Journal of Privacy and Confidentiality (2016-2017) **7**, Number 3, 17–51

Calibrating Noise to Sensitivity in Private Data Analysis

Cynthia Dwork*, Frank McSherry†, Kobbi Nissim‡ and Adam Smith§

We continue a line of research initiated in Dinur and Nissim (2003); Dwork and Nissim (2004); Blum et al. (2005) on privacy-preserving statistical databases.

Consider a trusted server that holds a database of sensitive information. Given a query function f mapping databases to reals, the so-called *true answer* is the result of applying f to the database. To protect privacy, the true answer is perturbed by the addition of random noise generated according to a carefully chosen distribution, and this response, the true answer plus noise, is returned to the user.

Previous work focused on the case of noisy sums, in which $f = \sum_i g(x_i)$, where x_i denotes the ith row of the database and g maps database rows to $[0, 1]$. We extend the study to general functions f, proving that privacy can be preserved by calibrating the standard deviation of the noise according to the *sensitivity* of the function f. Roughly speaking, this is the amount that any single argument to f can change its output. The new analysis shows that for several particular applications substantially less noise is needed than was previously understood to be the case.

The first step is a very clean definition of privacy—now known as differential privacy—and measure of its loss. We also provide a set of tools for designing and combining differentially private algorithms, permitting the construction of complex differentially private analytical tools from simple differentially private primitives.

Finally, we obtain separation results showing the increased value of interactive statistical release mechanisms over non-interactive ones.

1 Introduction

We continue a line of research initiated by Dinur and Nissim (2003) on privacy in statistical databases. A statistic is a quantity computed from a sample. Intuitively, if the database is a representative sample of an underlying population, the goal of a privacy-preserving statistical database is to enable an analyst to learn properties of the population as a whole while protecting the privacy of the individual contributors.

A preliminary version of this work in the proceedings of TCC 2006 (Dwork et al., 2006b)

*Microsoft Research SVC, mailto:dwork@microsoft.com

†This work was done while the author was at Microsoft Research SVC.

‡This work was done while the author was at the Department of Computer Science, Ben-Gurion University, mailto:kobbi@cs.bgu.ac.il

§This work was done while the author was at the Weizmann Institute of Science, supported by a Louis L. and Anita M. Perlman Postdoctoral Fellowship. At Penn State, Adam Smith is supported by awards from the NSF (IIS-1447700), Google and the Sloan Foundation, Pennsylvania State University, asmith@psu.edu

We assume the database is held by a trusted server, or *curator*, who will release information about the database, either as a single-shot release (the *noninteractive* model), or interactively, in response to a sequence of queries from analysts. We ask: what conditions can we place on the algorithm, or "mechanism", run by the server in order to be guaranteed that not too much is revealed about any one individual?

Previous work focused on the case of noisy sums, in which the server aims to release a statistic of the form $f(\mathbf{x}) = \sum_i g(x_i)$, where x_i denotes the ith row of the database $\mathbf{x}$ and g maps database rows to $[0, 1]$. We extend the study to general functions f, proving that privacy can be preserved by calibrating the standard deviation of the noise according to the *sensitivity* of the function f. Roughly speaking, this is the amount that a change in any single argument to f can change its output. The new analysis shows that for several particular applications substantially less noise is needed than was previously understood to be the case.

Our starting point is a new definition of privacy, *ε-differential privacy*[1]. An interaction between a user and a privacy mechanism results in a *transcript*. For now it is sufficient to think of transcripts corresponding to a single query function and response, but the notion is general and our results apply to transcripts that result from interaction between an analyst and a server responding to queries.

Roughly speaking, a mechanism is ε-differentially private if for all transcripts t and for all databases $\mathbf{x}$ and $\mathbf{x}'$ differing in a single row, the probability of obtaining transcript t when the database is $\mathbf{x}$ is within a $(1 + \varepsilon)$ multiplicative factor of the probability of obtaining transcript t when the database is $\mathbf{x}'$. More precisely, we require that the ratio of the two probabilities lie in $[e^{-\varepsilon}, e^{\varepsilon}]$. In our work, ε is a parameter chosen by policy.

We then formally define the sensitivity $S(f)$ of a function f. This is a quantity *inherent* in f; it is not chosen by policy and is independent of the database.

We show a simple method of adding noise that ensures ε-differential privacy; the noise depends only on ε. Specifically, to obtain ε-differential privacy it suffices to add noise according to the Laplace distribution, where $Pr[y] \propto e^{-\varepsilon|y|/S(f)}$.

The extension to privacy-preserving approximations to "holistic" functions f that operate on the entire database broadens the scope of private data analysis beyond the original motivation of a purely statistical, or "sample population" context. Now we can view the database as an object that is itself of intrinsic interest and that we wish to analyze in a privacy-preserving fashion. For example, the database may describe a concrete interconnection network—not a sample subnetwork—and we wish to learn certain properties of the network without releasing information about individual edges, nodes, or subnetworks. The technology developed herein therefore extends the scope of the line of research, beyond privacy-preserving statistical databases to privacy-preserving analysis of data.

Differential privacy, a stronger notion than appears in previous work Dinur and Nissim (2003); Dwork and Nissim (2004); Blum et al. (2005); Sweeney (2002), satisfies

[1] The original version of this paper used the term *ε-indistinguishability*.

a number of important properties. Foremost among these are closure under composition and postprocessing. It is these properties that permit differentially private *programming*, that is, the construction of privacy-preserving algorithms for sophisticated analytical tasks from the creative combination of differentially private primitives.

1.1 Contributions

Definitions of Privacy (Section 2) Definition of privacy requires care. We prove equivalence of differential privacy to notions based on semantic security and simulation. The definitions in previous work Dinur and Nissim (2003); Dwork and Nissim (2004); Blum et al. (2005) focused on "evolution of confidence" arguments that captured changes to an adversary's ability to distinguish among different possible values of the data of a single individual as the number of functions evaluated on the dataset increased. Our introduction of differential privacy, with its succinct bounds on cumulative privacy loss, allows substantial simplification of those analyses.

Databases $\mathbf{x}$ and $\mathbf{x}'$ are *adjacent* if one is a subset of the other, and the larger contains the data of just one additional individual. The new formulation has the following appealing interpretation: no matter what an adversary knows ahead of time—even if the adversary knows both $\mathbf{x}$ and $\mathbf{x}'$, the adversary learns essentially the same things, with essentially the same probabilities, independent of whether the actual dataset is $\mathbf{x}$ or $\mathbf{x}'$. For example, suppose only the larger dataset contains the data of Alice, and suppose the adversary knows that Alice is a smoker, but does not know whether or not she has lung disease. Seeing the outcome of a differentially private study could teach the adversary that there is a strong correlation between smoking and lung disease among the population sampled to create the data set. This is simply learning about the population, and such facts of life are the purpose of many studies and learning techniques. This particular learned fact allows the adveresary to conclude that Alice is at high risk for lung disease. However, differential privacy ensures that the attacker would draw this conclusion about Alice *whether or not Alice's data were included in the data set*. Differential privacy limits the adversary to these kinds of aggregate conclusions.

Differential privacy, which is multiplicative, differs from traditional definitions in cryptography, which consider additive changes in probabilities. A simple hybrid argument shows that, in our context, any nontrivial utility requires nonnegligible (in n, the size of the database) information leakage. The multiplicative measure provides meaningful guarantees even when ε is a small constant. We give an example highlighting why more standard measures, such as statistical difference, are insufficient in our setting and need to be replaced with a more discriminating one.

Examples of Sensitivity-Based Analysis (Section 3) We analyze the sensitivity of specific data analysis functions, including histograms, contingency tables, and covariance matrices, all of which have very high-dimensional output, showing these are independent of the dimension. Previous privacy-preserving approximations to

these quantities used noise proportional to the dimension; the new analysis permits noise of size $O(1)$. We also give two general classes of functions which have low sensitivity: functions which estimate distance from a set (e.g., the number of points that need to be deleted for a data set to be well-clustered) and functions which can be approximated from a random sample.

Limits on Non-Interactive Mechanisms (Section 4) There are two natural models for privacy-preserving data analysis: interactive and non-interactive. In the non-interactive setting, the curator—a trusted entity—publishes a "sanitized" version of the collected data; the literature uses terms such as "anonymization," "de-identification," and "synthetic data". Traditionally, sanitization employed some perturbation and data modification techniques, and may also have included some accompanying synopses and statistics. In the interactive setting, the data collector provides a mechanism with which users may pose queries about the data, and receive (possibly noisy) responses.

The first of these appears to be more difficult (see Evfimievski et al. (2003); Chawla et al. (2005a;b)), since any potential release must be useful for essentially all possible analyses. This would contradict powerful negative results that "overly accurate" answers to "too many" queries is blatantly non-private Dinur and Nissim (2003), meaning an attacker can reconstruct large parts of the data set. In contrast, powerful results for the interactive approach have been obtained (Dinur and Nissim (2003); Dwork and Nissim (2004); Blum et al. (2005) and the present paper).

We show that for any noninteractive differentially private mechanism $\mathcal{M}$, there exist low-sensitivity functions $f(\mathbf{x})$ which cannot be approximated at all based on $\mathcal{M}(\mathbf{x})$, unless the database is very large: If each database entry consists of d bits, then the database must have $2^{\Omega(d)}$ entries in order to answer all low-sensitivity queries with nontrivial accuracy—even to answer queries from a restricted class called *sum queries*. In other words, a noninteractive mechanism must be tailored to suit certain functions to the exclusion of others. This is not true in the interactive setting, since one can answer the query f with little noise regardless of n.

The separation results are significant given that the data-mining and statistical literature has focused almost exclusively on non-interactive mechanisms, and that many of these mechanisms fall into the "local", or "randomized response" framework (see Related Work below).

1.2 Related Work

The literature in statistics and computer science on disseminating statistical data while preserving privacy is extensive; we discuss only directly relevant work here.

Privacy from Perturbation. The venerable idea of achieving privacy by adding noise is both natural and appealing. An excellent and detailed exposition of the many variants of this approach explored in the context of statistical disclosure control un-

til 1989, many of which are still important elements of the toolkit for data privacy today, may be found in the survey of Adam and Wortmann Adam and Wortmann (1989). The "classical" antecedent closest in spirit to our approach is the work of Denning Denning (1980).

Perturbation techniques are classified into two basic categories: (i) *Input perturbation techniques*, where the underlying data are randomly modified, and answers to questions are computed using the modified data; and (ii) *Output perturbation*, where (correct) answers to queries are computed exactly from the real data, but noisy versions of these are reported. Both techniques suffer from certain inherent limitations (see below); it seems that these limitations caused a decline in interest within the computer science community in designing perturbation techniques for achieving privacy.

The work of Agrawal and Srikant Agrawal and Srikant (2000) rekindled this interest; their principal contribution was an algorithm that, given an input-perturbed database, learns the original input distribution. Subsequent work studied the applicability and limitations of perturbation techniques, and privacy definitions have started to evolve, as we next describe.

Definitional Work. Several privacy definitions have been put forward since Agrawal and Srikant (2000). Their definition measured privacy in terms of the noise magnitude added to a value. This was shown to be problematic, as the definition ignored what an adversary knowing the underlying probability distribution might infer about the data Agrawal and Aggarwal (2001). Evfimievsky et al. Evfimievski et al. (2003) noted, however, that such an *average* measure allows for infrequent but noticeable privacy breaches, and suggested measuring privacy in terms of the *worst-case* change in an adversary's *a priori* to *a posteriori* beliefs. Their definition is a special case of Definition 2.1 for input perturbation protocols of a limited form. A similar, more general, definition was suggested in Dinur and Nissim (2003); Dwork and Nissim (2004); Blum et al. (2005). This was modeled after semantic security of encryption.

Our basic definition of privacy, ε-differential privacy, requires that a change in one database entry induce a small change in the distribution on the view of the adversary, under a specific, "worst-case" measure of distance. It is the same as in Evfimievski et al. (2003), adapted to general interactive protocols. An equivalent, semantic security-flavored formulation is very close to the definitions from Dinur and Nissim (2003); Dwork and Nissim (2004); Blum et al. (2005); those definitions allowed a large loss of privacy to occur with negligible probability.

We note that *k-anonymity* Sweeney (2002) and the similarly motivated notion of protection against *isolation* Chawla et al. (2005a;b) have also been in the eye of privacy research. The former is a syntactic characterization of (input-perturbed) databases that does not immediately capture semantic notions of privacy; the latter definition is a geometric interpretation of protection against being brought to the attention of others. The techniques described herein yield protection against isolation.

Sum Queries. A cryptographic perspective on perturbation was initiated by Dinur and Nissim Dinur and Nissim (2003). They studied the amount of noise needed to maintain privacy in databases where a query returns (approximately) the number of 1's in any given subset of the entries. They showed that if queries are not restricted, the amount of noise added to each answer must be very high—linear (in n, the size of the database) for the case of a computationally unbounded adversary, and $\Omega(\sqrt{n})$ for a polynomially (in n) bounded adversary. Otherwise, the adversary can reconstruct the database almost exactly, producing a database that errs on, say, 0.01% of the entries. In contrast, jointly with Dwork, they initiated a sequence of work Dinur and Nissim (2003); Dwork and Nissim (2004); Blum et al. (2005) which showed that limiting the users to a sublinear (in n) number of queries ("SuLQ") allows one to release useful global information while satisfying a strong definition of privacy. For example, it was shown that the computationally powerful noisy sum queries discussed above, that is, $\sum_{i=1}^{n} g(i, x_i)$, where g maps rows to values in $[0, 1]$, can be safely answered by adding $o(\sqrt{n})$ noise (from a gaussian, binomial, or Laplace distribution)—a level well below the sampling error one would expect in the database initially. Dwork and Nissim (2004); Blum et al. (2005) also introduced the concept of a privacy-preserving computation, specifically, noise sums, as a computational primitive, a precursor of the programmable nature of differential privacy.

1.3 Discussion

Since the appearance of the preliminary version of this paper Dwork et al. (2006b), differential privacy has become a central concept in research on data privacy. By providing the first definition that ensures precise and meaningful guarantees in the presence of arbitrary side information, it has given rise to a large body of work that spans many areas of computer science as well as statistics, economics, law and policy. An impassioned argument for the choice of definition appears in Dwork (2006); Dwork and Naor (2010).

