Differential Privacy

Cynthia Dwork

Microsoft Research
dwork@microsoft.com

Abstract. In 1977 Dalenius articulated a desideratum for statistical databases: nothing about an individual should be learnable from the database that cannot be learned without access to the database. We give a general impossibility result showing that a formalization of Dalenius' goal along the lines of semantic security cannot be achieved. Contrary to intuition, a variant of the result threatens the privacy even of someone not in the database. This state of affairs suggests a new measure, *differential privacy*, which, intuitively, captures the increased risk to one's privacy incurred by participating in a database. The techniques developed in a sequence of papers [8, 13, 3], culminating in those described in [12], can achieve any desired level of privacy under this measure. In many cases, extremely accurate information about the database can be provided while simultaneously ensuring very high levels of privacy.

1 Introduction

A statistic is a quantity computed from a sample. If a database is a representative sample of an underlying population, the goal of a privacy-preserving statistical database is to enable the user to learn properties of the population as a whole, while protecting the privacy of the individuals in the sample. The work discussed herein was originally motivated by exactly this problem: how to reveal useful information about the underlying population, as represented by the database, while preserving the privacy of individuals. Fortuitously, the techniques developed in [8, 13, 3] and particularly in [12] are so powerful as to broaden the scope of private data analysis beyond this orignal "representatitive" motivation, permitting privacy-preserving analysis of an object that is itself of intrinsic interest. For instance, the database may describe a concrete interconnection network – not a sample subnetwork – and we wish to reveal certain properties of the network without releasing information about individual edges or nodes. We therefore treat the more general problem of *privacy-preserving analysis of data*.

A rigorous treatment of privacy requires definitions: What constitutes a failure to preserve privacy? What is the power of the adversary whose goal it is to compromise privacy? What auxiliary information is available to the adversary (newspapers, medical studies, labor statistics) even without access to the database in question? Of course, utility also requires formal treatment, as releasing no information or only random noise clearly does not compromise privacy;

we will return to this point later. However, in this work privacy is paramount: we will first define our privacy goals and then explore what utility can be achieved given that the privacy goals will be satisified[1].

A 1977 paper of Dalenius [6] articulated a desideratum that foreshadows for databases the notion of semantic security defined five years later by Goldwasser and Micali for cryptosystems [15]: access to a statistical database should not enable one to learn anything about an individual that could not be learned without access[2]. We show this type of privacy cannot be achieved. The obstacle is in *auxiliary information*, that is, information available to the adversary other than from access to the statistical database, and the intuition behind the proof of impossibility is captured by the following example. Suppose one's exact height were considered a highly sensitive piece of information, and that revealing the exact height of an individual were a privacy breach. Assume that the database yields the average heights of women of different nationalities. An adversary who has access to the statistical database and the auxiliary information "Terry Gross is two inches shorter than the average Lithuanian woman" learns Terry Gross' height, while anyone learning only the auxiliary information, without access to the average heights, learns relatively little.

There are two remarkable aspects to the impossibility result: (1) it applies regardless of whether or not Terry Gross is in the database and (2) Dalenius' goal, formalized as a relaxed version of semantic security, cannot be achieved, while semantic security for cryptosystems can be achieved. The first of these leads naturally to a new approach to formulating privacy goals: the risk to one's privacy, or in general, any type of risk, such as the risk of being denied automobile insurance, should not substantially increase as a result of participating in a statistical database. This is captured by *differential privacy*.

The discrepancy between the possibility of achieving (something like) semantic security in our setting and in the cryptographic one arises from the utility requirement. Our adversary is analagous to the eavesdropper, while our user is analagous to the message recipient, and yet there is no decryption key to set them apart, they are one and the same. Very roughly, the database is designed to convey certain information. An auxiliary information generator knowing the data therefore knows much about what the user will learn from the database. This can be used to establish a shared secret with the adversary/user that is unavailable to anyone not having access to the database. In contrast, consider a cryptosystem and a pair of candidate messages, say, $\{0, 1\}$. Knowing which message is to be encrypted gives one no information about the ciphertext; intuitively, the auxiliary information generator has "no idea" what ciphertext the eavesdropper will see. This is because by definition the ciphertext must have no utility to the eavesdropper.

[1] In this respect the work on privacy diverges from the literature on secure function evaluation, where privacy is ensured only modulo the function to be computed: if the function is inherently disclosive then privacy is abandoned.

[2] Semantic security against an eavesdropper says that nothing can be learned about a plaintext from the ciphertext that could not be learned without seeing the ciphertext.

In this paper we prove the impossibility result, define differential privacy, and observe that the interactive techniques developed in a sequence of papers [8, 13, 3, 12] can achieve any desired level of privacy under this measure. In many cases very high levels of privacy can be ensured while simultaneously providing extremely accurate information about the database.

Related Work. There is an enormous literature on privacy in databases; we briefly mention a few fields in which the work has been carried out. See [1] for a survey of many techniques developed prior to 1989.

By far the most extensive treatment of disclosure limitation is in the statistics community; for example, in 1998 the *Journal of Official Statistics* devoted an entire issue to this question. This literature contains a wealth of privacy supportive techniques and investigations of their impact on the statistics of the data set. However, to our knowledge, rigorous definitions of privacy and modeling of the adversary are not features of this portion of the literature.

Research in the theoretical computer science community in the late 1970's had very specific definitions of privacy compromise, or what the adversary must achieve to be considered successful (see, eg, [9]). The consequent privacy guarantees would today be deemed insufficiently general, as modern cryptography has shaped our understanding of the dangers of the leakage of partial information. Privacy in databases was also studied in the security community. Although the effort seems to have been abandoned for over two decades, the work of Denning [7] is closest in spirit to the line of research recently pursued in [13, 3, 12].

