functions f with high sensitivity yield flatter curves, again yielding higher expected noise magnitudes.

For simplicity, consider the case $k = 1$. The proof that $\mathcal{K}$ yields ϵ-differential privacy on the single query function f is straightforward. Consider any subset $S \subseteq Range(\mathcal{K})$, and let D_1, D_2 be any pair of databases differing in at most one element. When the database is D_1, the probability mass at any $r \in Range(\mathcal{K})$ is proportional to $\exp(-|f(D_1) - r|(\Delta f/\epsilon))$, and similarly when the database is D_2. Applying the triangle inequality in the exponent we get a ratio of at most $\exp(-|f(D_1) - f(D_2)|(\Delta f/\epsilon))$. By definition of sensitivity, $|f(D_1) - f(D_2)| \leq \Delta f$, and so the ratio is bounded by $exp(-\epsilon)$, yielding ϵ-differential privacy.

It is easy to see that ϵ-differential privacy can be achieved for any (adaptively chosen) query sequence $f_1, \ldots, f_d$ by running $\mathcal{K}$ with noise distribution $\mathrm{Lap}(\sum_i \Delta f_i/\epsilon)$ on *each* query. In other words, the quality of each answer deteriorates with the sum of the sensitivities of the queries. Interestingly, it is sometimes possible to do better than this. Roughly speaking, what matters is the maximum possible value of $\Delta = ||(f_1(D_1), f_2(D_1), \ldots, f_d(D_1)) - (f_1(D_2), f_2(D_2), \ldots, f_d(D_2))||_1$. The precise formulation of the statement requires some care, due to the potentially adaptive choice of queries. For a full treatment see [19]. We state the theorem here for the non-adaptive case, viewing the (fixed) sequence of queries $f_1, f_2, \ldots, f_d$, with respective arities $k_1, \ldots, k_d$, as a single $k = \sum_{i=1}^{d} k_i$-ary query f, and recalling Definition 2 for the case of arbitrary k.

Theorem 1 ([19]). *For $f : \mathcal{D} \to R^k$, the mechanism $\mathcal{K}_f$ that adds independently generated noise with distribution* $\mathrm{Lap}(\Delta f/\epsilon)$ *to each of the k output terms enjoys ϵ-differential privacy.*

The mechanism $\mathcal{K}$ described above has excellent accuracy for insensitive queries. In particular, the noise needed to ensure differential privacy depends only on the sensitivity of the function and on the parameter ϵ. Both are independent of the database and the number of rows it contains. Thus, if the database is very large, the errors for many typical queries introduced by the differential privacy mechanism is relatively quite small.

We can think of $\mathcal{K}$ as a differential privacy-preserving interface between the analyst and the data. This suggests a general approach to privacy-preserving data analysis: find algorithms that require few, insensentitive, queries. See, *e.g.*, [7, 8, 32]. Indeed, even counting queries are extremely powerful, permitting accurate and differentially private computations of many standard datamining tasks including principal component analysis, k-means clustering, perceptron learning of separating hyperplanes, and generation of an ID3 decision tree [7], as well as (nearby) halfspace learning [8] (see Section 4.3 below).

Among the many applications of Theorem 1, of particular interest is the class of *histogram* queries. A histogram query is an arbitrary partitioning of the domain of database rows into disjoint "cells," and the true answer is the set of counts describing, for each cell, the number of database rows in this cell. Although a histogram query with k cells may be viewed as k individual counting queries, the addition or removal of a single database row can affect the entire k-tuple of counts in at most one location (the count corresponding to the cell to (from) which the row is added (deleted); moreover, the count of this cell is affected by at most 1, so by Definition 2, every histogram query has sensitivity 1. Many data analyses are simply histograms; it is thus particularly encouraging that complex histograms, rather than requiring large variance in each cell, require very little.

3.1 When Noise Makes No Sense

In some tasks, the addition of noise makes no sense. For example, the function f might map databases to strings, strategies, or trees. In a recent paper McSherry and Talwar address the problem of optimizing the output of such a function while preserving ϵ-differential privacy [35]. Assume the curator holds a database X and the goal is to produce an object y. In a nutshell, their *exponential mechanism* works as follows. There is assumed to be a *utility function u(X,y)* that measures the quality of an output y, given that the database is X. For example, if the database holds the valuations that individuals assign a digital good during an auction, $u(X, y)$ might be the revenue, with these valuations, when the price is set to y. Auctions are a good example of where noise makes no sense, since an even slightly too high price may prevent many bidders from buying.

McSherry and Talwar's *exponential mechanism* outputs y with probability proportional to $\exp(-\epsilon u(X, y)/2)$. This ensures $\epsilon\Delta u$-differential privacy, or

ϵ-differential privacy whenever $\Delta u \leq 1$. Here Δu is defined slightly differently from above; it is the maximum possible change to the value of u caused by *changing* the data of a single row (as opposed to removing or adding a row; the notions differ by at most a factor of two); see [35].

With this approach McSherry and Talwar obtain approximately-truthful auctions with nearly optimal selling price. Roughly speaking, this says that a participant cannot dramatically reduce the price he pays by lying about his valuation. Interestingly, they show that the simple composition of differential privacy can be used to obtain auctions in which no cooperating group of c agents can significantly increase their utility by submitting bids other than their true valuations. This is analagous to the situation of Remark 1 above, where composition is used to obtain privacy for groups of c individuals,

4 Algorithms for Specific Tasks

In this section we describe differentially private algorithms for three unrelated tasks.

4.1 Statistical Data Inference

The results in this Section are due to Dwork and Nissim [18].