The first large scale public implementation of a variation of differential privacy is in the US Census Bureau's OnTheMap, a mapping and reporting tool integrating administrative records with census and survey data Machanavajjhala et al. (2008). Differential privacy in the local model, in which data are randomized to satisfy differential privacy *before* collection (see Example 1 below) has been deployed for browser telemetry Erlingsson et al. (2014) and for learning trending behaviors Federighi (2016).

This paper is an update of the conference version Dwork et al. (2006b), with revisions to presentation and terminology. We do *not* aim to survey all the subsequent work on differential privacy. At the end of each section, we briefly discuss how the technical ideas we present have affected subsequent research. We refer the interested reader to several recent monographs and tutorials Dwork and Roth (2014); Vadhan (2016); Hardt et al. (2016); Ligett et al. (2016).

2 Definitions

We are concerned with the interaction between two parties: a database access protocol $\mathcal{M}$, given a static data set $\mathbf{x}$ as input, and an adversary $\mathcal{A}$ who makes queries to and receives answers from $\mathcal{M}$. It is convenient to allow messages to be either finite strings or real-valued vectors. We model the adversary as a computationally unbounded, probabilistic interactive function. Given a database access protocol $\mathcal{M}$, an adversary $\mathcal{A}$, and a particular database $\mathbf{x}$, let the random variable $\mathsf{View}_{\mathcal{M},\mathcal{A}}(\mathbf{x})$ denote the adversary's *view* of their interaction (meaning the list of messages sent back and forth as well as the adversary's internal state and random choices). The randomness in $\mathsf{View}_{\mathcal{M},\mathcal{A}}(\mathbf{x})$ comes from the coins of $\mathcal{M}$ and of $\mathcal{A}$. We will drop either or both of the subscripts $\mathcal{M}$ and $\mathcal{A}$ when the context is clear. Note that for noninteractive schemes, there is no dependence on the adversary $\mathcal{A}$ and we simply write $\mathcal{M}(\mathbf{x})$.

While we have been speaking of a database as a collection of rows, each containing the data of an individual, it can also be useful to think of a database as a histogram describing, for all possible types of individuals, the number of individuals of this type in the database. For example, if an individual is represented by some number d of attributes, the histogram would say, for each possible d-bit string s, the number of individuals in the database with attribute string s. More generally, letting $\mathbb{N} = \{0, 1, \ldots, \}$, a *dataset* is a finite multiset in a domain D, represented as a vector $\mathbf{x} \in \mathbb{N}^{|D|}$, where the ith entry counts the number of occurences in the dataset of the ith element in D (according to some canonical order). We typically consider domains D of the form $\{0,1\}^d$ or $\mathbb{R}^d$. Distance between datasets is measured via the set difference metric $d_{\triangle}(\cdot,\cdot)$, defined as the ℓ_1 distance $\|\mathbf{x} - \mathbf{x}'\|_1$ between the two input multisets (that is, the size of their symmetric difference). Differential privacy requires that similar datasets give rise to similar distributions on outputs. More generally, it imposes a Lipschitz condition on the mapping from datasets to the distribution on outputs.

Given a distance measure on datasets, we say two datasets are *neighbors* (or *adjacent*) if they are at distance 1. Differential privacy requires that neighboring datasets lead to very similar distributions on transcripts. The stringency of the definitions is measured by a positive parameter ε, called the *privacy loss*.

We begin by stating a definition for *noninteractive* mechanisms, which do not take explicit input from an outside analyst, and discuss interactive mechanisms later in this section. Formally, a noninteractive mechanism $\mathcal{M}$ is a randomized function from data sets to some space $\mathcal{O}$ of outputs, namely a map from $\mathbb{N}^D \times \Omega$ to $\mathcal{O}$, where Ω is a probability space. Each dataset $\mathbf{x}$ induces a distribution $\mathcal{M}(\mathbf{x})$ over $\mathcal{O}$.

Definition 2.1. *A noninteractive mechanism $\mathcal{M}$ is* ε-differentially private *(with respect to a given distance measure) if for all neighboring datasets $\mathbf{x}, \mathbf{x}' \in \mathbb{N}^D$, and for all events (measurable sets) S in the space of outputs of $\mathcal{M}$:*

$$\Pr(\mathcal{M}(\mathbf{x}) \in S) \leq e^{\varepsilon} \Pr(\mathcal{M}(\mathbf{x}') \in S)\,. \tag{1}$$

The probabilities are over the coin flips of $\mathcal{M}$.

We say that ε is a bound on the *privacy loss* of $\mathcal{M}$. As we will see when we

describe specific differentially private mechanisms, ε is typcially a parameter: the same mechanism can usually be run with different values of ε.

Remark 2.1. In the preliminary version of this paper, datasets were vectors of known length n with entries in D, with closeness measured via Hamming distance $d_H(\cdot,\cdot)$ over $D^n \times D^n$. The definitions and results in this paper extend to the Hamming metric with only minor changes, though the resulting definitions have subtly different interpretations; see Section 2.2. More generally, one can think of differential privacy as a family of definitions parameterized by a collection of possible datasets and a metric on this collection; see Section 2.2.

The Hamming distance definition is implied by the set difference definition (up to a factor of 2 in the leakage ε). To see why, consider a domain of the form $D' = [n] \times D$ for a fixed value of n. We can embed ordered lists as subsets of this new domain by considering datasets of the form $\mathbf{x} = \{(1, \tilde{x}_1), (2, \tilde{x}_2), \ldots, (n, \tilde{x}_n)\}$ where the $\tilde{x}_i$ all lie in D. Note that for any two such datasets $\mathbf{x}, \mathbf{x}'$ that differ on a single element, the set difference distance between $\mathbf{x}$ and $\mathbf{x}'$ is 2. If $\mathcal{M}$ is ε-differentially private (as in Def. 2.1) then, for all events S, we have $\Pr[\mathcal{M}(\mathbf{x}) \in S) \leq e^{2\varepsilon} \Pr[\mathcal{M}(\mathbf{x}') \in S]$. $\diamondsuit$

Definition 2.1 is unusual for cryptography, in that in most cryptographic settings it is sufficient to require that distributions be statistically close or that they be computationaly indistinguishable. In contrast, Definition 2.1 is much more stringent than statistical closeness: one can have a pair of distributions whose statistical difference is arbitrarily small, yet where the ratio $\frac{\Pr(\mathcal{M}(\mathbf{x}) \in S)}{\Pr(\mathcal{M}(\mathbf{x}') \in S)}$ is infinite (by having a point where one distribution assigns probability zero and the other, non-zero).

Definition 2.1 looks at the worst case over pairs of neighboring databases; the probabilities in the definition only over the random choices of the mechanism $\mathcal{M}$. In particular, *differential privacy requires randomization*. That is, except for constant mechanisms, which ignore their input and output a fixed value, deterministic mechanisms cannot satisfy differential privacy.

For most mechanisms it suffices to quantify over singleton events S. Namely, if the output space $\mathcal{O}$ is discrete, it suffices that $\Pr(\mathcal{M}(\mathbf{x}) = t) \leq e^{\varepsilon} \Pr(\mathcal{M}(\mathbf{x}') = t)$ for all outputs $t \in \mathcal{O}$ in order for (1) to hold (since the probability of an event $E \subseteq \mathcal{O}$ is the sum of the probabilities of its elements). Similarly, if $\mathcal{O}$ is infinite but the distribution $\mathcal{M}(\mathbf{x})$ has a well-defined density $\mathbf{p}_{\mathbf{x}}$ for each $\mathbf{x}$, then it suffices that $\mathbf{p}_{\mathbf{x}}(t) \leq e^{\varepsilon} \mathbf{p}_{\mathbf{x}'}(t)$ for all outputs $t \in \mathcal{O}$.

Differential privacy captures a classical technique used in the social sciences to survey the prevalence of embarassing or illegal practices Warner (1965). Usually described in terms of a few flips of a fair coin, we frame the technique in terms of ε to illustrate its adjustment to accomodate any given bound on the privacy loss.

Example 1 (Randomized response). Consider a survey setting, in which the individuals in the survey are known and the goal is to determine the (approximate) fraction engaging in a specific activity. Thus, for each individual i there is a secret bit $x_i \in \{0, 1\}$. Two such datasets are neighbors if they differ in one entry: That is, for some $1 \leq i \leq n$, the *ith* person does, or does not, engage in the activity, everyone else stays the same.

Consider randomizing each bit independently by flipping it with a certain probability. Specifically, for $b \in \{0,1\}$, let $R(b)$ denote a Bernouilli random variable with $\Pr(R(b) = b) = \frac{e^\varepsilon}{e^\varepsilon+1}$ and $\Pr(R(b) = 1 - bb) = \frac{1}{e^\varepsilon+1}$. The mechanism outputs

$$\mathcal{M}(x_1, ..., x_n) = (R(x_1), ..., R(x_n)).$$

This mechanism is ε-differentially private, since for any two neighboring datasets $\mathbf{x}, \mathbf{x}'$ differing only in the ith entry, and output $y = y_1, ..., y_n$, the ratio $\frac{\Pr(\mathcal{M}(\mathbf{x})=y)}{\Pr(\mathcal{M}(\mathbf{x}')=y)}$ equals $\frac{\Pr(R(x_i)=y_i)}{\Pr(R(x_i')=y_i)}$. By definition of R, this ratio lies in $e^{\pm\varepsilon}$, as desired.

We can use the mechanism to estimate the proportion of 1's and 0's in any subset of individuals. For every $\mathbf{x}$, given a response $y_1, ..., y_n$, the function $g(y) = \sum_i \frac{(e^\varepsilon+1)y_i-1}{e^\varepsilon-1}$ is an unbiased estimator of the number of 1's in $\mathbf{x}$, with standard deviation $\Theta(\sqrt{n}/\varepsilon)$ as $n \to \infty$ and $\varepsilon \to 0$.

This technique has the strength of permitting estimations for the number of 1's in multiple subsets of the x_j's with no further privacy loss. $\diamondsuit$

Definition 2.2 (Laplace Distribution)**.** *The Laplace distribution* $\mathrm{Lap}(\lambda)$ *has density function* $h(y) = \frac{1}{2\lambda}\exp(-|y|/\lambda)$*, mean 0, and standard deviation* $\sqrt{2}\lambda$*.*

Example 2 (Laplace Noise)*.* Suppose that the domain D is $\{0,1\}$ (so each person's data is a single bit), and again the analyst wants to learn $f(\mathbf{x}) = \mathbf{x}(1)$, the total number of 1's in the database. Here we are using the histogram representation of the dataset, and adjacent datasets $\mathbf{x}, \mathbf{x}'$ satisfy $||\mathbf{x} - \mathbf{x}'||_1 = 1$.

Consider the mechanism that computes the true answer $f(\mathbf{x})$ and then adds noise drawn from the Laplace distribution with parameter $1/\varepsilon$:

$$\mathcal{M}(\mathbf{x}) = f(\mathbf{x}) + Y, \quad \text{where } Y \sim \mathrm{Lap}(1/\varepsilon).$$

This mechanism, which adds noise of magnitude roughly $\frac{1}{\varepsilon}$, independent of n, is ε-differentially private. To see why, note that for any real numbers y, y', we have $\frac{h(y)}{h(y')} \leq e^{|y-y'|/\lambda}$. For any two databases $\mathbf{x}$ and $\mathbf{x}'$ which differ in a single entry, the sums $f(\mathbf{x})$ and $f(\mathbf{x}')$ differ by one. Thus, for every $t \in \mathbb{R}$, the ratio of the densities of the distribution of $\mathcal{M}$ on inputs $\mathbf{x}$ and $\mathbf{x}'$ is $\frac{\mathbf{p}_\mathbf{x}(t)}{\mathbf{p}_{\mathbf{x}'}t)} = \frac{h(t-f(\mathbf{x}))}{h(t-f(\mathbf{x}'))}$, which is at most $e^{\varepsilon|f(\mathbf{x})-f(\mathbf{x}')|} \leq e^\varepsilon$, as desired. The same mechanism can be used to release a close approximation to the total number of entries in the dataset $\mathbf{x}(0) + \mathbf{x}(1)$.

The standard deviation of this estimator is roughly a factor of $\sqrt{n}$ smaller than that of the previous mechanism from Example 1. $\diamondsuit$

Taken together, these examples illustrate that, for a given goal, there may be several differentially private algorithms with the same privacy parameter and different levels of success achieving the goal. For a given value of ε, the definition specifies a set of acceptable algorithms or processes.

Non-negligible Leakage and the Choice of Distance Measure. Differential privacy can be viewed as requiring that the probability distributions of $\mathcal{M}(\mathbf{x})$ and $\mathcal{M}(\mathbf{x}')$ be close in the multiplicative metric on probability distributions. For two random probability distributions p and q defined on the same σ-algebra of events, the distance is given by

$$\sup_{\text{events } S} \left| \ln \left(\frac{p(S)}{q(S)} \right) \right| = \inf \left\{ \varepsilon : \text{ for all events } S,\ p(S) \leq e^{\varepsilon} q(S) \text{ and } q(S) \leq e^{\varepsilon} p(S) \right\}.$$

Note that, while we talk of "multiplicative" distance, since we are concerned with the ratio of probabilities of a given event under two neighboring datasets, our formal measure is the logarithm of this ratio. This is done for ease of use; for example, so that we can apply the triangle inequality. Given random variables A and B, we write $A \approx_{\varepsilon} B$ to denote that the distributions of A and B are within multiplicative distance at most ε.

In Example 2 above it is clear that to get any reasonable approximation to $f(\mathbf{x})$, we must have ε at least as large as $1/\|\mathbf{x}\|$, where $\|\mathbf{x}\|$ denotes the number of entries in $\mathbf{x}$. This value of ε is large for cryptography, where the usual requirement is for the leakage, or security parameter (what we call here the privacy loss), to drop faster than any polynomial in the lengths of the inputs. However, non-negligible privacy loss is necessary for statistical utility: If the distance ε between the distributions induced by close databases is much less than $1/\|\mathbf{x}\|$, then the distance between the distributions induced by *any* two databases of size $\|\mathbf{x}\|$ is close to zero (at most $\varepsilon\|\mathbf{x}\|$) and so *no* statistic about the database can be usefully approximated.

The necessity of nonnegligible leakage helps to explain the choice of multiplicative distance measure used in Definition 2.1. The following example illustrates why more standard, "average-case", distance measures such as total variation distance do not yield meaningful guarantees when $\varepsilon \geq 1/\|\mathbf{x}\|$.