Consider a setting in which each element in the database is described by a set of k Boolean attributes $\alpha_1, \ldots \alpha_k$, and the rows are independently sampled from some underlying distribution on $\{0,1\}^k$. Let $1 \leq \ell \leq k/2$ be an integer. The goal here is to use information about the incidence of settings of any ℓ attribute values to learn the incidence of settings of any 2ℓ attribute values.

Although we use the term "queries," these will all be known in advance, and the mechanism will be non-interactive. From something like $\binom{k}{\ell}2^\ell$ pieces of released information it will be possible to compute approximations to the incidence of all $\binom{k}{2\ell}2^{2\ell}$ minterms. This will allow the data analyst to approximate the probabilities of all $2^{\binom{k}{2\ell}2\ell}$ subsets of 2ℓ-ary minterms of length 2ℓ, provided the initial approximations are sufficiently accurate.

We will identify probability with incidence, so the probability space is over rows in the database. Fix any set of ℓ attributes. The incidence of all possible settings of these attribute values is described by a histogram with 2^ℓ cells, and histograms have sensitivity 1, so we are going to be working with a query sequence of overall sensitivity proportional to $\binom{k}{\ell}$ (in fact, it will be worse than this by a factor t, discussed below).

Let α and β be attributes. We say that α *implies* β *in probability* if the conditional probability of β given α exceeds the unconditional probability of β. The ability to measure implication in probability is crucial to datamining. Note that since $\Pr[\beta]$ is simple to estimate well using counting queries, the problem of measuring implication in probability reduces to obtaining a good estimate of $\Pr[\beta|\alpha]$. Moreover, once we can estimate $\Pr[\beta|\alpha]$, $\Pr[\beta]$, and $\Pr[a]$, we can use Bayes' Rule and de Morgan's Laws to determine the statistics for any Boolean function of

attribute values. For example, $\Pr[\alpha \wedge \beta] = \Pr[\alpha] \Pr[\beta|\alpha]$, so if we have estimates of the two multiplicands, within an additive η, we have an estimate for the products that is accurate within 3η.

As a step toward the non-interactive solution, consider the interactive case and assume that we have a good estimate for $\Pr[\alpha]$ and $\Pr[\beta]$. The key to determining $\Pr[\beta|\alpha]$ is to find a *heavy set for* α, that is, a set $q \subseteq [n]$ such that the incidence of α is at least, say, a standard deviation higher than expected, and then to determine whether the incidence of β on this heavy set is higher than the overall incidence of β. More specifically, one can test whether this conditional incidence is higher than a given threshold, and then use binary search to find the "right" threshold value. Finding the heavy set is easy because a randomly chosen subset of $[n]$ has constant probability of exceeding the expected incidence of α by at least one standard deviation.

To "simulate" the interactive case, the curator chooses some number of random subsets and for each one releases (noisy) estimates of the incidence of α and the incidence of β within this subset. With high probability (depending on t), at least one of the subsets is heavy for α.

Putting the pieces together: for t random subsets of $[n]$ the curator releases good approximations to the incidence of all $m = \binom{k}{\ell} 2^\ell$ conjunctions of ℓ literals. Specifically, we require that with probability at least $1 - \delta/m^2$ a computed implication in distribution $\Pr[\alpha|\beta]$ is accurate to within $\eta/3m^2$, where α and β are now minterms of ℓ literals. This ensures that with probability least $1 - \delta$ all computed implications in distribution are accurate to within $\eta/3m^2$, and so all estimated probabilities for minterms of 2ℓ literals are accurate to within η/m^2. The number t is rather large, and depends on many factors, including the differential privacy parameter ϵ as well as η, δ, k and ℓ. The analysis in [18] shows that, when η and δ are constant, this approach reduces the number of queries from $\binom{k}{2\ell}$ (one histogram for each 2ℓ-tuple of variables (not literals!)), to $O(2^{4\ell} k^\ell \ell^2 \log k)$. Note the interesting tradeoff: we require accuracy that depends on m^2 in order to avoid making m^2 queries. When the database is sufficiently large this tradeoff can be accomplished.

4.2 Contingency Table Release

The results in this Section are due to Barak, Chaudhuri, Dwork, Kale, McSherry, and Talwar [5]. A contingency table is a table of counts. In the context of a census or other survey, we think of the data of an individual as a *row* in a database. We do not assume the rows are mutually independent. For the present, each row consists of k bits describing the values of k binary attributes $a_1, \ldots, a_k$.[2] Formally, the contingency table is a vector in $\mathbb{R}^{2^k}$ describing, for each setting of the k attributes, the number of rows in the database with this setting of the attribute values. In other words, it is a histogram with 2^k cells.

Commonly, the contingency table itself is not released, as it is likely to be sparse when k is large. Instead, for various subsets of attributes, the data curator

[2] Typically, attributes are non-binary. Any attribute with m possible values can be decomposed into $\log(m)$ binary attributes.

releases the projection of the contingency table onto each such subset, *i.e.*, the counts for each of the possible settings of the restricted set of attributes. These smaller tables of counts are called marginals, each marginal being named by a subset of the attributes. A marginal named by a set of j attributes, $j \leq k$, is called a *j-way* marginal. The data curator will typically release many sets of low-order marginals for a single contingency table, with the goal of revealing correlations between many different, and possibly overlapping, sets of attributes.

Since a contingency table is a histogram, we can add independently generated noise proportional to ϵ^{-1} to each cell of the contingency table to obtain an ϵ-differentially private (non-integer and not necessarily non-negative) table. We will address the question of integrality and non-negativity later. For now, we simply note that any desired set of marginals can be computed directly from this noisy table, and consistency among the different marginals is immediate. A drawback of this approach, however, is that while the noise in each cell of the contingency table is relatively small, the noise in the computed marginals may be large. For example, the variance in the 1-way table describing attribute a_1 is $2^{k-1}\epsilon^{-2}$. We consider this unacceptable, especially when $n \ll 2^k$.

Marginals are also histograms. A second approach, with much less noise in the (common) case of low-order marginals, but not offering consistency between marginals, works as follows. Let C be the set of marginals to be released. We can think of a function f that, when applied to the database, yields the desired marginals. Now apply Theorem 1 with this choice of f, (adding noise to each cell in the collection of tables independently), with sensitivity $\Delta f = |C|$. When n (the number of rows in the database) is large compared to $|C|/\epsilon$, this also yields excellent accuracy. Thus we would be done if the small table-to-table inconsistencies caused by independent randomization of each (cell in each) table are not of concern, and if the user is comfortable with occasionally negative and typically non-integer cell counts.

We have no philosophical or mathematical objection to these artifacts – inconsistencies, negativity, and non-integrality – of the privacy-enhancing technology, but in practice they can be problematic. For example, the cell counts may be used as input to other, possibly off-the-shelf, programs that anticipate positive integers, giving rise to type mismatch. Inconsistencies, not to mention negative values, may also be confusing to lay users, such as casual users of the American FactFinder website.

We now outline the main steps in the work of Barak *et al* [5].

Move to the Fourier Domain. When adding noise, two natural solutions present themselves: adding noise to entries of the source table (this was our first proposal; accuracy is poor when k is large), or adding noise to the reported marginals (our second proposal; consistency is violated). A third approach begins by transforming the data into the Fourier domain. This is just a change of basis. Were we to compute all 2^k Fourier coefficients we would have a non-redundant encoding of the entire consistency table. If we were to perturb the Fourier coefficients and then convert back to the contingency table domain, we would get a (different, possibly non-integer, possibly negative) contingency table, whose "distance" (for example,

ℓ_2 distance) from the original is determined by the magnitude of the perturbations. The advantage of moving to the Fourier domain is that if only a set C of marginals is desired then we do not need the full complement of Fourier coefficients. For example, if C is the set of all 3-way marginals, then we need only the Fourier coefficients of weight at most 3, of which there are $\binom{k}{3} + \binom{k}{2} + k + 1$. This will translate into a much less noisy set of marginals.

The Fourier coefficients needed to compute the marginals C form a model of the dataset that captures everything that can be learned from the set C of marginals. Adding noise to these coefficients as indicated by Theorem 1 and then converting back to the contingency table domain yields a procedure for generating *synthetic datasets* that ensures differential privacy and yet to a great (and measurable) extent captures the information in the model. This is an example of a concrete method for generating synthetic data with provable differential privacy.

The Fourier coefficients exactly describe the information required by the marginals. By measuring exactly what is needed, Barak *et al.* add the least amount of noise possible using the techniques of [19]. Moreover, the Fourier basis is particularly attractive because of the natural decomposition according to sets of attribute values. Even tighter bounds than those in Theorem 4 below can be placed on sub-marginals (that is, lower order marginals) of a given marginal, by noting that no additional Fourier coefficients are required and fewer noisy coefficients are used in computing the low-order marginal, improving accuracy by reducing variance.

Use Linear Programming and Rounding. Barak *et al.* [5] employ linear programming to obtain a non-negative, but likely non-integer, data set with (almost) the given Fourier coefficients, and then round the results to obtain an integer solution. Interestingly, the marginals obtained from the linear program are no "farther" (made precise in [5]) from those of the noisy measurements than are the true marginals of the raw data. Consequently, the additional error introduced by the imposition of consistency is no more than the error introduced by the privacy mechanism itself.

Notation and Preliminaries. Recall that, letting k denote the number of (binary) attributes, we can think of the data set as a vector $x \in \mathbb{R}^{2^k}$, indexed by attribute tuples. For each $\alpha \in \{0,1\}^k$ the quantity x_α is the number of data elements with this setting of attributes. We let $n = \|x\|_1$ be the total number of tuples, or rows, in the data set.

For any $\alpha \in \{0,1\}^k$, we use $\|\alpha\|_1$ for the number of non-zero locations. We write $\beta \preceq \alpha$ for $\alpha, \beta \in \{0,1\}^k$ if every zero location in α is also a zero in β.

The Marginal Operator. Barak *et al.* describe the computation of a set of marginals as the result of applying a *marginal operator* to the contingency table vector x. The operator $C^\alpha : \mathbb{R}^{2^k} \to \mathbb{R}^{2^{\|\alpha\|_1}}$ for $\alpha \in \{0,1\}^k$ maps contingency tables to the marginal of the attributes that are positively set in α (there are $2^{\|\alpha\|_1}$ possible settings of these attributes). Abusing notation, $C^\alpha x$ is only defined at those

locations β for which $\beta \preceq \alpha$: for any $\beta \preceq \alpha$, the outcome of $C^\alpha x$ at position β is the sum over those coordinates of x that agree with β on the coordinates described by α:

$$(C^\alpha(x))_\beta = \sum_{\gamma:\gamma\wedge\alpha=\beta} x_\gamma \tag{3}$$

Notice that the operator C^α is linear for all α.

Theorem 2. *The f^α form an orthonormal basis for $\mathbb{R}^{2^k}$.*

Consequently, one can write any marginal as the small summation over relevant Fourier coefficients:

$$C^\beta x = \sum_{\alpha\preceq\beta} \langle f^\alpha, x\rangle C^\beta f^\alpha \ . \tag{4}$$

The coefficients $\langle f^\alpha, x\rangle$ are necessary and sufficient data from x for the computation of $C^\beta x$.