Example 3. Consider a setting where each entry in a dataset includes some identifying information that makes it unique (say, name or social security number). Look at the candidate sanitization $\mathcal{M}$ which, on input a set $\mathbf{x}$, independently outputs each element with probability ε (that is, it outputs a subsample of $\mathbf{x}$ where each entry appears with probability ε). If $\mathbf{x}$ and $\mathbf{x}'$ differ in a single position, the statistical difference between $\mathcal{M}(\mathbf{x})$ and $\mathcal{M}(\mathbf{x}')$ is ε, since the probability that the one element in the symmetric difference $\mathbf{x} \triangle \mathbf{x}'$ appears in the output is ε (and the transcript distributions are identical otherwise). Nevertheless, it is clear that such a mechanism reveals private information about some subset (about an ε fraction) of the individuals in the dataset. When $\varepsilon \geq 1/\|\mathbf{x}\|$, then the expected size of the set is at least 1.

This example mechanism does *not* satisfy differential privacy for any finite value of ε, since if $\mathbf{x}$ and $\mathbf{x}'$ differ by the addition of one entry x' in $\mathbf{x}' \setminus \mathbf{x}$, then the set of outputs that include x' has probability zero when the database is $\mathbf{x}$, and probability ε when the database in $\mathbf{x}'$. With this mechanism, when n is large, the probability that any specific individual's privacy is compromised is small; but we find unpalatable a "lottery" philosophy that sacrifices a small number of individuals on every invocation. $\diamondsuit$

Interactive Mechanisms: Quantifying Over Adversaries. In many cases, it makes sense to consider an interactive mechanism, which responds to queries from a potentially adversarial analyst. In that case, the mechanism consists of a series of randomized functions, one for each round, mapping a set of allowable queries to a set of outputs. The adversary's *view* of the interaction with the mechanism, denoted $\mathsf{View}_{\mathcal{M},\mathcal{A}}(\mathbf{x})$, can then be described by the *transcript*—the sequence of messages exchanged between mechanism and adversary—together with the randomness of the adversary. For a given dataset $\mathbf{x}$ and adversary $\mathcal{A}$, the adversary's view is a random variable $\mathsf{View}_{\mathcal{M},\mathcal{A}}(\mathbf{x})$.

Definition 2.3. *An interactive mechanism $\mathcal{M}$ is* ε-differentially private *(with respect to a given distance measure) if for all neighboring datasets $\mathbf{x}, \mathbf{x}' \in D^*$, for all adversaries $\mathcal{A}$, and for all events (measurable sets) S in the space of views:*

$$\Pr(\mathsf{View}_{\mathcal{M},\mathcal{A}}(\mathbf{x}) \in S) \leq e^{\varepsilon} \Pr(\mathsf{View}_{\mathcal{M},\mathcal{A}}(\mathbf{x}') \in S)\,. \tag{2}$$

As we will see in the next section, this more complicated definition is usually not needed, since we can design interactive mechansism by ensuring that ear round (viewed as a noninteractive mechanism) is differentially private.

2.1 Fundamental Properties

In this section we discuss some fundamental properties that follow from the definition of differential privacy. All differentially private algorithms enjoy these properties. Throughout the section, we use the notation for noninteractive mechanisms since it is easier to read; the discussion applies equally well to the interactive case.

Immunity to Auxiliary Information. Differential privacy makes no reference to an input distribution, and it makes sense regardless of what other information the adversary has—or will have in the future—about the data set. We discuss this idea further in Section 2.2.

Postprocessing. Anything derived from the output of a differentially private algorithm is itself differentially private, and the derivation incurs no further privacy loss.

Proposition 2.4 (Closure under postprocessing)**.** *Let g be a randomized function. If $\mathcal{M}$ is an ε-differentially private mechanism, then the mechanism $\mathcal{M}' = g \circ \mathcal{M}$ is ε-differentially private.*

Proof. Fix an event S in the output space of g and neighboring data sets $\mathbf{x}, \mathbf{x}'$. Let R be a random variable denoting the randomness used by g, and write $g(z) = \tilde{g}(z; R)$ where $\tilde{g}$ is deterministic. Then $\Pr(g(\mathcal{M}(\mathbf{x})) \in S) = \Pr((\mathcal{M}(\mathbf{x}), R) \in \tilde{g}^{-1}(S))$. For each possible value r of R, let $S_r = \{s : (s, r) \in \tilde{g}^{-1}(S)\}$. Since $\mathcal{M}$ is ε-differentially private, for each value r we have $\Pr(\mathcal{M}(\mathbf{x}) \in S_r) \leq e^{\varepsilon} \Pr(\mathcal{M}(\mathbf{x}') \in S_r)$, hence $\Pr(\tilde{g}(\mathcal{M}(\mathbf{x}), R) \in S | R = r) \leq e^{\varepsilon} \Pr(\tilde{g}(\mathcal{M}(\mathbf{x}'), R) \in S | R = r)$. We conclude that $\Pr(\tilde{g}(\mathcal{M}(\mathbf{x}), R) \in S) \leq e^{\varepsilon} \Pr(\tilde{g}(\mathcal{M}(\mathbf{x}'), R) \in S)$, as desired. □

Composition. Differentially private algorithms *compose*, in the sense that when several differentially private algorithms are run "independently" (that is, using independent randomness), then the joint output of all the algorithms is still differentially private. The simplest form of this is the following:

Lemma 2.5 (Simple Composition)**.** *Let $\mathcal{M}_1, ..., \mathcal{M}_k$ be a fixed sequence of mechanisms, where $\mathcal{M}_i$ is ε_i-differentially private. Then their joint output, given by $\mathcal{M}(\mathbf{x}) = (\mathcal{M}_1(\mathbf{x}), \ldots, \mathcal{M}_k(\mathbf{x}))$, is ε-differentially private for $\varepsilon = \sum_{i=1}^{k} \varepsilon_i$.*

In fact, the lemma holds even when the mechanisms run on the data maybe specified adaptively, based on the outputs of previous mechanisms. To formalize the adaptive model, imagine an interaction between an analyst/adversary $\mathcal{A}$ and a curator $\mathcal{M}$ holding a dataset $\mathbf{x}$. At each round i from 1 to k, where k is finite but need not be fixed in advance, the analyst $\mathcal{A}$ specifies a positive number ε_i and a mechanism $\mathcal{M}_i$ that is ε_i-differentially private. The curator computes $a_i \leftarrow \mathcal{M}_i(\mathbf{x})$ and sends a_i to $\mathcal{A}$. The transcript of the interaction is the sequence of triples $\{(\varepsilon_i, \mathcal{M}_i, a_i)\}_{i=1,\ldots,k}$.

Lemma 2.6 (Composition)**.** *Suppose that there is an $\varepsilon > 0$ such that the curator $\mathcal{M}$ stops answering queries in the first round j where $\sum_{i=1}^{j} \varepsilon_i > \varepsilon$, thereby ensuring that $\sum_i \varepsilon_i \leq \varepsilon$ for the rounds at which queries were answered. Then $\mathcal{M}$ is ε-differentially private.*

Before proving the lemma, we note that composition has (at least) two important consequences. First, differentially private algorithms may be *constructed modularly*. For example, if we wish to design an interactive protocol, then it suffices that each round of the protocol be differentially private. The computation done at a given round may depend arbitrarily on the outputs of past rounds. Similarly, we may design an iterative algorithm that proceeds in stages, where outputs from early stages determine the computations performed in later ones. We give examples of such iterative mechanisms in Section 3.

Second, if differentially private mechanisms are run separately on different datasets, then the joint output is still differentially private even for individuals whose data appear in several of the data sets. For example, suppose that two hospitals which serve overlapping populations independently run ε-differentially private algorithms $\mathcal{M}_1$ and $\mathcal{M}_2$ on their datasets. We may view the data collected by the two hospitals as a single large dataset $\mathbf{x}$ (where each individual's entry contains data from one or both hospitals), and view $\mathcal{M}_1$ and $\mathcal{M}_2$ as each operating on different parts of $\mathbf{x}$. The composition lemma then implies that the joint output of $\mathcal{M}_1$ and $\mathcal{M}_2$ is 2ε-differentially private. Many natural approaches to defining privacy do *not* yield composable definitions.

Proof. For simplicity we will assume that all the mechanisms $\mathcal{M}_i$ have the same output space $\mathcal{O}$. [2] Also for simplicity, assume that $\mathcal{O}$ is discrete; the general case follows by a similar argument. Consider the case $k = 2$. Fix a pair of neighboring datasets $\mathbf{x}$ and

[2] This is without loss of generality, as otherwise we can define $\mathcal{O} = \cup_i \mathcal{O}_i$.

$\mathbf{x}'$, an analyst $\mathcal{A}$, and random coins r for the analyst (or, more simply, we can consider a deterministic analyst). Let $\mathcal{M}_1$ denote the first mechanism selected by $\mathcal{A}$ the analyst. The second mechanism depends on the output of $\mathcal{M}_1$; let $\mathcal{M}_2^{(a)}$ denote the mechanism selected in the second stage when a is the output of $\mathcal{M}_1$, and let $\varepsilon_2^{(a)}$ denote the privacy loss of $\mathcal{M}_2^{(a)}$. Consider an event $S \subseteq \mathcal{O} \times \mathcal{O}$. For each element a of $\mathcal{O}$, define the set $S_a = \{s : (a, s) \in S\}$. We can write

$$\Pr(\mathcal{M}(\mathbf{x}) \in S) = \sum_{a \in \mathcal{O}} \Pr(\mathcal{M}_1(\mathbf{x}) = a) \Pr(\mathcal{M}_2^{(a)}(\mathbf{x}) \in S_a).$$

By the differential privacy of $\mathcal{M}_1$ and $\mathcal{M}_2^{(a)}$, we get $\Pr(\mathcal{M}_1(\mathbf{x}) = a) \leq e^{\varepsilon_1} \Pr(\mathcal{M}_1(\mathbf{x}') = a)$ and $\Pr(\mathcal{M}_2^{(a)}(\mathbf{x}) \in S_a) \leq e^{\varepsilon_2^{(a)}} \Pr(\mathcal{M}_2^{(a)}(\mathbf{x}') \in S_a)$. Since $\varepsilon_1 + \varepsilon_2^{(a)} \leq \varepsilon$ for all $a \in \mathcal{O}$ (by the constraint imposed by the curator), we have $\Pr(\mathcal{M}(\mathbf{x}) \in S) \leq e^{\varepsilon} \Pr(\mathcal{M}(\mathbf{x}') \in S)$, as desired.

The argument extends to all finite $k > 2$ by induction. □

Group Privacy. Differential privacy with respect to changes of an individual's data implies differential privacy with respect to changes in the data of small sets of individuals.

Definition 2.7. *A mechanism is (k, ε)-differentially private if for all pairs $\mathbf{x}, \mathbf{x}'$ which differ in at most k entries (that is, $d_{\triangle}(\mathbf{x}, \mathbf{x}') \leq k$), for all adversaries $\mathcal{A}$ and for all events S in the output space,* $\Pr(\mathcal{M}(\mathbf{x}) \in S) \leq e^{\varepsilon} \Pr(\mathcal{M}(\mathbf{x}') \in S)$.

In this case we may say the mechanism is *ε-differentially private for groups of size k*.

Lemma 2.8. *Every $(1, \frac{\varepsilon}{k})$-differentially private mechanism is also (k, ε)-differentially private.*

Proof. Consider a chain of at most k databases connecting $\mathbf{x}$ to $\mathbf{x}'$, where only one entry changes at each step. The probability of any event changes by a factor of $\exp(\pm\varepsilon/k)$ at each step, so $\frac{\Pr(\mathcal{M}(\mathbf{x}) \in S)}{\Pr(\mathcal{M}(\mathbf{x}') \in S)} \in \exp(\pm\varepsilon/k)^k = \exp(\pm\varepsilon)$. □

2.2 Bibliographic Notes and Discussion

In the initial version of this paper, differential privacy was called *indistinguishability*. The name "differential privacy" was suggested by Michael Schroeder, and was first used in Dwork (2006). The initial version of this paper used the Hamming metric on data sets; we adopt the more general set difference metric here.

The composition and postprocessing properties stated in Section 2.1 did not appear in the initial version, although they were known at the time and have since become folklore. The failures to compose suffered by other approaches to defining privacy are discussed by Ganta et al. (2008). Some relaxations of differential privacy are known to satisfy even stronger composition properties (Dwork et al. (2010); Dwork and Rothblum (2016); Bun and Steinke (2016)).

Interpreting Differential Privacy

In Appendix 4.4, we discuss some alternate formulations of differential privacy in terms of simulation, and the change to an adversary's prior distribution about an individual.

Subsequent to the original version of this paper, significant research considered different interpretations of differential privacy. The definition of semantic privacy given in this paper provides guarantees on how a Bayesian adversary's posterior distribution compares to its prior, under assumptions on the form of the adversary's prior. Such assumptions are known to be necessary (Dwork, 2006; Dwork and Naor, 2010; Kifer and Machanavajjhala, 2011). In contrast, one may also formulate definitions by comparing the adversary's posterior distributions in different settings (say, assuming that someone's data was or was not used in the computation) Kasiviswanathan and Smith (2008); Bassily et al. (2013).

McSherry and Talwar (2007) provided a game-theoretic interpretation of differential privacy. Specifically, suppose that individual i has preferences over the possible outcomes $\mathcal{O}$ of the mechanism $\mathcal{Z}$, and that these preferences are expressed via a nonnegative utility function $u_i : \mathcal{O} \to \mathbb{R}_{\geq 0}$. Then, for any two neighboring databases $\mathbf{x}, \mathbf{x}'$ differing in the data of the ith individual, the expected utility experienced by i will be essentially the same regardless of which dataset is used: $\mathbb{E}(u_i(\mathcal{M}(\mathbf{x}))) \leq e^{\varepsilon}\mathbb{E}(u_i(\mathcal{M}(\mathbf{x}')))$ and $\mathbb{E}(u_i(\mathcal{M}(\mathbf{x}))) \geq e^{-\varepsilon}\mathbb{E}(u_i(\mathcal{M}(\mathbf{x}')))$. This has implications for (approximate) truthfulness, since there is litte incentive to misreport one's data.