Theorem 3 ([5]). *Let $B \subseteq \{0,1\}^k$ describe a set of Fourier basis vectors. Releasing the set $\phi_\beta = \langle f^\beta, x\rangle + Lap(|B|/\epsilon 2^{k/2})$ for $\beta \in B$ preserves ϵ-differential privacy.*

Proof: Each tuple contributes exactly $\pm 1/2^{k/2}$ to each output coordinate, and consequently the L_1 sensitivity of the set of $|B|$ outputs is at most $|B|/2^{k/2}$. By Theorem 1, the addition of symmetric exponential noise with standard deviation $|B|/\epsilon 2^{k/2}$ gives ϵ-differential privacy.

Remark: To get a sense of scale, we could achieve a similar perturbation to each coordinate by randomly adding or deleting $|B|^2/\epsilon$ individuals in the data set, which can be much smaller than n.

Putting the Steps Together. To compute a set A of marginals, we need all the Fourier coefficients f^β for β in the downward closure of A uner $\preceq$.

Marginals$(A \subseteq \{0,1\}^k, D)$:

1. Let B be the downward closure of A under $\preceq$.
2. For $\beta \in B$, compute $\phi_\beta = \langle f^\beta, D\rangle + Lap(|B|/\epsilon 2^{k/2})$.
3. Solve for w_α in the following linear program, and round to the nearest integral weights, w'_α.

$$\begin{aligned} \text{minimize} \quad & b \\ \text{subject to:} \quad & \\ w_\alpha &\geq 0 \ \forall\alpha \\ \phi_\beta - \sum_\alpha w_\alpha f^\beta_\alpha &\leq b \ \forall\beta \in B \\ \phi_\beta - \sum_\alpha w_\alpha f^\beta_\alpha &\geq -b \ \forall\beta \in B \end{aligned}$$

4. Using the contingency table w'_α, compute and return the marginals for A.

Theorem 4 ([5]). *Using the notation of* ***Marginals***(A)*, with probability* $1-\delta$*, for all* $\alpha \in A$*,*

$$\|C^\alpha x - C^\alpha w'\|_1 \leq 2^{\|\alpha\|_1} 2|B| \log(|B|/\delta)/\epsilon + |B| \ . \tag{5}$$

When k is Large. The linear program requires time polynomial in 2^k. When k is large this is not satisfactory. However, somewhat surprisingly, non-negativity (but not integrality) can be achieved by adding a relatively small amount to the first Fourier coefficient before moving back to the data domain. No linear program is required, and the error introduced is pleasantly small. Thus if polynomial in 2^k is an unbearable cost and one can live with non-integrality then this approach serves well. We remark that non-integrality was a non-issue in a pilot implementation of this work as counts were always converted to percentages.

4.3 Learning (Nearby) Halfspaces

We close this Section with an example inspired by questions in learning theory, appearing in a forthcoming paper of Blum, Ligett, and Roth [8]. The goal is to give a non-interactive solution to half-space queries. At a high level, their approach is to publish information that (approximately) answers a large set of "canonical" queries of a certain type, with the guarantee that for any (possibly non-canonical) query of the given type there is a "nearby" canonical query. Hence, the data analyst can obtain the answer to a query that in some sense is close to the query of interest.

The queries in [8] are *halfspace* queries in $\mathbb{R}^d$, defined next. Throughout this section we adopt the assumption in [8] that the database points are scaled into the unit sphere.

Definition 3. *Given a database* $D \subset \mathbb{R}^d$ *and unit length* $y \in \mathbb{R}^d$*, a halfspace query* H_y *is*

$$H_y(D) = \frac{|\{x \in D : \sum_{i=1}^d x_i \cdot y_i \geq 0\}|}{|D|}.$$

Note that a halfspace query can be estimated from two counting queries: "What is $|D|$?" and "What is $|\{x \in D : \sum_{i=1}^d x_i \cdot y_i \geq 0\}|$?" Thus, the halfspace query has sensitivity at most 2.

The *distance* between halfspace queries H_{y_1} and H_{y_2} is defined to be the sine of the angle between them, $\sin(y_1, y_2)$. With this in mind, the algorithm of Blum, Ligett, and Roth, ensures the following notion of utility:

Definition 4 ([8]). *A database mechanism* A *is* $(\epsilon, \delta, \gamma)$-useful *for queries in class* C *according to some metric* d *if with probability* $1-\delta$*, for every* $Q \in C$ *and every database* D*,* $|Q(A(D)) - Q'(D)| \leq \epsilon$ *for some* $Q' \in C$ *such that* $d(Q, Q') \leq \gamma$*.*

Note that it is the queries that are close, not (necessarily) their answers.

Given a halfspace query H_{y_1}, the algorithm below will output a value v such that $|v - H_{y_2}(D)| < \epsilon$ for some H_{y_2} that is γ-close to H_{y_1}. Equivalently, the

algorithm arbitrarily counts or fails to count points $x \in D$ such that $\cos(x, y_1) \leq \gamma$. Blum *et al.* note that γ plays a role similar to the notion of margin in machine learning, and that even if H_{y_1} and H_{y_2} are γ-close, this does not imply that true answers to the queries $H_{y_1}(D)$ and $H_{y_2}(D)$ are close, unless most of the data points are outside a γ margin of H_{y_1} and H_{y_2}.

Definition 5 ([8]). *A halfspace query H_y is* b-discretized *if for each $i \in [d]$, y_i can be specified with b bits. Let C_b be the set of all b-discretized halfspaces in $\mathbb{R}^d$.*

Consider a particular $k < d$ dimensional subspace of $\mathbb{R}^d$ defined by a random $d \times k$ matrix M with entries chosen independently and uniformly from $\{-1, 1\}$. Consider the projection $P_M(x) = (1/\sqrt{k})x \cdot M$, which projects database points into the subspace and re-scales them to the unit sphere. For a halfspace query H_y, the projection $P_M(H_y)$ is simply the k-dimensional halfspace query defined by the projection $P_M(y)$. The key fact is that, for a randomly chosen M, projecting a database point x and the halfspace query specifier y is very unlikely to significantly change the angle between them:

Theorem 5 (Johnson-Lindenstrauss Theorem). *Consider a projection of a point x and a halfspace H_y onto a random k-dimensional subspace as defined by a projection matrix M. Then*

$$\Pr[|\cos(x, H_y) - \cos(P_M(x), H_{P_M(y)})| \geq \gamma/4] \leq 2e^{-((\gamma/16)^2 - (\gamma/16)^3)k/4}.$$

The dimension k of the subspace is chosen such that the probability that projecting a point and a halfspace changes the angle between them by more than $\gamma/4$ is at most $\epsilon_1/4$. This yields

$$k \geq \frac{4\ln(8/\epsilon_1)}{(\gamma/16)^2 - (\gamma/16)^3}.$$

Thus, the answer to the query H_y can be estimated by a privacy-preserving estimate of the answer to the projected halfspace query, and overall accuracy could be improved by choosing m projection matrices; the angle between x and y would be estimated by the median of the angles induces by the m resulting pairs of projections of x and y.

Of course, if the goal were to respond to a few half-space queries there would be no point in going through the projection process, let alone taking several projections. But the goal of [8] is more ambitious: an$(\epsilon, \delta, \gamma)$-*useful* non-interactive mechanism for (non-discretized) halfspace queries; this is where the lower dimensionality comes into play.

The algorithm chooses m projection matrices, where m depends on the discretization parameter b, the dimension d, and the failure probability δ (more specifically, $m \in O(\ln(1/\delta) + \ln(bd))$). For each random subspace (defined by a projection matrix M), the algorithm selects a *net* N_M of "canonical" halfspaces (defined by canonical vectors in the subspace) such that for every vector $y \in R^k$ there is a nearby canonical vector, specifically, of distance (induced sine) at most $(3/4)\gamma$.

The number of canonical vectors needed is $O(1/\gamma^{k-1})$. For each of these, the curator publishes a privacy-preserving estimate of the projected halfspace query. The mechanism is non-interactive and the curator will play no further role.

To handle an arbitrary query y, the analyst begins with an empty multiset. For each of the m projections M, the analyst finds a vector $\hat{y} \in N_M$ closest to $P_M(y)$, adding to the multiset the answer to that halfspace query. The algorithm outputs the median of these m values.

Theorem 6 ([8]). *Let*

$$n \geq \frac{\log(1/\delta) + \log m + (k-1)\log(1/\gamma) + mO(1/\gamma)^{k-1}}{\epsilon_2 \alpha}$$

Then the above algorithm is $(\epsilon, \gamma, \delta)$-useful while maintaining α-differential privacy for a database of size n. The algorithm runs in time poly$(\log(1/\delta), 1/\epsilon, 1/\alpha, b, d)$ *for constant γ.*

5 General Learning Theory Results

We briefly outline three general results regarding what can be learned privately in the interactive model.

We begin with a result of Blum, Dwork, McSherry and Nissim, showing that anything learnable in the statistical queries learning model can also be efficiently privately learned interactively [7]. We then move to results that ignore computational issues, showing that the exponential mechanism of McSherry and Talwar [35] can be used to

1. Privately learn anything that is PAC learnable [32]; and
2. Generate, for any class of functions C with polynomial VC dimension, a differentially private "synthetic database" that gives "good" answers to any query in C [8].

The use of the exponential mechanism in this context is due to Kasiviswanathan, Lee, Nissim, Raskhodnikova, and Smith [32].

5.1 Emulating the Statistical Query Model

The Statistical Query (SQ) model, proposed by Kearns in [10], is a framework for examining statistical algorithms executed on samples drawn independently from an underlying distribution. In this framework, an algorithm specifies predicates $f_1, \ldots, f_k$ and corresponding accuracies $\tau_1, \ldots, \tau_k$, and is returned, for $1 \leq i \leq k$, the expected fraction of samples satisfying f_i, to within additive error τ_i. Conceptually, the framework models drawing a sufficient number of samples so that the observed count of samples satisfying each f_i is a good estimate of the actual expectation.

The statistic/al queries model is most commonly used in the computational learning theory community, where the goal is typically to learn a (in this case, Boolean) *concept*, that is, a predicate on the data, to within a certain degree of

accuracy. Formally, an algorithm δ-learns a concept c if it produces a predicate such that the probability of misclassification under the latent distribution is at most $1 - \delta$.

Blum, Dwork, McSherry, and Nissim have shown that any concept that is learnable in the statistical query model is privately learnable using the equivalent algorithm on a so-called "SuLQ" database [7]; the proof is not at all complicated. Reformulting these results using the slightly different technology presented in the current survey the result is even easier to argue.

Assume we have an algorithm in the SQ model that makes the k statistical queries $f_1, \ldots, f_k$, and let $\tau = \min\{\tau_1, \ldots, \tau_k\}$ be the minimum required tolerance in the SQ algorithm. Assume that n, the size of the database, is known. In this case, we can compute the answer to f_i by asking the predicate/counting query corresponding to f_i, call it p_i, and dividing the result by n. Thus, we are dealing with a query squence of sensitivity at most k.