Finally, Wasserman and Zhou (2010) provided an interpretation of differential privacy, based on hypothesis-testing, that is close to our notion of semantic privacy. Specifically, fix an ε-differentially private mechanism $\mathcal{M}$, an i.i.d. distribution on the data $\mathbf{x}$, an index i, and disjoint sets S and T of possible values for the i-th entry x_i of $\mathbf{x}$. Then any hypothesis test (given $\mathcal{M}(\mathbf{x})$, and full knowledge of the input product distribution on $\mathbf{x}$ and the differentially private mechanism $\mathcal{M}$) for the hypothesis $H_1 : x_i \in S$ versus the alternative $H_1 : x_i \in T$ must satisfy

$$1 - \beta \leq e^{\varepsilon}\alpha\,, \tag{3}$$

where α is the significance level (maximum type-I error, or "false positive") and $1 - \beta$ is the power (β is the maximum type-II error, or "false negative") of the test. In other words, the test rejects the hypothesis with approximately the same probability regardless of whether the hypothesis is true.

Generalizations and Variants

The mostly widely used variant of differential privacy is (ε, δ)-differential privacy (first stated in Dwork et al. (2006a) and closely related to the definition used in Dinur and Nissim (2003); Dwork and Nissim (2004); Blum et al. (2005)); this definition adds an additive approximation ($+\delta$) to the inequality in Equation (1). When δ is sufficiently small (in particular, much smaller than $1/n$), the definition provides similar semantics to ε-differential privacy Kasiviswanathan and Smith (2008).

Several other variants and generalizations of differential privacy have appeared since the original version of this paper. Among other features, these variants seek to incorporate specific types of data (such as graphs, where each edge may depend on more than one person's information), known dependencies among data records, adversarial uncertainty, computational considerations, and optimization for a high degree of composition.

3 Sensitivity and Privacy

We now present the most basic tool for constructing differentially private mechanisms. We formally define sensitivity of functions and then prove that choosing noise distributed according to the Laplace distribution with expected magnitude $\frac{S(f)}{\varepsilon}$ ensures ε-differential privacy when the query function f has sensitivity $S(f)$. We extend the analysis to vector-valued functions f, and even to adaptively chosen series of query functions. Intuitively, this analysis shows how a "privacy budget" ε can be spent by a sequence of queries.

Definition 3.1 (L_1 Sensitivity)**.** *The L_1 sensitivity of a function $f : D^* \to \mathbb{R}^d$ is the smallest number $S(f)$ such that for all neighboring datasets $\mathbf{x}, \mathbf{x}'$,*

$$\|f(\mathbf{x}) - f(\mathbf{x}')\|_1 \leq S(f) \ .$$

Recall that neighboring datasets are defined to be multisets whose symmetric difference is a singleton. This yields $S(f) = \sup_{\mathbf{x},\mathbf{x}':d_\triangle(\mathbf{x},\mathbf{x}')=1} \|f(\mathbf{x}) - f(\mathbf{x}')\|_1$. A simple application of the triangle inequality shows that sensitivity is a Lipschitz condition on f: for all pairs of data sets $\mathbf{x}, \mathbf{x}' \in D^*$:

$$\frac{\|f(\mathbf{x}) - f(\mathbf{x}')\|_1}{d_\triangle(\mathbf{x}, \mathbf{x}')} \leq S(f).$$

We note that the neighboring relation is often defined in terms of the Hamming distance (instead of size of symmetric difference) yielding a slightly different notion of sensitivity. All our results can be easily modified to accomodate this variant. Furthermore, one can define sensitivity with respect to any metric on the output space; see Section 3.3.

Example 4 (Sums, Counts and Histograms). Consider $\mathrm{SUM}_v(\mathbf{x}) = \sum_{i=1}^{|\mathbf{x}|} v(x_i)$ where $v : D \to \mathbb{R}^d$ and assume a bound $\|v(x)\|_1 \leq \gamma$ for all $x \in D$, so that $S(\mathrm{SUM}_v) \leq \gamma$. As a special case, taking $v(x) = 1$ for all $x \in D$, we obtain the function computing the number of individuals in $\mathbf{x}$, $\mathrm{COUNT}(\mathbf{x}) = |\mathbf{x}| = \sum_{i=1}^{|\mathbf{x}|} v(x_i) = \sum_{i=1}^{|\mathbf{x}|} 1$, and hence $S(\mathrm{COUNT}) = 1$.

Now consider an arbitrary partition of the domain D into d disjoint regions $B_1, \ldots, B_d$. The histogram function $\mathrm{HIST} : \mathbb{N}^{|D|} \to \mathbb{Z}^d$ counts the number of dataset points which fall into each of the bins $\mathrm{HIST}(\mathbf{x}) = (|\mathbf{x} \cap B_1|, \ldots, |\mathbf{x} \cap B_d|)$. We may capture this as $\mathrm{HIST}(\mathbf{x}) = \sum_{i=1}^{|\mathbf{x}|} v(x_i)$, where, the function v maps each data item to a vector of

dimension d containing $d-1$ zeroes and a single one: $v(x) = (0, 0, ..., 1, ...0)$. Note that $||v(x)||_1 = 1$ for all $x \in D$ and so $S(\text{HIST}) = 1$. Indeed, neighboring databases have histograms that differ in exactly one of the counts—one bin loses or gains a point. $\diamondsuit$

3.1 Calibrating Noise According to $S_{L_1}(f)$

Definition 3.2 (Laplace Probability Distribution)**.** *The Laplace distribution with parameter λ, denoted* $\text{Lap}(\lambda)$*, is the distribution on $\mathbb{R}$ with probability density function*

$$\mathbf{p}(y) = \frac{1}{2\lambda} e^{-\lambda|x|}.$$

The distribution has mean 0 and standard deviation $\lambda\sqrt{2}$. The d-dimension Laplace distribution on $\mathbb{R}^d$, denoted $\text{Lap}(\lambda)^d$*, is a product of d Laplace distributions; that is, the coordinates are i.i.d.* $\text{Lap}(\lambda)$*.*

Observe that for Y drawn from the Laplace distribution, $\mathbf{p}(y)/\mathbf{p}(y') \leq e^{|y-y'|/\lambda}$. Similarly, for Y drawn from the d-dimensional Laplace distribution, the density function at y is proportional to $\exp(-\lambda\|y\|_1)$, and so for any points $y, y' \in \mathbb{R}^d$, we have:

$$\frac{\mathbf{p}(y)}{\mathbf{p}(y')} \leq \exp(\lambda\|y - y'\|_1)\,. \tag{4}$$

It follows that to release a (perturbed) value $f(\mathbf{x})$ while satisfying differential privacy, it suffices to add Laplace noise with $\lambda = S(f)/\varepsilon$ in each coordinate:

Proposition 3.3 (Laplace Mechanism—Nonadaptive Version)**.** *For all $f : D^* \to \mathbb{R}^d$, the following mechanism is ε-differentially private:*

$$\mathcal{M}_f(\mathbf{x}) = f(\mathbf{x}) + (Y_1, \ldots, Y_d) \;\; \textit{where the } Y_i \textit{ are drawn i.i.d. from } \text{Lap}(S(f)/\varepsilon).$$

Proof. We introduce some notation. Consider the random variable corresponding to the outcome of mechanism $\mathcal{M}_f$ when executed with input $\mathbf{x}$. We use $\mathbf{p}[\mathcal{M}(\mathbf{x}) = z]$ to denote the probability density of this variable at point z. Let $\mathbf{x}, \mathbf{x}'$ be neighboring databases. From Equation (4) we have that

$$\frac{\mathbf{p}[\mathcal{M}_f(\mathbf{x}) = z]}{\mathbf{p}[\mathcal{M}_f(\mathbf{x}') = z]} \leq \exp\left(\frac{\varepsilon \cdot \|f(\mathbf{x}) - f(\mathbf{x}')\|_1}{S_{L_1}(f)}\right) \leq \exp(\varepsilon). \tag{5}$$

Let $Z \subset \mathbb{R}$ be a measurable set. Using Eqn. (5) we get

$$\begin{aligned}
\Pr[\mathcal{M}_f(\mathbf{x}) \in Z] &= \int_{z \in Z} \mathbf{p}[\mathcal{M}_f(\mathbf{x}) = z]\mathrm{d}z \\
&\leq \exp(\varepsilon) \int_{z \in Z} \mathbf{p}[\mathcal{M}_f(\mathbf{x}') = z]\mathrm{d}z \\
&= \exp(\varepsilon) \cdot \Pr(\mathcal{M}_f(\mathbf{x}') \in Z).
\end{aligned}$$

□

Dealing with Adaptivity

Proposition 3.3 is a special case of the privacy of a more general, possibly adaptive, interactive process. Adaptivity complicates the nature of the "query function", which is no longer predetermined, but rather a strategy for producing queries based on answers given thus far. For example, an adaptive histogram query might ask to refine those regions with a substantial number of respondents, and we would expect the set of such selected regions to depend on the random noise incorporated into the initial responses.

Consider a querying strategy $\mathcal{A}$ issuing a sequence of k adaptive queries to $\mathcal{M}$. The i-th query $f_i : \mathbb{N}^{|D|} \to \mathbb{R}$ is a function that maps data sets to reals; let a_i denote the answer sent by $\mathcal{M}$ back to $\mathcal{A}$. Note that f_i is determined by $a_1, \ldots, a_{i-1}$ and the random coins r tossed by $\mathcal{A}$. The *view* of $\mathcal{A}$ in the interaction can be summarized by the vector $(r, a_1, a_2, \ldots, a_k)$—we can omit the f_i since they are determined by r and the a_j, for $j < i$. For any particular view v, consider the function $f^v : \mathbb{N}^D \to \mathbb{R}^k$ whose ith coordinate f_i^v is the i-th query in the interaction described by v. We use the superscript v to make the dependency on the transcript explicit.

Consider a mechanism $\mathcal{M}$ that, upon receiving the i-th query, either (a) answers $a_i = f_i^v(\mathbf{x}) + \mathrm{Lap}(\lambda)$ for some fixed λ, or (b) refuses to answer ($a_i = \perp$). We can bound the privacy loss in this interaction in terms of the L_1 sensitivity of the joint function f^v, yielding a more nuanced version of composition than given in Lemma 2.6. The server limits the queries by refusing to answer once the sensitivity $S(f_1^v, \ldots, f_i^v)$ is above a certain threshold T. Note that the decision whether or not to respond is based solely on $S(f_1^v, \ldots, f_i^v)$, *independent* of the data set $\mathbf{x}$. This decision is not disclosive, since it can be computed by the user submitting the queries.

Theorem 3.4 (Laplace Mechanism—Adaptive Version)**.** *Consider an arbitrary adaptive query strategy and let $f^v(\mathbf{x}) : \mathbb{N}^{|D|} \to \mathbb{R}^k$ be its query function as parameterized by a view $v = (r, a_1, a_2, \ldots, a_k)$. The mechanism above, which answers $a_i = f_i^v(\mathbf{x}) + \mathrm{Lap}(T/\varepsilon)$ as long as $S(f^v) \leq T$, is ε-differentially private.*

Proof. Let $\mathbf{x}, \mathbf{x}'$ be neighboring data sets. Using the law of conditional probability,

$$\frac{\mathbf{p}[(r, \mathcal{M}(\mathbf{x})) = (r, a_1, a_2, \ldots, a_k)]}{\mathbf{p}[(r, \mathcal{M}(\mathbf{x}')) = (r, a_1, a_2, \ldots, a_k)]} = \prod_{i \leq k} \frac{\mathbf{p}[\mathcal{M}(\mathbf{x})_i = a_i | r, a_1, \ldots, a_{i-1}]}{\mathbf{p}[\mathcal{M}(\mathbf{x}')_i = a_i | r, a_1, \ldots, a_{i-1}]}.$$

For each term in the product, fixing r and the first $i-1$ answers $a_1, \ldots, a_{i-1}$ fixes the values of $f_i^v(\mathbf{x})$ and $f_i^v(\mathbf{x}')$. As such, the conditional distributions on a_i are either $\mathrm{Lap}(\lambda)$, with $\lambda = T/\varepsilon$, or the constant $\perp$. We can bound the product as

$$\begin{aligned} \prod_{i \leq k} \frac{\mathbf{p}[\mathcal{M}(\mathbf{x})_i = a_i | r, a_1, \ldots, a_{i-1}]}{\mathbf{p}[\mathcal{M}(\mathbf{x}')_i = a_i | r, a_1, \ldots, a_{i-1}]} &\leq \prod_{i \leq k} \exp(|f_i^v(\mathbf{x}) - f_i^v(\mathbf{x}')|/\lambda) \\ &= \exp(\|f^v(\mathbf{x}) - f^v(\mathbf{x}')\|_1/\lambda) \\ &\leq \exp(S_{L_1}(f^v)/\lambda) \leq \exp(\varepsilon), \end{aligned}$$

where the last inequality follows from our setting of λ. □

3.2 Some Useful Insensitive Functions

We describe specific functionalities which have low sensitivity, and which consequently can be released with little added noise using the protocols of the previous section. In many cases, the sensitivity-based approach presented here permits the introduction of far less distortion than suggested by previous frameworks Blum et al. (2005), while ensuring a cleaner notion of privacy.

Disjoint Analyses. There are many types of analyses that first partition the input space into disjoint regions and then examine each region separately. One very simple example of such an analysis is a histogram, which simply counts the number of elements that fall into each region. Imagining that D is subdivided into d disjoint regions and that $f : D^* \to \mathbb{Z}^d$ is the function that counts the number of elements in each region, we saw in Example 4 that $S(f) = 1$, and so we can apply the Laplace mechanism with $\lambda = 1/\varepsilon$ to each count independently. In contrast, in the framework of Blum et al. (2005), the noise has expected magnitude $\Theta(\sqrt{d}/\varepsilon)$. This $\Theta(\sqrt{d})$ factor can be significant in applications where the number of regions d exceeds the number n of data points—which is often the case with contingency tables.

More generally, letting D be partitioned into d disjoint regions $B_1, \ldots, B_d$, let $f = (f_1, \ldots, f_d) : D^* \to \mathbb{R}^d$ be a function whose i-th output coordinate $f(\mathbf{x})_i$ depends only on those elements in the i-th region, i.e., $\mathbf{x} \cap B_i$. Let $\mathbf{x}'$ be a neighbor of $\mathbf{x}$ resulting from the addition or removal of an entry and observe that $f(\mathbf{x})_i \neq f(\mathbf{x}')_i$ for at most one $i \in [d]$. We hence get that $S(f) \leq \max_i S(f_i)$. Again, and importantly, the bound is independent of the output dimension d.