Let $b = k/\epsilon$. Write $\tau = \rho/n$, for ρ to be determined later, so that if the noise added to the true answer when the counting query is p_i has magnitude bounded by ρ, the response to the statistical query is within tolerance τ.

We want to find ρ so that the probability that a response has noise magnitude at least ρ is bounded by δ/k, when the noise is generated according to $\mathrm{Lap}(k/\epsilon)$. The Laplace distribution is symmetric, so it is enough to find $x < 0$ such that the cumulative distribution function at x is bounded by $\delta/2k$:

$$\frac{1}{2}e^{-|x|/b} < \frac{\delta}{2k}$$

By a straightforward calculation, this is true provided $|x| > b\ln(k/\delta)$, ie, when $|x| > (k/\epsilon)\ln(k/\delta)$. We therefore set $\rho > (k/\epsilon)\ln(k/\delta)$. So long as $\rho/n < \tau$, or, more to the point, so long as $n > \rho/\tau$, we can emulate the SQ algorithm.

This analysis only takes into account noise introduced by $\mathcal{K}$; that is, it assumes $\frac{1}{n}\sum_{i=1}^{n} f_j(d_i) = \Pr_{x \in_R \mathcal{D}}[f_j(x)]$, $1 \leq j \leq k$, where $\mathcal{D}$ is the distribution on examples. The results above apply, *mutatis mutandis*, when we assume that the rows in the database are drawn iid according to $\mathcal{D}$ using the well known fact that taking $n > \tau^{-2}\log(k/\delta)$ is sufficient to ensure tolerance τ with probability at least $1 - \delta$ for all f_j, $1 \leq j \leq k$, simultaneously. Replacing τ by $\tau/2$ everywhere and finding the maximum lower bound on n handles both types of error.

5.2 Private PAC Learning

The results in this section are due to Kasiviswanathan, Lee, Nissim, Raskhodnikova, and Smith [32]. We begin with an informal and incomplete review of the concept of probably-appproximately-correct (PAC) learning, a notion due to Valiant [47].

Consider a concept $t : X \longrightarrow Y$ that assigns to each *example* taken from the domain X a *label* from the range Y. As in the previous section, a learning algorithm is given labeled examples drawn from a distribution $\mathcal{D}$, labeled by a target concept; the goal is to produce an *hypothesis* $h : X \longrightarrow Y$ from a specified hypothesis class, with small *error*, defined by:

$$\text{error}(h) = \Pr_{x \in_R \mathcal{D}}[t(x) \neq h(x)].$$

A *concept class* is a set of concepts. Learning theory asks what kinds of concept classes are learnable. Letting α, β denote two error bounds, if the target concept belongs to C, then the goal is to minimize error, or at least to ensure that with probability $1 - \beta$ the error is bounded by α. This is the setting of traditional PAC learning. If the target concept need not belong to C, the goal is to produce an hypothesis that, intuitively, does almost as well as any concept in C: Letting $\text{OPT} = \min_{c \in C}\{\text{error}(c)\}$, we want

$$\Pr[\text{error}(h) \leq \text{OPT} + \alpha] \geq 1 - \beta,$$

where the probability is over the samples seen by the learner, and the learner's randomness. This is known as *agnostic learning*.[3]

Following [32] we will index concept classes, domains, and ranges by the length d of their binary encodings. For a target concept $t : X_d \to Y_d$ and a distribution $\mathcal{D}$ over X_d, let $Z \in D^n$ be a database containing n labeled independent draws from $\mathcal{D}$. That is, Z contains n pairs (x_i, y_i) where $y_i = t(x_i)$, $1 \leq i \leq n$. The goal of the data analyst will be to agnostically learn a hypothesis class C; the goal of the curator will be to ensure ϵ-differential privacy.

Theorem 7 ([32]). *Every concept class C is ϵ-differentially pivately agnostically learnable using hypothesis class $\mathcal{H} = C$ with $n \in O(\log C_d + \log(1\beta) \cdot \max\{\frac{1}{\epsilon\alpha}, \frac{1}{\alpha^2}\})$. The learner may not be efficient.*

The theorem is proved using the exponential mechanism of McSherry and Talwar, with utility function

$$u(Z, h) = -|\{i : t(x_i) \neq h(x_i)\}|$$

for $Z \in (X \times Y)^n, h \in \mathcal{H}_d$. Note that u has sensitivity 1, since changing any element in the database can change the number of misclassifications by at most 1. The (inefficient) algorithm outputs $c \in \mathcal{H}_d$ with probability proportional to $\exp(\epsilon u(Z, c)/2)$. Privacy follows from the properties of the exponential mechanism and the small sensitivity of u. Accuracy (low error with high probablity) is slightly more difficult to argue; the proof follows, intuitively, from the fact that outputs are produced with probability that falls exponentially in the number of misclassifications.

5.3 Differentially Private Queries of Classes with Polynomial VC Dimension

Our last example is due to Blum, Ligett, and Roth [8] and was inspired by the result of the previous section, This learning result again ignores computational

[3] The standard agnostic model has the input drawn from an arbitrary distribution over labeled examples (x, y) (that is, the label need not be a deterministic function of the example). The error of a hypothesis is defined with respect to the distribution (i.e. probability that $y \neq h(x)$). The results (and proofs) of Kasiviswanathan *et al.* stay the same in this more general setting.

efficiency, using the exponential mechanism of McSherry and Talwar. The object here is to release a "synthetic dataset" that ensures "reasonably" accurate answers to *all* queries in a specific class C. The reader is assumed to be familiar with the Vapnick-Chervonenkis (VC) dimension of a class of concepts. Roughly speaking, it is a measure of how complicated a concept in the class can be.

Definition 6 ([8]). *A database mechanism A is* (γ, δ)-useful *for queries in class C if with probability $1-\delta$, for every $Q \in C$ and every database D, for $\widehat{D} = A(D)$, $|Q(\widehat{D}) - Q(D)| \leq \gamma$.*

Let C be a fixed class of queries. Given a database $D \in (\{0,1\}^d)^n$ of size n, where n is sufficiently large (as a function of the VC dimension of C, as well as of ϵ and δ), the goal is to produce produce a synthetic dataset $\hat{D}$ that is (γ, δ)-useful for queries in C, while ensuring ϵ-differential privacy.

The synthetic dataset will contain $m = O(\text{VC dim}(C)/\gamma^2)$ d-tuples. It is chosen according to the exponential mechanism using the utility function

$$u(D, \hat{D}) = -\max_{h \in C} \left| h(D) - \frac{n}{m} h(\hat{D}) \right|.$$

Theorem 8 ([8]). *For any class of functions C, and any database $D \subset \{0,1\}^d$ such that*

$$|D| \geq O\left(\frac{d \cdot \text{VC dim}(C)}{\gamma^3 \epsilon} + \frac{\log(1/\delta)}{\epsilon\gamma} \right)$$

we can output an (γ, δ)-useful database $\widehat{D}$ that preserves ϵ-differential privacy. Note that the algorithm is not necessarily efficient.

Blum *et al.* note that this suffice for (γ, δ)-usefulness because the set of all databases of this size forms a γ-cover with respect to C of the set of all possible databases. One can resolve the fact that, since $|\hat{D}| < |D|$, the number of database entries matching any query will be proportionately smaller by considering the fraction of entries matching any query.

6 Concluding Remarks

The privacy mechanisms discussed herein add an amount of noise that grows with the complexity of the query sequence applied to the database. Although this can be ameliorated to some extent using Gaussian noise instead of Laplacian, an exciting line of research begun by Dinur and Nissim [13] (see also [17, 20]) shows that this increase is essential. To a great extent, the results of Dinur and Nissim drove the development of the mechanism $\mathcal{K}$ and the entire interactive approach advocated in this survey. A finer analysis of *realistic* attacks, and a better understanding of what failure to provide ϵ-differential privacy can mean in practice, are needed in order to sharpen these results – or to determine this is impossible, in order to understand the how to use these techniques for all but very large, "internet scale," data sets.

Acknowledgements. The author thanks Avrim Blum, Katrina Ligett, Aaron Roth and Adam Smith for helpful technical discussions, and Adam Smith for excellent comments on an early draft of this survey. The author is grateful to all the authors of [32, 8] for sharing with her preliminary versions of their papers.

References

[1] Achugbue, J.O., Chin, F.Y.: The Effectiveness of Output Modification by Rounding for Protection of Statistical Databases. INFOR 17(3), 209–218 (1979)
[2] Adam, N.R., Wortmann, J.C.: Security-Control Methods for Statistical Databases: A Comparative Study. ACM Computing Surveys 21(4), 515–556 (1989)
[3] Agrawal, D., Aggarwal, C.: On the Design and Quantification of Privacy Preserving Data Mining Algorithms. In: Proceedings of the 20th Symposium on Principles of Database Systems (2001)
[4] Agrawal, R., Srikant, R.: Privacy-Preserving Data Mining. In: Proceedings of the ACM SIGMOD Conference on Management of Data, pp. 439–450 (2000)
[5] Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., Talwar, K.: Privacy, Accuracy, and Consistency Too: A Holistic Solution to Contingency Table Release. In: Proceedings of the 26th Symposium on Principles of Database Systems, pp. 273–282 (2007)
[6] Beck, L.L.: A Security Mechanism for Statistical Databases. ACM TODS 5(3), 316–338 (1980)
[7] Blum, A., Dwork, C., McSherry, F., Nissim, K.: Practical Privacy: The SuLQ framework. In: Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (June 2005)
[8] Blum, A., Ligett, K., Roth, A.: A Learning Theory Approach to Non-Interactive Database Privacy. In: Proceedings of the 40th ACM SIGACT Symposium on Thoery of Computing (2008)
[9] Chawla, S., Dwork, C., McSherry, F., Smith, A., Wee, H.: Toward Privacy in Public Databases. In: Proceedings of the 2nd Theory of Cryptography Conference (2005)
[10] Dalenius, T.: Towards a methodology for statistical disclosure control. Statistik Tidskrift 15, 222–429 (1977)
[11] Denning, D.E.: Secure Statistical Databases with Random Sample Queries. ACM Transactions on Database Systems 5(3), 291–315 (1980)
[12] Denning, D., Denning, P., Schwartz, M.: The Tracker: A Threat to Statistical Database Security. ACM Transactions on Database Systems 4(1), 76–96 (1979)
[13] Dinur, I., Nissim, K.: Revealing Information While Preserving Privacy. In: Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 202–210 (2003)
[14] Duncan, G.: Confidentiality and statistical disclosure limitation. In: Smelser, N., Baltes, P. (eds.) International Encyclopedia of the Social and Behavioral Sciences, Elsevier, New York (2001)
[15] Dwork, C.: Differential Privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006)
[16] Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our Data, Ourselves: Privacy Via Distributed Noise Generation. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 486–503. Springer, Heidelberg (2006)