Distance from a Property. A property is a subset S of D^*. The distance $f_S(\mathbf{x})$ between a particular dataset $\mathbf{x}$ and S is the cardinality of the symmetric difference between $\mathbf{x}$ and its nearest point $\mathbf{x}'$ in S (i.e., the minimum number of changes in terms of removing and inserting elements to $\mathbf{x}$ that lead to an element of S). For example, we might ask how close to well-clustered a dataset is, meaning, how many data points need to be changed for a clustering of at least given quality to exist. For any set $S \subseteq D^*$, $f_S(\mathbf{x})$ has sensitivity (at most) 1.

Mean and Covariance. One very common analysis applied to datasets is estimating the mean and covariance of attributes of the data. Let $v : D \to \mathbb{R}^d$ be some function mapping rows in the database to column vectors in $\mathbb{R}^d$, and assume an upper bound $\gamma = \max_{x \in D} \|v(x)\|_1$. The mean vector μ and covariance matrix C are defined as

follows:

$$\begin{aligned} \mu(\mathbf{x}) &= \operatorname*{avg}_i v(x_i) = \frac{1}{|\mathbf{x}|}\sum_{i=1}^{|\mathbf{x}|} v(x_i) \\ \text{and } C(\mathbf{x}) &= \operatorname*{avg}_i v(x_i)v(x_i)^T - \mu\mu^T = \frac{1}{|\mathbf{x}|}\sum_{i=1}^{|\mathbf{x}|} v(x_i)v(x_i)^T - \mu\mu^T. \end{aligned}$$

The mean can be simply estimated as $\mu(\mathbf{x}) = \mathrm{SUM}_v(\mathbf{x})/\mathrm{COUNT}(\mathbf{x})$ where $\mathrm{COUNT}(\mathbf{x})$ and $\mathrm{SUM}_v(\mathbf{x})$ are from Example 4. The covariance matrix can be treated similarly: viewing the $d \times d$ matrice $v(x)v(x)^T$ as a d^2 dimensional vector, $\operatorname{avg}_i v(x_i)v(x_i)^T$ can be estimated as $\mathrm{SUM}_{vv^T}(\mathbf{x})/\mathrm{COUNT}(\mathbf{x})$, where $\mathrm{SUM}_{vv^T}(\mathbf{x}) = \sum_{i=1}^{|\mathbf{x}|} v(x_i)v(x_i)^T$. Observe that $\|v(x)v(x)^T\|_1 \le \gamma^2$ and hence $S(\mathrm{SUM}_{vv^T}) \le \gamma^2$. We can therefore construct the following 3ε-differentially private mechanism for estimating mean and covariance:

$$\begin{aligned} \mathcal{M}(\mathbf{x}): \quad & \text{Let } \tilde{n} = \mathrm{COUNT}(\mathbf{x}) + \mathrm{Lap}(1/\varepsilon). \\ & \text{Let } \tilde{s} = \mathrm{SUM}_v(\mathbf{x}) + \mathrm{Lap}(\gamma/\varepsilon)^d. \\ & \text{Let } \tilde{c} = \mathrm{SUM}_{vv^T}(\mathbf{x}) + \mathrm{Lap}(\gamma^2/\varepsilon)^{d^2}. \\ & \text{Output } \tilde{\mu} = \tilde{s}/\tilde{n}. \\ & \text{Output } \tilde{C} = \tilde{c}/\tilde{n} - \tilde{s}\tilde{s}^T/\tilde{n}^2. \end{aligned}$$

The mechanism is 3ε-differentially private by the composition property (Lemma 2.6).

In the framework of Blum et al. (2005), the noise added to each query is proportional to the square root of the number of queries. For SUM_v this gives noise proportional to $\gamma\sqrt{d}$ in each of its d coordinates and for SUM_{vv^T} this gives noise proportional to $\gamma^2 d$ in each of its d^2 coordinates. By treating the coordinates jointly, our mechanism $\mathcal{M}$ hence improves the noise in estimating SUM_v and SUM_{vv^T} by factors of $\sqrt{d}$ and d respectively, while also providing a stronger privacy guarantee than Blum et al. (2005).

Functions with Low Sample Complexity. Any function f which can be accurately approximated by an algorithm which looks only at a small fraction of the database has low sensitivity, and so the value can be released with relatively little noise. In particular, functions which can be approximated based on a random sample of the data points fit this criterion.

Lemma 3.5. *Let $f : D^* \to \mathbb{R}^d$. Suppose there is a randomized algorithm A such that for all input datasets $\mathbf{x}$: (1) A operates on a subsample $\tilde{\mathbf{x}}$ of $\mathbf{x}$ where each element of $\mathbf{x}$ appears in $\tilde{\mathbf{x}}$ with probability at most α, and (2) $\|A(\mathbf{x}) - f(\mathbf{x})\|_1 \le \sigma$ with probability at least $\beta > \frac{1+\alpha}{2}$. Then $S(f) \le 2\sigma$.*

The lemma translates a property of f related to sample complexity into a combinatorial property related to privacy. It captures many of the low-sensitivity functions described in the preceding sections, although the bounds on sensitivity given by the lemma are often quite loose.

Proof. Let $\mathbf{x} = (x_1, \ldots, x_{|\mathbf{x}|})$. For any $i \in [|\mathbf{x}|]$ denote by $A(\mathbf{x})\big|_{-i}$ the distribution on the outputs of A conditioned on the event that x_i is exluded from the subsample $\tilde{\mathbf{x}}$. By the definition of conditional probability, we get that for all $\mathbf{x}$ the probability that $A(\mathbf{x})\big|_{-i}$ is within distance σ of $f(\mathbf{x})$ is strictly greater than $(\beta - \alpha)/(1 - \alpha) \geq \frac{1}{2}$. Let $\mathbf{x}' = \mathbf{x} \setminus \{x_i\}$. By the union bound, there exists some point p in the support of $A(\mathbf{x})\big|_{-i}$ which is within distance σ of *both* $f(\mathbf{x})$ and $f(\mathbf{x}')$, and hence $\|f(\mathbf{x}) - f(\mathbf{x}')\|_1 \leq \|f(\mathbf{x}) - p\|_1 + \|p - f(\mathbf{x}')\|_1 \leq 2\sigma$. □

One might hope for a converse to Lemma 3.5, but it does not hold; not all functions with low sensitivity can be approximated by an algorithm with low sample complexity. The counterexample is easiest to state for Hamming distance, though it extends directly to set difference: let $D = GF(2^{\lceil \log n \rceil})$ and let $f(\mathbf{x})$ denote the Hamming distance between $\mathbf{x}$ and the nearest codeword in a Reed-Solomon code of dimension $k = n(1 - o(1))$. One cannot learn anything about $f(\mathbf{x})$ if one sees fewer than k positions of the input[3], and yet f has sensitivity 1.

3.3 Sensitivity in General Metric Spaces

The intuition that *in*sensitive functions of a database can be released privately is not specific to the L_1 norm. Indeed, it seems that if removing or adding one entry to $\mathbf{x}$ induces a small change in $f(\mathbf{x})$—under any measure of distance on $f(\mathbf{x})$—then we should be able to release $f(\mathbf{x})$ privately with relatively little noise. We formalize this intuition for a general metric $d_{\mathcal{Z}}$ on the output $f(\mathbf{x})$. We will use symmetry, i.e. $d_{\mathcal{Z}}(x, y) = d_{\mathcal{Z}}(y, x)$, and the triangle inequality: $d_{\mathcal{Z}}(x, y) \leq d_{\mathcal{Z}}(x, z) + d_{\mathcal{Z}}(z, y)$.

Definition 3.6. *Let $\mathcal{Z}$ be a metric space with a distance function $d_{\mathcal{Z}}(\cdot, \cdot)$. The sensitivity $S_{\mathcal{Z}}(f)$ of a function $f : D^* \to \mathcal{Z}$ is the amount that the function value varies when a single entry of the input is changed.*

$$S_{\mathcal{Z}}(f) \stackrel{def}{=} \sup_{\mathbf{x}, \mathbf{x}' :\ d_{\triangle}(\mathbf{x}, \mathbf{x}') = 1} d_{\mathcal{Z}}(f(\mathbf{x}), f(\mathbf{x}'))$$

Given a point $z \in \mathcal{Z}$, (and a measure on $\mathcal{Z}$) we can attempt to define a probability density function

$$\mathbf{p}_{z,\varepsilon}(y) \propto \exp\left(\frac{\varepsilon \cdot d_{\mathcal{Z}}(y, z)}{2 \cdot S_{\mathcal{Z}}(f)}\right).$$

There may not always exist such a density function, since the right-hand expression may not integrate to a finite quantity. However, if it is finite then the distribution given by $\mathbf{p}_{z,\varepsilon}(\cdot)$ is well-defined.

[3]This follows from the fact that a random Reed-Solomon codeword can be thought of as a Shamir secret sharing of a random secret, and so any subset of k positions will look uniformly random, regardless of whether the data set is a random codeword or a uniformly random string drawn from the entire space. See Ben-Sasson et al. (2003) for a discussion of other, simpler sets that satisfy similar properties.

To reveal an approximate version of $f(\mathbf{x})$ with sensitivity S, one can sample a value according to $\mathbf{p}_{f(\mathbf{x}),\varepsilon/S_{\mathcal{Z}}(f)}()$.

$$\begin{aligned}\mathbf{p}[\mathcal{M}(\mathbf{x}) = y] &= \mathbf{p}_{f(\mathbf{x}),\varepsilon/S_{\mathcal{Z}}(f)}(y) \\ &= \frac{\exp\left(\frac{\varepsilon}{2S_{\mathcal{Z}}(f)} \cdot d_{\mathcal{Z}}(y, f(\mathbf{x}))\right)}{\int_{z\in\mathcal{Z}} \exp\left(\frac{\varepsilon}{2S_{\mathcal{Z}}(f)} \cdot d_{\mathcal{Z}}(z, f(\mathbf{x}))\right) dz}. \end{aligned} \tag{6}$$

Theorem 3.7. *In a metric space where $\mathbf{p}_{f(\mathbf{x}),\varepsilon/S_{\mathcal{Z}}(f)}()$ is well-defined for all $\mathbf{x}$, adding noise to $f(\mathbf{x})$ as in Eqn.* (6) *preserves ε-differential privacy.*

Proof. Let $\mathbf{x}$ and $\mathbf{x}'$ be two neighboring databases. The distance $d_{\mathcal{Z}}(f(\mathbf{x}), f(\mathbf{x}'))$ is at most $S_{\mathcal{Z}}(f)$. For all y we get

$$\begin{aligned}\frac{\exp\left(\frac{\varepsilon}{2S_{\mathcal{Z}}(f)} \cdot d_{\mathcal{Z}}(y, f(\mathbf{x}))\right)}{\exp\left(\frac{\varepsilon}{2S_{\mathcal{Z}}(f)} \cdot d_{\mathcal{Z}}(y, f(\mathbf{x}'))\right)} &= \exp\left(\frac{\varepsilon}{2S_{\mathcal{Z}}(f)} \cdot (d_{\mathcal{Z}}(y, f(\mathbf{x})) - d_{\mathcal{Z}}(y, f(\mathbf{x}')))\right) \\ &\leq \exp\left(\frac{\varepsilon}{2S_{\mathcal{Z}}(f)} \cdot (d_{\mathcal{Z}}(f(\mathbf{x}), f(\mathbf{x}')))\right) \\ &\leq e^{\varepsilon/2},\end{aligned}$$

where the first inequality follows by triangle inequality and the second inequality follows from the definition of $S_{\mathcal{Z}}(f)$.

For the normalization factor $\int_{y\in\mathcal{Z}} \exp\left(\frac{\varepsilon \cdot d_{\mathcal{Z}}(y, f(\mathbf{x}))}{2S(f)}\right) dy$ a similar analysis gives that on any point y the integrand differs on $\mathbf{x}$ and $\mathbf{x}'$ by at most a factor of $e^{\varepsilon/2}$. Hence

$$\frac{\int_{y\in\mathcal{Z}} \exp\left(\frac{\varepsilon \cdot d_{\mathcal{Z}}(y, f(\mathbf{x}'))}{2S(f)}\right) dy}{\int_{y\in\mathcal{Z}} \exp\left(\frac{\varepsilon \cdot d_{\mathcal{Z}}(y, f(\mathbf{x}))}{2S(f)}\right) dy} \leq e^{\varepsilon/2}.$$

We conclude that the ratio $\mathbf{p}_{f(\mathbf{x}),\varepsilon/S_{\mathcal{Z}}(f)}(y) \Big/ \mathbf{p}_{f(\mathbf{x}'),\varepsilon/S_{\mathcal{Z}}(f)}(y)$ is bounded by $e^{\varepsilon/2} \cdot e^{\varepsilon/2} = e^{\varepsilon}$, as desired. □

Remark 3.1. One can dispense with the factor of 2 in the definition of $\mathbf{p}_{z,\varepsilon}()$ in cases where the normalization factor does not depend on $\mathbf{x}$. This introduces slightly less noise. ♢

As a simple example, consider a function whose output lies in the Hamming cube $\{0,1\}^d$. By Theorem 3.7, one can release $f(\mathbf{x})$ safely by flipping each bit of the output $f(\mathbf{x})$ independently with probability roughly $\frac{1}{2} - \frac{\varepsilon}{2S(f)}$.

3.4 Bibliographic Notes and Discussion

The mechanisms discussed in this section add data-independent noise to the value of a function of the data set. There are now many differentially private mechanisms based on more sophisticated kinds of randomization. However, the Laplace mechanism of Proposition 3.3 remains widely used, both as is, and as a subroutine in more sophisticated differentially private algorithms. The Laplace mechanism is also the basis for most current implementations of differentially private mechanisms.

While L_1 sensitivity is often the right notion for differential privacy, it is not the only relevant notion of sensitivity. For example, L_2 sensitivity is the right notion for adding Gaussian noise instead of Laplace noise; this yields (ε, δ)-differential privacy Dwork et al. (2006a). Subsequent to the initial version of this work, research showed that *data-dependent* distortion is often much more powerful. The framework of *smooth sensitivity* Nissim et al. (2007) allows the release of $f(\mathbf{x})$ with noise magnitude that depends both on f and $\mathbf{x}$. To ensure that the noise magnitude does not leak information about $\mathbf{x}$, noise is calibrated to a measure of variability of f in the neighborhood of the instance $\mathbf{x}$. In the PTR (propose-test-release) framework Dwork and Lei (2009), a mechanism first tests (with differential privacy) that the sensitivity of f is low in the neighborhood of the dataset $\mathbf{x}$, and only proceeds if the test indicates low sensitivity, in which case Laplace noise is added to $f(\mathbf{x})$.

A significant line of work investigates the release of functions f that consist of m linear queries. The seminal paper of Blum et al. (2008) showed that data-dependent noise could allow one to release exponentially many such queries with high accuracy. This led to a large body of follow-up work, for example, the development of on-line techniques based on the multiplicative weights algorithm Dwork et al. (2009); Hardt and Rothblum (2010), and geometric techniques that vastly generalize Section 3.3 (e.g., Hardt and Talwar (2010)). These investigations have also led to a fascinating interplay with computational complexity theory (e.g., Dwork et al. (2009); Vadhan (2016)).

The Laplace mechanism applies only to functions that return numerical values, but there is also now a wide range of differentially private mechanisms for other types of outputs. The exponential mechanism McSherry and Talwar (2007) extends the Laplace mechanism as one of the basic constructions for differential privacy. This mechanism provides the basis for both practical algorithms and fundamental feasibility results, for example on differentially private learning Kasiviswanathan et al. (2008).

4 Separating Interactive from Non-Interactive Mechanisms

Consider a query set of interest, and consider database access mechanisms providing (distorted) answers to queries from this set. So far, we have focused on mechanisms based on Laplace noise (Proposition 3.3) that answer one query at a time. However, ideally, one would want to release a single "synopsis" that allows users to answer many different queries. More formally, a non-interactive ε-differentially private mechanism for

answering a set F of queries consists of two algorithms $\mathcal{M}_{NI} = (\mathcal{M}, \mathsf{Eval})$ where $\mathcal{M}$ is is an ε-differentially private mechanism. First, $\mathcal{M}$ is executed on an input dataset $\mathbf{x}$ and outputs a "summary" s (this can be an arbitraty data structure). Then, $\mathsf{Eval}(s, f)$ is executed to answer the query function f in the set F. We show that generating such synopses differentially privately is impossible when the set F is very large.

In this section, it will be more convenient to work with the Hamming distance version of differential privacy. Here data sets are ordered vectors of a known length n, with entries in a domain D (so that $\mathbf{x} \in D^n$). Two data sets are neighbors if they differ in one element, and the privacy condition holds for all such pairs of neighbors. As usual, ε-differential privacy requires probabilities of events to be within a factor of e^{ε} on any pair of neighboring databases.

We consider "counting" queries, specified by a predicate $v : [n] \times D \to \{0, 1\}$, where the query value is

$$\mathrm{SUM}_v(\mathbf{x}) = \sum_{i=1}^{|\mathbf{x}|} v(i, x_i).$$

Note that the the predicate takes as input the index i as well as x_i.

Observation 1. *The sensitivity of* SUM_v *is* $S(\mathrm{SUM}_v) = \max_{x \in D} v(i, x) \leq 1$. *Hence, the mechanism that answers any* single *query with Laplace noise* $\mathrm{Lap}(1/\varepsilon)$ *is* ε*-differentially private. This mechanism gives a good additive approximation to* $\mathrm{SUM}_v(\mathbf{x})$ *as long as* ε *is larger than* $1/|\mathbf{x}|$.

We show below that for any *noninteractive* ε-differentially private mechanism $\mathcal{M}_{NI}$, there are many functions v for which $\mathrm{SUM}_v()$ cannot be answered by $\mathcal{M}_{NI}$ unless the dataset consists of at least about $|D|^{1/3}$ points. For these queries, it is not possible to distinguish the synopsis of a dataset in which $v(i, x_i) = 0$ for all $x_i \in \mathbf{x}$ from a dataset in which $v(i, x_i) = 1$ for all $x_i \in \mathbf{x}$.

Let $D = \{0, 1\}^d$. We look at a class of predicates based on the inner product modulo 2. Given strings $r, x \in \{0, 1\}^d$, let $r \odot x$ denote the modulo 2 inner product of r and x, that is $r \odot x = \bigoplus_j x^{(j)} r^{(j)}$. Given a list of strings $\mathbf{r} = (r_1, r_2, ..., r_n) \in (\{0, 1\}^d)^n$, we can define a predicate $v_{\mathbf{r}} : [n] \times D \to \{0, 1\}$ by $v_{\mathbf{r}}(i, x) = r_i \odot x$. For such a predicate $v_{\mathbf{r}}$ and a data set $\mathbf{x} = (x_1, x_2, ..., x_n)$, we have

$$\mathrm{SUM}_{v_{\mathbf{r}}}(\mathbf{x}) = \sum_{i=1}^{n} v_{\mathbf{r}}(i, x_i) = \sum_{i=1}^{n} r_i \odot x_i \,.$$

Theorem 4.1 (Non-interactive Mechanisms Require Large Databases)**.** *Suppose that* $\mathcal{M}$ *is an* ε*-differentially private* non-interactive *mechanism with domain* $D = \{0, 1\}^d$. *For all* n *and for at least* $2/3$ *of the strings* $\mathbf{r} = (r_1, ..., r_n)$ *in* $(\{0, 1\}^d)^n$, *the following two distributions have statistical difference* $O(n^{4/3}\varepsilon^{2/3}2^{-d/3})$*:*

Distribution 0: $\mathcal{M}(\mathbf{x})$ *where* $\mathbf{x} = \{(1, x_2), \ldots, (n, x_n)\}, x_i \in_R \{\mathbf{x} \in D^n : \ \mathrm{SUM}_{v_{\mathbf{r}}}(\mathbf{x}) = 0\}$,

Distribution 1: $\mathcal{M}(\mathbf{x})$ *where* $\mathbf{x} = \{(1, x_2), \ldots, (n, x_n)\}, x_i \in_R \{\mathbf{x} \in D^n : \ \mathrm{SUM}_{v_{\mathbf{r}}}(\mathbf{x}) = n\}$.

The proof of Theorem 4.1 is postponed until Section 4.2, where it accompanies a similarly structured proof for a second separation result stated in Proposition 4.2 below.

For now, observe that Theorem 4.1 implies that in any given run of the mechanism $\mathcal{M}$, it is impossible to learn a reasonable approximation to $\mathrm{SUM}_{v_\mathbf{r}}(\mathbf{x})$ (since even distinguishing between its extreme values, 0 and n, is impossible to do reliably). The order of the quantifiers is important: as pointed out in Observation 1, there exists a mechanism that can accurately answer any particular query $\mathrm{SUM}_{v_\mathbf{r}}(\cdot)$. However, no single non-interactive scheme can *simultaneously* answer most queries of this form, unless n grows exponentially with d.

The strong, multiplicative notion of ε-differential privacy in Definition 2.1 is essential to Theorem 4.1. Consider, for example, the candidate synopsis which outputs m pairs (i, x_i) for uniformly random indices in i in $[n]$. When $m = \theta(1)$ this is essentially Example 3; it fails to satisfy differential privacy but yields $O(1/n)$-close distributions on every pair of neighboring, n-entry databases. However, such a mechanism does permit estimating $\mathrm{SUM}_{v_\mathbf{r}}(\mathbf{x})$ for most $\mathbf{r}$ within additive error $O(n/\sqrt{m})$. Thus, even for constant m, this is better than what is possible for any ε-differentially private non-interactive mechanism scheme when $n = 2^{o(d)}$.

Relation to Previous Impossibility Results. For binary data sets (where $D = \{0, 1\}$), Dinur and Nissim Dinur and Nissim (2003) showed that every synopsis mechanism that answers *all* 2^n possible counting queries within additive error cn, for small enough c, breaches any reasonable notion of privacy in that it allows one to reconstruct almost the entire data set. The statement of Theorem 4.1 is incomparable: in one sense, it is weaker since it is specific to differential privacy. On the other hand, it is stronger since it rules out answering *most* queries within any nontrivial error whatsoever.

One can formulate these impossibility results in different terms. We consider three key parameters: $|D|$, the size of the domain, n the size of the database, and k the number of queries that cannot be reliably answered. The results of Dinur and Nissim Dinur and Nissim (2003) use $k = 2^n = |D|^n$ queries to reconstruct a dataset with n. In contrast, our results rule out differentially private mechanisms that answer $k = 2^{nd} = |D|^n$ queries (so the size of D is larger, but the relation between D, n and k remains the same). We discuss more recent lower bounds in Section 4.4.

4.1 A Stronger Separation for Local Mechanisms (Randomized Response)

Local mechanisms (sometimes called *randomized response* schemes) are a class of non-interactive schemes in which each user's data is perturbed individually, and then the perturbed values are published or collected by the analyst. Formally, a local algorithm is a randomization operator $Z : D \to \{0, 1\}^*$ such that

$$\mathcal{M}_Z(x_1, \ldots, x_n) = (Z(x_1), \ldots, Z(x_n)).$$

With a local algorithm, no central server need ever see the individuals' private data: each user i computes $Z(x_i)$ and releases only that.[4] Local mechanisms have their roots in Warner (1965).

We now strengthen Theorem 4.1 for randomized response schemes. We consider a simpler class of sum queries, where the same parity predicate is applied to all element of the data set. Specifically, let $D = \{0,1\}^d$ and , for $r \in \{0,1\}^d$, let $v_r(x) = r \odot x$. For most vectors r, we show that the parity check $v_r(x) = r \odot x$ will be difficult to learn from $Z(x)$, and so $f(\mathbf{x}) = \mathrm{SUM}_{v_r}(\mathbf{x})$ will be difficult to learn from $\mathcal{M}_Z(\mathbf{x})$ unless n is exponentially large in d.

Proposition 4.2 (Impossibility result for local mechanisms)**.** *Suppose that $\mathcal{M}$ is a ε-differentially private randomized response mechanism. For at least $2/3$ of the values $r \in \{0,1\}^d \setminus \{0^d\}$, the following two distributions have statistical difference $O(n\varepsilon^{2/3}2^{-d/3})$:*

$$\textit{Distribution 0: } \mathcal{M}_Z(\mathbf{x}) \textit{ where each } x_i \in_R \left\{x \in \{0,1\}^d : \ r \odot x = 0\right\}$$
$$\textit{Distribution 1: } \mathcal{M}_Z(\mathbf{x}) \textit{ where each } x_i \in_R \left\{x \in \{0,1\}^d : \ r \odot x = 1\right\}$$

In particular, if $n = o(2^{d/3}/\varepsilon^{2/3})$, no user can learn the relative frequency of database items satisfying the predicate $g_r(x) = r \odot x$, for most values $r \in \{0,1\}^d$. Substituting in $k = 2^d$, we get a lower bound that $n = \Omega((k/\varepsilon^2)^{1/3})$ data points are necessary to answer k counting queries when $|D| \geq \log(k)$.

4.2 Proving the Separation Results

The two proofs have the same structure: a hybrid argument with a chain of length $2n$, in which the bound on statistical distance at each step in the chain is given by Lemma 4.3 below. Adjacent elements in the chain will differ according to the domain from which one of the entries in the database is chosen, and the elements in the chain are the probability distributions of the sanitizations when the database is chosen according to the given n-tuple of distributions.

For any r, partition the domain D into two sets: $D_r = \left\{x \in \{0,1\}^d : r \odot x = 0\right\}$, and $\bar{D}_r = D \setminus D_r = \left\{x \in \{0,1\}^d : r \odot x = 1\right\}$. We abuse notation and let D_r also stand for a random vector chosen uniformly from D_r (similarly for D and $\bar{D}_r$).

The intuition for the key step is as follows. Given a randomized map $Z : D \to \{0,1\}^*$, we wish to show that the quantity $\Pr[Z(D_r) = z]$ estimates $\Pr[Z(D) = z]$ with very low *multiplicative* error (much lower than $1 + \varepsilon$). This will allow us to show that $Z(D_r)$ and Z are very close in total variation distance, without having to pay a factor proportional to the domain size (as we would if we only bounded the additive error between $\Pr[Z(D_r) = z]$ and $\Pr[Z(D) = z]$). Two important facts come into play. First, when r is chosen at random, D_r consists of 0^d (which doesn't affect things significantly) together with $2^{d-1} - 1$ points chosen pairwise independently in $\{0,1\}^d$.

[4] This local version of differential privacy is equivalent to the notion of γ-*amplification* of Evfimievski et al. (2003), with $\gamma = e^{\varepsilon}$.

Second, ε-differential privacy implies that each of the terms $\Pr[Z(x) = z]$ lies in a small multiplicative interval around their expectation $\Pr[Z(D) = z]$. This implies a very small variance for the estimator, which can be combined with pairwise independence to get the desired bound.

Lemma 4.3. *Let $Z : D \to \{0,1\}^*$ be a randomized map such that for all pairs $x, x' \in D$, and all outputs z, $\Pr[Z(x) = z] \leq e^{\varepsilon} \Pr[Z(x') = z]$. Let $0 < \alpha$. With probability at least $1 - \alpha$ over $r \in_R \{0,1\}^d \setminus \{0^d\}$,*

$$\mathbf{SD}\left(Z(D_r)\ ,\ Z(D)\right) \leq O\Big(\frac{\varepsilon^2}{\alpha \cdot 2^d}\Big)^{1/3}.$$

Similarly, With probability at least $1-\alpha$ over $r \in_R \{0,1\}^d \setminus \{0^d\}$, $\mathbf{SD}\left(Z(\bar{D}_r)\ ,\ Z(D)\right) \leq O\Big(\frac{\varepsilon^2}{\alpha \cdot 2^d}\Big)^{1/3}$.

The lemma is proved in Section 4.3. We now use it to prove the two separation results.