[17] Dwork, C., McSherry, F., Talwar, K.: The Price of Privacy and the Limits of LP Decoding. In: Proceedings of the 39th ACM Symposium on Theory of Computing, pp. 85–94 (2007)
[18] Dwork, C., Nissim, K.: Privacy-Preserving Datamining on Vertically Partitioned Databases. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 528–544. Springer, Heidelberg (2004)
[19] Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating Noise to Sensitivity in Private Data Analysis. In: Proceedings of the 3rd Theory of Cryptography Conference, pp. 265–284 (2006)
[20] Dwork, C., Yekhanin, S.: New Efficient Attacks on Statistical Disclosure Control Mechanisms (manuscript, 2008)
[21] Evfimievski, A.V., Gehrke, J., Srikant, R.: Limiting Privacy Breaches in Privacy Preserving Data Mining. In: Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 211–222 (2003)
[22] Agrawal, D., Aggarwal, C.C.: On the design and Quantification of Privacy Preserving Data Mining Algorithms. In: Proceedings of the 20th Symposium on Principles of Database Systems, pp. 247–255 (2001)
[23] Agrawal, R., Srikant, R.: Privacy-Preserving Data Mining. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 439–450 (2000)
[24] Chin, F.Y., Ozsoyoglu, G.: Auditing and infrence control in statistical databases. IEEE Trans. Softw. Eng. SE-8(6), 113–139 (1982)
[25] Dobkin, D., Jones, A., Lipton, R.: Secure Databases: Protection Against User Influence. ACM TODS 4(1), 97–106 (1979)
[26] Fellegi, I.: On the question of statistical confidentiality. Journal of the American Statistical Association 67, 7–18 (1972)
[27] Fienberg, S.: Confidentiality and Data Protection Through Disclosure Limitation: Evolving Principles and Technical Advances. In: IAOS Conference on Statistics, Development and Human Rights (September 2000), http://www.statistik.admin.ch/about/international/fienberg_final_paper.doc
[28] Fienberg, S., Makov, U., Steele, R.: Disclosure Limitation and Related Methods for Categorical Data. Journal of Official Statistics 14, 485–502 (1998)
[29] Franconi, L., Merola, G.: Implementing Statistical Disclosure Control for Aggregated Data Released Via Remote Access. In: United Nations Statistical Commission and European Commission, joint ECE/EUROSTAT work session on statistical data confidentiality, Working Paper No.30 (April 2003), http://www.unece.org/stats/documents/2003/04/confidentiality/wp.30.e.pdf
[30] Goldwasser, S., Micali, S.: Probabilistic Encryption. J. Comput. Syst. Sci. 28(2), 270–299 (1984)
[31] Gusfield, D.: A Graph Theoretic Approach to Statistical Data Security. SIAM J. Comput. 17(3), 552–571 (1988)
[32] Kasiviswanathan, S., Lee, H., Nissim, K., Raskhodnikova, S., Smith, S.: What Can We Learn Privately? (manuscript, 2007)
[33] Lefons, E., Silvestri, A., Tangorra, F.: An analytic approach to statistical databases. In: 9th Int. Conf. Very Large Data Bases, October-November 1983, pp. 260–274. Morgan Kaufmann, San Francisco (1983)
[34] Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-Diversity: Privacy Beyond k-Anonymity. In: Proceedings of the 22nd International Conference on Data Engineering (ICDE 2006), p. 24 (2006)

[35] McSherry, F., Talwar, K.: Mechanism Design via Differential Privacy. In: Proceedings of the 48th Annual Symposium on Foundations of Computer Science (2007)
[36] Narayanan, A., Shmatikov, V.: How to Break Anonymity of the Netflix Prize Dataset, http://www.cs.utexas.edu/~shmat/shmat_netflix-prelim.pdf
[37] Nissim, K., Raskhodnikova, S., Smith, A.: Smooth Sensitivity and Sampling in Private Data Analysis. In: Proceedings of the 39th ACM Symposium on Theory of Computing, pp. 75–84 (2007)
[38] Raghunathan, T.E., Reiter, J.P., Rubin, D.B.: Multiple Imputation for Statistical Disclosure Limitation. Journal of Official Statistics 19(1), 1–16 (2003)
[39] Reiss, S.: Practical Data Swapping: The First Steps. ACM Transactions on Database Systems 9(1), 20–37 (1984)
[40] Rubin, D.B.: Discussion: Statistical Disclosure Limitation. Journal of Official Statistics 9(2), 461–469 (1993)
[41] Shoshani, A.: Statistical databases: Characteristics, problems and some solutions. In: Proceedings of the 8th International Conference on Very Large Data Bases (VLDB 1982), pp. 208–222 (1982)
[42] Samarati, P., Sweeney, L.: Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement Through Generalization and Specialization, Technical Report SRI-CSL-98-04, SRI Intl. (1998)
[43] Samarati, P., Sweeney, L.: Generalizing Data to Provide Anonymity when Disclosing Information (Abstract). In: Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, p. 188 (1998)
[44] Sweeney, L.: Weaving Technology and Policy Together to Maintain Confidentiality. J. Law Med Ethics 25(2-3), 98–110 (1997)
[45] Sweeney, L.: k-anonymity: A Model for Protecting Privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10(5), 557–570 (2002)
[46] Sweeney, L.: Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems 10(5), 571–588 (2002)
[47] Valiant, L.G.: A Theory of the Learnable. In: Proceedings of the 16th Annual ACM SIGACT Symposium on Theory of computing, pp. 436–445 (1984)
[48] Xiao, X., Tao, Y.: M-invariance: towards privacy preserving re-publication of dynamic datasets. In: SIGMOD 2007, pp. 689–700 (2007)