Proof of Theorem 4.1. Distribution 0 in the statement is $\mathcal{M}(D_{r_1}, \ldots, D_{r_n})$. We show that with high probability over the choice of the r_i's, this distribution is close to the distribution induced by a uniform input, i.e. $\mathcal{M}(D, \ldots, D)$. We proceed by a hybrid argument, adding one constraint at a time. For each i, we want to show that the following hybrids are statistically close:

$$\begin{array}{ll} i\text{-th hybrid:} & \mathcal{M}(D_{r_1}, \ldots, D_{r_i}, D\ \ \ , D, \ldots, D) \\ (i+1)\text{-st hybrid:} & \mathcal{M}(D_{r_1}, \ldots, D_{r_i}, D_{r_{i+1}}, D, \ldots, D). \end{array}$$

Suppose that we have chosen $r_1, \ldots, r_i$ already. For any $x \in \{0,1\}^d$, consider the randomized map where the $(i+1)$-th coordinate is fixed to x:

$$Z(x) = \mathcal{M}(D_{r_1}, \ldots, D_{r_i},\ x\ , D, \ldots, D).$$

Note that $Z(D)$ is equal to the i-th hybrid, and $Z(D_{r_{i+1}})$ is equal to the $(i+1)$-st hybrid. Because $\mathcal{M}$ is ε-differentially private, it follows that $Z(\cdot)$ satisfies $\Pr[Z(x) = z] \leq e^{\varepsilon} \Pr[Z(x') = z]$ for all $x, x' \in D$ and all z, and hence we can apply Lemma 4.3 and get that with probability at least $1 - \frac{1}{6n}$ over r_{i+1},

$$\mathbf{SD}\left(Z(D_{r_i}), Z(D)\right) = \sigma = O(\sqrt[3]{n\varepsilon^2/2^d}).$$

By a union bound, for all i the statistical distance between hybrids i and $i+1$ is bounded by σ with probability at least $\frac{5}{6}$. We hence conclude that with probability $\frac{5}{6}$,

$$\mathbf{SD}\left(\mathcal{M}(D_{r_1}, \ldots, D_{r_n}), \mathcal{M}(D, \ldots, D)\right) \leq n\sigma.$$

Applying the same reasoning to hybrids starting from Distribution1, i.e., $\mathcal{M}(\bar{D}_{r_1}, \ldots, \bar{D}_{r_n})$, and ending with $\mathcal{M}(D, \ldots, D)$ we get that $\mathbf{SD}\left(\mathcal{M}(D_{r_1}, \ldots, D_{r_n}), \mathcal{M}(D, \ldots, D)\right) \leq n\sigma$ with probability at least $\frac{5}{6}$.

We conclude that with probability at least 2/3, both chains of hybrids accumulate statistical difference bounded by $n\sigma$, and the distance between Distributions 0 and 1 is at most $2n\sigma = O(n^{4/3}\varepsilon^{2/3}2^{-d/3})$, as claimed. □

Proof of Proposition 4.2. Let $\mathcal{M}_Z$ be an ε-differentially private randomized reponse scheme, i.e., there is a randomized map $Z()$ from D to $\{0,1\}^*$, such that $\mathcal{M}_Z(x_1,\ldots,x_n) = (Z(x_1),\ldots,Z(x_n))$. Since $\mathcal{M}_Z$ is ε-differentially private, then it must hold that $\Pr[Z(x) = z] \leq e^{\varepsilon}\Pr[Z(x') = z]$ for all $x, x' \in D$ and for all outputs z.

It is sufficient to show that with probability at least 2/3 over a random choice $r \in_R \{0,1\}^d \setminus \{0\}$, the distributions $Z(D_r)$ and $Z(\bar{D}_r)$ are within statistical difference $O(\varepsilon^{2/3}2^{-d/3})$. This follows by applying Lemma 4.3 with $\alpha = 1/3$. By a hybrid argument, the difference between Distributions 0 and 1 above is then $O(n\varepsilon^{2/3}2^{-d/3})$. □

4.3 Proving that Random Subsets Approximate the Output Distribution

We now recall and prove Lemma 4.3.

Lemma 4.3. *Let $Z : D \to \{0,1\}^*$ be a randomized map such that for all pairs $x, x' \in D$, and all outputs z, $\Pr[Z(x) = z] \leq e^{\varepsilon}\Pr[Z(x') = z]$. Let $0 < \alpha$. With probability at least $1-\alpha$ over $r \in_R \{0,1\}^d \setminus \{0^d\}$,*

$$\mathbf{SD}\left(Z(D_r)\ ,\ Z(D)\right) \leq O\Big(\frac{\varepsilon^2}{\alpha \cdot 2^d}\Big)^{1/3}.$$

Similarly, With probability at least $1-\alpha$ over $r \in_R \{0,1\}^d \setminus \{0^d\}$, $\mathbf{SD}\left(Z(\bar{D}_r)\ ,\ Z(D)\right) \leq O\Big(\frac{\varepsilon^2}{\alpha \cdot 2^d}\Big)^{1/3}$.

Proof. Let $p_x(z)$ denote the probability that $Z(x) = z$. If x is chosen uniformly in $\{0,1\}^d$, then the probability of outcome z is $p(z) = \frac{1}{2^d}\sum_x p_x(z)$.

For $r \in \{0,1\}^d$ and $b \in \{0,1\}$ let $D_{r,b} = \left\{x \in \{0,1\}^d : r \odot x = b\right\}$. This choice of picking not only the string r but also an affine term $b \in \{0,1\}$ simplifies our calculations. One can think of $\Pr[Z(D_{r,b}) = z]$ as estimating $p(z)$ by pairwise-independently sampling $2^d/2$ values from the set D and only averaging over that subset. By the assumption on Z, the values $p_x(z)$ all lie in the interval $p(z)\cdot[e^{-\varepsilon}, e^{\varepsilon}]$, which is of width $(e^{\varepsilon} - e^{-\varepsilon})p(z) \approx 2\varepsilon p(z)$ around $p(z)$. This estimator will hence have small standard deviation, which we will use to bound the statistical difference.

For $r \in \{0,1\}^d$ and $b \in \{0,1\}$ let $\hat{p}(z; r, b) = \Pr[Z(D_{r,b}) = z]$, where the probability is taken over the coin flips of Z and the choice of $x \in D_{r,b}$. For a fixed z, $\hat{p}(z; r, b)$ is a random variable depending on the choice of r, b satisfying

$$\mathop{\mathbb{E}}_{r \in_R \{0,1\}^d, b \in_R \{0,1\}}[\hat{p}(z; r, b)] = p(z).$$

Claim 4.4. *Let* $\tilde{\varepsilon} = e^{\varepsilon} - 1$. $\mathsf{Var}_{r,b}\left[\hat{p}(z; r, b)\right] \leq \dfrac{2 \cdot \tilde{\varepsilon}^2 \cdot p(z)^2}{2^d}$, *where the randomness is over the choice* r, b *uniformly at random from* $\{0,1\}^d, \{0,1\}$ *resp.*

Proof. Let p^* be the minimum over x of $p_x(z)$. Let $q_x = p_x(z) - p^*$ and $\bar{q} = p(z) - p^*$. The variance of $\hat{p}(z)$ is the same as the variance of $\hat{p}(z) - p^*$. We can write $\hat{p}(z) - p^*$ as $\frac{2}{2^d}\sum_x q_x \chi_0(x)$, where $\chi_0(x)$ is 1 if $x \in D_{r,b}$ and 0 otherwise. The expectation of $\hat{p}(z) - p^*$ is $\bar{q}$, which we can write $\frac{1}{2^d}\sum_x q_x$.

$$\underset{r,b}{\mathsf{Var}}\left[\hat{p}(z)\right] = \underset{r,b}{\mathbb{E}}\left[\left(\tfrac{2}{2^d}\sum_x q_x\chi_0(x) - \tfrac{1}{2^d}\sum_x q_x\right)^2\right] = \underset{r,b}{\mathbb{E}}\left[\left(\tfrac{1}{2^d}\sum_x q_x\big(2\chi_0(x)-1\big)\right)^2\right] \quad (7)$$

Now $\big(2\chi_0(x)-1\big) = (-1)^{r\odot x \oplus b}$. This has expectation 0. Moreover, for $x \neq y$, the expectation of $\big(2\chi_0(x)-1\big)\big(2\chi_0(y)-1\big)$ is exactly $1/2^d$ (if we chose r with no restriction it would be 0, but we have the restriction that $r \neq 0^d$ from the lemma statement). Expanding the square in Eqn. (7),

$$\begin{aligned}
\underset{r,b}{\mathsf{Var}}\left[\hat{p}(z)\right] &= \tfrac{1}{2^{2d}}\sum_x q_x^2 + \tfrac{1}{2^{3d}}\sum_{x\neq y} q_x q_y \\
&= \tfrac{1-\frac{1}{2^d}}{2^{2d}}\sum_x q_x^2 + \tfrac{1}{2^d}\left(\tfrac{1}{2^d}\sum_x q_x\right)^2 \\
&\leq \tfrac{1}{2^d}\left(\max_x q_x^2 + \bar{q}^2\right).
\end{aligned}$$

By the indistinguishability condition, both $(\max_x q_x)$ and $\bar{q}$ are at most $(e^{\varepsilon}-1)p^* \leq \tilde{\varepsilon}\cdot p(z)$. Plugging this into the last equation proves Claim 4.4. □

We now complete the proof of Lemma 4.3. We say that a value z is δ-*good* for a pair (r, b) if $\hat{p}(z) - p(z) \leq \delta \cdot p(z)$. By the Chebyshev bound, for all z,

$$\Pr_{r,b}\left[z \text{ is not } \delta\text{-good for } (r,b)\right] \leq \frac{\mathsf{Var}\left[\hat{p}(z)\right]}{\delta^2 p(z)^2} \leq \frac{2\tilde{\varepsilon}^2}{\delta^2 2^d}.$$

If we take the distribution on z given by $p(z)$, then with probability at least $1-\alpha$ over pairs (r, b), the fraction of z's (under $p(\cdot)$) which are good is at least $1 - \frac{2\tilde{\varepsilon}^2}{\alpha\delta^2 2^d}$.

Finally, if a $1-\gamma$ fraction of the z's are δ-good for a particular pair (r, b), then the statistical difference between the distribution $\hat{p}(z)$ and $p(z)$ is at most $2(\gamma+\delta)$. Setting $\delta = \sqrt[3]{\frac{2\alpha\tilde{\varepsilon}^2}{2^d}}$, we get a total statistical difference of at most 4δ. Since $\tilde{\varepsilon} < 2\varepsilon$ for $\varepsilon \leq 1$, the total distance between $\hat{p}(\cdot)$ and $p(\cdot)$ is at most $4\sqrt[3]{12\varepsilon^2 2^{-d}}$, for at least a $1-\alpha$ fraction of the pairs (r, b). The bit b is unimportant here since it only switches D_r and its complement $\bar{D}_r$. The distance between $Z(D_r)$ and $Z(D)$ is exactly the same as the distance between $Z(\bar{D}_r)$ and $Z(D)$, since $Z(D)$ is the mid-point between the two. Thus, the statement holds even over pairs of the form $(r, 0)$. This proves Lemma 4.3. □

4.4 Bibliographic Notes and Discussion

There is a large and active literature developing lower bounds for various notions of privacy. Some of these are specific to differential privacy and its variants (as with the bounds in thispaper), while others (such as those of Dinur and Nissim Dinur and Nissim (2003)) are more general and provide concrete algorithmic attacks, essentially ruling out all reasonable notions of privacy.

Also relevant is the large literature on developing algorithms for answering very high-dimensional queries subject to differential privacy, starting with the work of Blum, Ligett and Roth Blum et al. (2008). Our bounds (and those of Dinur and Nissim (2003)) show that to accurately answer k counting queries, one must roughly have n at least logarithmic in k (unless $|D|$ is small, less than a specific polynomial in n). The algorithms of Blum et al. (2008) and subsequent works provide nontrivial answers (with error $o(n)$) to k counting queries provided $n \geq \text{poly}(\log k, \log |D|)$, where the polynomial varies depending on the context and choice of algorithm.

Acknowledgments

Conversations with Helen Nissenbaum introduced Dwork to the general problems of privacy in the digital age and inspired her to find a piece of the privacy puzzle amenable to mathematical analysis. We thank Moni Naor and Prahladh Harsha for helpful discussions during the writing of the original version of this work, and the graduate students in Smith's data privacy class at the Weizmann Institute, whose questions prompted some of the results in this paper.

Finally, the authors thank Mike Schroeder and, especially, Roy Levin for creating and nurturing a world-class research environment at Microsoft Research, Silicon Valley. Dwork and McSherry were members of the MSR Silicon Valley lab for more than a decade; both Nissim and Smith were lab visitors; Smith's internship there kindled his interest in data privacy.

References

Adam, N. R. and Wortmann, J. C. (1989). Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys*, 25(4).

Agrawal, D. and Aggarwal, C. C. (2001). On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the Twentieth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. ACM.

Agrawal, R. and Srikant, R. (2000). Privacy-preserving data mining. In Chen, W., Naughton, J. F., and Bernstein, P. A. (eds.), *SIGMOD Conference*, 439–450. ACM.

Bassily, R., Groce, A., Katz, J., and Smith, A. (2013). Coupled-worlds privacy: Exploiting adversarial uncertainty in private data analysis. In *Foundations of Computer Science (FOCS)*.

Ben-Sasson, E., Harsha, P., and Raskhodnikova, S. (2003). Some 3cnf properties are hard to test. In *STOC*, 345–354. ACM.

Blum, A., Dwork, C., McSherry, F., and Nissim, K. (2005). Practical privacy: The SuLQ framework. In *PODS*.

Blum, A., Ligett, K., and Roth, A. (2008). A learning theory approach to non-interactive database privacy. In *STOC*, 609–618. ACM.

Bun, M. and Steinke, T. (2016). Concentrated differential privacy: Simplifications, extensions, and lower bounds. *arXiv preprint arXiv:1605.02065*.

Chawla, S., Dwork, C., McSherry, F., Smith, A., and Wee, H. (2005a). Toward privacy in public databases. In *Theory of Cryptography Conference (TCC)*, 363–385.

Chawla, S., Dwork, C., McSherry, F., and Talwar, K. (2005b). On the utility of privacy-preserving histograms. In *21st Conference on Uncertainty in Artificial Intelligence (UAI)*.

Denning, D. E. (1980). Secure statistical databases with random sample queries. *ACM Transactions on Database Systems*, 5(3): 291–315.

Dinur, I. and Nissim, K. (2003). Revealing information while preserving privacy. In *PODS*, 202–210.

Dwork, C. (2006). Differential privacy. In *ICALP*, 1–12.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. (2006a). Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, 486–503. URL `DBLP, http://dblp.uni-trier.de`

Dwork, C. and Lei, J. (2009). Differential privacy and robust statistics. In *STOC*, 371–380. ACM.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006b). Calibrating noise to sensitivity in private data analysis. In *TCC*, 265–284.

Dwork, C. and Naor, M. (2010). On the difficulties of disclosure prevention in statistical databases or the case for differential privacy. *J. Privacy and Confidentiality*, 2(1).

Dwork, C., Naor, M., Reingold, O., Rothblum, G. N., and Vadhan, S. P. (2009). On the complexity of differentially private data release: efficient algorithms and hardness results. In *STOC*, 381–390.

Dwork, C. and Nissim, K. (2004). Privacy-preserving datamining on vertically partitioned databases. In *CRYPTO*, 528–544.

Dwork, C. and Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4): 211–407.

Dwork, C. and Rothblum, G. N. (2016). Concentrated differential priacy. *arXiv preprint arXiv:1603.01887*.

Dwork, C., Rothblum, G. N., and Vadhan, S. P. (2010). Boosting and differential privacy. In *FOCS*.

Erlingsson, Ú., Pihur, V., and Korolova, A. (2014). Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 1054–1067. ACM.

Evfimievski, A. V., Gehrke, J., and Srikant, R. (2003). Limiting privacy breaches in privacy preserving data mining. In *PODS*, 211–222.

Federighi, C. (2016). Apple Worldwide Developers Conference 2016 (WWDC 2016) Keynote. Available from `http://www.apple.com/apple-events/june-2016/`. Privacy starts at 1:41:07. Downloaded June 13, 2016.

Ganta, S. R., Kasiviswanathan, S. P., and Smith, A. (2008). Composition attacks and auxiliary information in data privacy. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 265–273. ACM.

Goldwasser, S. and Micali, S. (1984a). Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2): 270–299.

— (1984b). Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2): 270–299.

Hardt, M., Miklau, G., Pierce, B., and Roth, A. (2016). Video tutorials from DIMACS Workshop on Recent Work on Differential Privacy across Computer Science (Rutgers University, December 2012).
URL `http://dimacs.rutgers.edu/Workshops/DifferentialPrivacy/Slides/slides.html`

Hardt, M. and Rothblum, G. N. (2010). A Multiplicative Weights Mechanism for Privacy-Preserving Data Analysis. In *FOCS*, 61–70.

Hardt, M. and Talwar, K. (2010). On the geometry of differential privacy. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, 705–714. New York, NY, USA: ACM.
URL http://doi.acm.org/10.1145/1806689.1806786

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. (2008). What can we learn privately? In *FOCS*, 531–540. IEEE Computer Society.

Kasiviswanathan, S. P. and Smith, A. (2008). On the 'semantics' of differential privacy: A bayesian formulation. *CoRR*, arXiv:0803.39461.

Kifer, D. and Machanavajjhala, A. (2011). No Free Lunch in Data Privacy. In *SIGMOD*, 193–204.

Ligett, K. et al. (2016). Video tutorials from Simons Institute Workshop on Big Data and Differential Privacy (UC Berkeley, December 2013).
URL http://dimacs.rutgers.edu/Workshops/DifferentialPrivacy/Slides/slides.html

Machanavajjhala, A., Kifer, D., Abowd, J. M., Gehrke, J., and Vilhuber, L. (2008). Privacy: Theory meets practice on the map. In Alonso, G., Blakeley, J. A., and Chen, A. L. P. (eds.), *ICDE*, 277–286. IEEE.

McSherry, F. and Talwar, K. (2007). Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, 94–103. Washington, DC, USA: IEEE Computer Society.
URL http://portal.acm.org/citation.cfm?id=1333875.1334185

Nissim, K., Raskhodnikova, S., and Smith, A. (2007). Smooth sensitivity and sampling in private data analysis. In *Symp. Theory of Computing (STOC)*, 75–84. Full paper on authors' web sites.

Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5): 557–570.

Vadhan, S. (2016). The complexity of differential privacy. Unpublished survey.

Warner, S. L. (1965). Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*.

Wasserman, L. and Zhou, S. (2010). A statistical framework for differential privacy. *J. American Statistical Association*, 105(489): 375–389.

A Simulatability and Inference-based Privacy

Definition 2.1 equates privacy with the inability to distinguish two close databases. This is a convenient notion to work with (as is indistinguishability of encryptions Goldwasser

and Micali (1984a)); however, it does not directly say what an adversary may do and learn. In this section we present some "semantically" flavored definitions of privacy, and show their equivalence to Definition 2.1.

Because of the need to have some utility conveyed by the database, it is not possible to get as strong a notion of security as we can, say, with encryption. We discuss two definitions which we consider meaningful, suggestively named *simulatability* and *semantic privacy*. Simulatability requires that the adversary's view when interacting with the mechanism can be "faked" even when the data of any (small) subset of individuals have been removed.

Definition A-1. *A mechanism is (k, ε)-differentially private if for all pairs $\mathbf{x}, \mathbf{x}'$ which differ in at most k entries (that is, $d_{\triangle}(\mathbf{x}, \mathbf{x}') \leq k$), for all adversaries $\mathcal{A}$ and for all events S in the output space,* $\Pr(\mathcal{M}(\mathbf{x}) \in S) \leq e^{\varepsilon} \Pr(\mathcal{M}(\mathbf{x}') \in S)$.

Recall that $A \approx_{\varepsilon} B$ denotes that the probability distributions of the random variables A and B are within multiplicative distance ε.

Definition A-2. *A mechanism $\mathcal{M}$ is* (k, ε)-simulatable *if for every adversary $\mathcal{A}$, there exists a* simulator *$\mathcal{A}'$ such that, for every $\mathbf{x} \in \mathbb{N}^D$, for every $I \subseteq \mathbf{x}$ of size k,*

$$\mathsf{View}_{\mathcal{M},\mathcal{A}}(\mathbf{x}) \approx_{\varepsilon} \mathcal{A}'(\mathbf{x} \setminus I)$$

Lemma A-3. *1. Every (k, ε)-differentially private mechanism is (k, ε)-simulatable.*

2. Every (k, ε)-simulatable mechanism is $(k, 2\varepsilon)$-differentially private.

Proof. (1) Suppose $\mathcal{Z}$ is (k, ε)-differentially private. Fix the adversary $\mathcal{A}$. On input $\mathbf{x}' = \mathbf{x} \setminus I$, the simulator $\mathcal{A}'$ runs an interaction between $\mathcal{A}$ and $\mathcal{M}(\mathbf{x}')$. Thus, $\mathcal{A}'(\mathbf{x}) = \mathsf{View}_{\mathcal{M},\mathcal{A}}(\mathbf{x}') \approx_{\varepsilon} \mathsf{View}_{\mathcal{M},\mathcal{A}}(\mathbf{x})$ (since $d_{\triangle}(\mathbf{x}, \mathbf{x}') \leq k$).

(2) Suppose $\mathcal{M}$ is (k, ε)-simulatable. Suppose that $\mathbf{x}, \mathbf{x}'$ differ in at most k entries, that is, $d_{\triangle}(\mathbf{x}, \mathbf{x}') \leq k$. Let I be their symmetric difference, so that $\mathbf{x} \setminus I = \mathbf{x}' \setminus I$. Simulatability implies that there exists a simulator $\mathcal{A}'$ such that $\mathcal{A}'(\mathbf{x} \setminus I) \approx_{\varepsilon} \mathcal{M}(\mathbf{x})$ and $\mathcal{A}'(\mathbf{x}' \setminus I) \approx_{\varepsilon} \mathcal{M}(\mathbf{x}')$. By construction of I, the input to $\mathcal{A}'$ is the same in both these equations. By the triangle inequality, the distributions of $\mathcal{M}(\mathbf{x})$ and $\mathcal{M}(\mathbf{x}')$ differ by at most 2ε. □

Simulatability still leaves implicit what, exactly, the adversary can compute about the database. Following the spirit of the definition of semantic security of encryptions Goldwasser and Micali (1984b), we may ask how an adversary's knowledge about a person changes after seeing the output of the mechanism. Extending terminology from Blum et al. Blum et al. (2005), we say an adversary is *informed* if she knows all but k database entries before interacting with the mechanism, and tries to learn about the remaining ones. The parameter k measures her remaining uncertainty. Simulatability says that the adversary's view can be simulated by an informed adversary—hence, that an informed adversary learns nothing about any particular person.

We now turn to yet another definition, which formalizes the intuition that an informed adversary "learns nothing" about any individual (or small set of individuals). Consider an informed adversary who knows a subset $\mathbf{x}_0$ of a dataset $\mathbf{x}$. Let $\mathbf{x}_1$ denote the remainder $\mathbf{x}_1 = \mathbf{x} \setminus \mathbf{x}_0$. We encode the adversary's a priori information about $\mathbf{x}_0$ via a random variable $\mathbf{X}_1$. We ask: how much does the distribution of $\mathbf{X}_1$ change given the output of the mechanism?

Definition A-4 (Semantic Privacy)**.** *A mechanism is* (k, ε)-semantically private *if for all fixed datasets* $\mathbf{x}_0$*, for all random variables* $\mathbf{X}_1$ *on sets of size at most* k*, and for all outputs* t*, we have*

$$\mathbf{X}_1 \approx_\varepsilon \mathbf{X}_1\big|_{\mathcal{M}(\mathbf{x}_0 \cup \mathbf{X}_1)=t}$$

where $\mathbf{X}_1\big|_{\mathcal{M}(\mathbf{x}_0 \cup \mathbf{X}_1)=t}$ *denotes the conditional distribution of* $\mathbf{X}_1$ *given that* $\mathcal{M}(\mathbf{x}_0 \cup \mathbf{X}_1) = t$*.*

The role of the multiplicative notion of distance used in the definition of differential privacy becomes clear here: it allows one to directly infer statements about how an adversary's prior distribution changes.

Proposition A-5. *A mechanism is* (k, ε)*-differentially private if and only if it is* (k, ε)*-semantically private.*

Proof. (1) Let $\mathcal{M}$ be a (k, ε)-differentially private mechanism. We show that $\mathcal{M}$ is (k, ε)-semantically private. Fix $\mathbf{x}_0$ and a distribution on $\mathbf{X}_1$. For every value $\mathbf{x}_1$ of $\mathbf{X}_1$ and every event $S \subseteq \mathcal{O}$, we need to bound the ratio

$$\frac{\Pr(\mathbf{X}_1 = \mathbf{x}_1 \mid \mathcal{M}(\mathbf{x}_0 \cup \mathbf{X}_1) \in S)}{\Pr(\mathbf{X}_1 = \mathbf{x}_1)}$$

Applying Bayes' rule to the numerator, this ratio equals

$$\frac{\Pr(\mathcal{M}(\mathbf{x}_0 \cup \mathbf{X}_1) \in S \mid \mathbf{X}_1 = \mathbf{x}_1) \cdot \Pr(\mathbf{X}_1 = \mathbf{x}_1)}{\Pr(\mathcal{M}(\mathbf{x}_0 \cup \mathbf{X}_1) \in S) \cdot \Pr(\mathbf{X}_1 = \mathbf{x}_1)} = \frac{\Pr(\mathcal{M}(\mathbf{x}_0 \cup \mathbf{x}_1) \in S)}{\Pr(\mathcal{M}(\mathbf{x}_0 \cup \mathbf{X}_1) \in S)}$$

We can write the denominator of the right-hand side as an expectation over $\mathbf{x}'_1 \sim \mathbf{X}_1$ of the conditional probability $\Pr(\mathcal{M}(\mathbf{x}_0 \cup \mathbf{x}'_1) \in S)$, that is

$$\frac{\Pr(\mathbf{X}_1 = \mathbf{x}_1 \mid \mathcal{M}(\mathbf{x}_0 \cup \mathbf{X}_1) \in S)}{\Pr(\mathbf{X}_1 = \mathbf{x}_1)} = \frac{\Pr(\mathcal{M}(\mathbf{x}_0 \cup \mathbf{x}_1) \in S)}{E_{\mathbf{x}'_1 \sim \mathbf{X}_1} Pr(\mathcal{M}(\mathbf{x}_0 \cup \mathbf{x}'_1) \in S)}$$

Since every term in the expectation is at least $e^{-\varepsilon} \Pr(\mathcal{M}(\mathbf{x}_0 \cup \mathbf{x}_1) \in S)$, the entire ratio is at most e^{ε}, as desired.

(2) Let $\mathcal{M}$ be a (k, ε)-semantically private mechanism. We show that $\mathcal{M}$ is (k, ε)-differentially private. Fix two datasets $\mathbf{x}, \mathbf{x}'$ with symmetric difference of size at most k. The idea is to define a distribution with support on just these two datasets, and apply semantic privacy.

Let $\mathbf{x}_0 = \mathbf{x} \cap \mathbf{x}'$. Let $\mathbf{X}_1$ be a random variable that takes the value $\mathbf{x} \setminus \mathbf{x}_0$ with probability α and the value $\mathbf{x}' \setminus \mathbf{x}_0$ with probability $1 - \alpha$, for some $\alpha > 0$ which we will make tend to 0. Note that $\mathbf{X}_1$ always has size at most k.

By Bayes' rule, for any event $S \subseteq \mathcal{O}$, we have

$$\begin{aligned} \frac{\Pr(\mathbf{X}_1 = \mathbf{x} \setminus \mathbf{x}_0 \mid \mathcal{M}(\mathbf{x}_0 \cup \mathbf{X}_1) \in S)}{\Pr(\mathbf{X}_1 = \mathbf{x} \setminus \mathbf{x}_0)} &= \frac{\Pr(\mathbf{X}_1 = \mathbf{x} \setminus \mathbf{x}_0 \wedge \mathcal{M}(\mathbf{x}) \in S)}{\Pr(\mathbf{X}_1 = \mathbf{x} \setminus \mathbf{x}_0) \Pr(\mathcal{M}(\mathbf{x}_0 \cup \mathbf{X}_1) \in S)} \\ &= \frac{\alpha}{\alpha} \cdot \frac{\Pr(\mathcal{M}(\mathbf{x}) \in S)}{\alpha \Pr(\mathcal{M}(\mathbf{x}) \in S) + (1-\alpha) \Pr(\mathcal{M}(\mathbf{x}') \in S)}. \end{aligned}$$

The left-hand-side above is at most e^{ε}, by semantic privacy, no matter how we set α. In particular, letting α tend to 0, we get that $\frac{\Pr(\mathcal{M}(\mathbf{x}) \in S)}{\Pr(\mathcal{M}(\mathbf{x}') \in S)} \leq e^{\varepsilon}$, as desired. $\square$