The work of Agrawal and Srikant [2] and the spectacular privacy compromises achieved by Sweeney [18] rekindled interest in the problem among computer scientists, particularly within the database community. Our own interest in the subject arose from conversations with the philosopher Helen Nissenbaum.

2 Private Data Analysis: The Setting

There are two natural models for privacy mechanisms: interactive and non-interactive. In the non-interactive setting the data collector, a trusted entity, publishes a "sanitized" version of the collected data; the literature uses terms such as "anonymization" and "de-identification". Traditionally, sanitization employs techniques such as data perturbation and sub-sampling, as well as removing well-known identifiers such as names, birthdates, and social security numbers. It may also include releasing various types of synopses and statistics. In the interactive setting the data collector, again trusted, provides an interface through which users may pose queries about the data, and get (possibly noisy) answers.

Very powerful results for the interactive approach have been obtained ([13, 3, 12] and the present paper), while the non-interactive case has proven to be more difficult, (see [14, 4, 5]), possibly due to the difficulty of supplying utility that has not yet been specified at the time the sanitization is carried out. This intuition is given some teeth in [12], which shows concrete separation results.

3 Impossibility of Absolute Disclosure Prevention

The impossibility result requires some notion of utility – after all, a mechanism that always outputs the empty string, or a purely random string, clearly preserves privacy[3]. Thinking first about deterministic mechanisms, such as histograms or k-anonymizations [19], it is clear that for the mechanism to be *useful* its output should not be predictable by the user; in the case of randomized mechanisms the same is true, but the unpredictability must not stem only from random choices made by the mechanism. Intuitively, there should be a vector of questions (most of) whose answers *should* be learnable by a user, but whose answers are not in general known in advance. We will therefore posit a *utility vector*, denoted w. This is a binary vector of some fixed length κ (there is nothing special about the use of binary values). We can think of the utility vector as answers to questions about the data.

A *privacy breach* for a database is described by a Turing machine $\mathcal{C}$ that takes as input a description of a distribution $\mathcal{D}$ on databases, a database DB drawn according to this distribution, and a string – the purported privacy breach– and outputs a single bit[4]. We will require that $\mathcal{C}$ always halt. We say the adversary *wins*, with respect to $\mathcal{C}$ and for a given $(\mathcal{D}, DB)$ pair, if it produces a string s such that $\mathcal{C}(\mathcal{D}, DB, s)$ accepts. Henceforth "with respect to $\mathcal{C}$" will be implicit.

An auxiliary information generator is a Turing machine that takes as input a description of the distribution $\mathcal{D}$ from which the database is drawn as well as the database DB itself, and outputs a string, z, of auxiliary information. This string is given both to the adversary and to a *simulator*. The simulator has no access of any kind to the database; the adversary has access to the database via the privacy mechanism.

We model the adversary by a communicating Turing machine. The theorem below says that for any privacy mechanism San() and any distribution $\mathcal{D}$ satisfying certain technical conditions with respect to San(), there is always some particular piece of auxiliary information, z, so that z alone is useless to someone trying to win, while z in combination with access to the data through the privacy mechanism permits the adversary to win with probability arbitrarily close to 1. In addition to formalizing the entropy requirements on the utility vectors as discussed above, the technical conditions on the distribution say that learning the *length* of a privacy breach does not help one to guess a privacy breach.

Theorem 1. *Fix any privacy mechanism* San() *and privacy breach decider* $\mathcal{C}$*. There is an auxiliary information generator* $\mathcal{X}$ *and an adversary* $\mathcal{A}$ *such that for all distributions* $\mathcal{D}$ *satisfying Assumption 3 and for all adversary simulators* $\mathcal{A}^*$*,*

$$\Pr[\mathcal{A}(\mathcal{D}, \text{San}(\mathcal{D}, DB), \mathcal{X}(\mathcal{D}, DB)) \text{ wins}] - \Pr[\mathcal{A}^*(\mathcal{D}, \mathcal{X}(\mathcal{D}, DB)) \text{ wins}] \geq \Delta$$

where Δ *is a suitably chosen (large) constant. The probability spaces are over choice of* $DB \in_R \mathcal{D}$ *and the coin flips of* San*,* $\mathcal{X}$*,* $\mathcal{A}$*, and* $\mathcal{A}^*$*.*

[3] Indeed the height example fails in these trivial cases, since it is only through the sanitization that the adversary learns the average height.

[4] We are agnostic as to *how* a distribution $\mathcal{D}$ is given as input to a machine.

The distribution $\mathcal{D}$ completely captures any information that the adversary (and the simulator) has about the database, prior to seeing the output of the auxiliary information generator. For example, it may capture the fact that the rows in the database correspond to people owning at least two pets. Note that in the statement of the theorem all parties have access to $\mathcal{D}$ and may have a description of $\mathcal{C}$ hard-wired in; however, the adversary's strategy does not use either of these.

Strategy for $\mathcal{X}$ and $\mathcal{A}$ when all of w is learned from San(DB)*:* To develop intuition we first describe, slightly informally, the strategy for the special case in which the adversary always learns all of the utility vector, w, from the privacy mechanism[5]. This is realistic, for example, when the sanitization produces a histogram, such as a table of the number of people in the database with given illnesses in each age decile, or a when the sanitizer chooses a random subsample of the rows in the database and reveals the average ages of patients in the subsample exhibiting various types of symptoms. This simpler case allows us to use a weaker version of Assumption 3:

Assumption 2 *1.* $\forall\, 0 < \gamma < 1\ \exists n_\gamma\ \Pr_{DB \in_R \mathcal{D}}[|DB| > n_\gamma] < \gamma$*; moreover n_γ is computable by a machine given $\mathcal{D}$ as input.*

2. There exists an ℓ such that both *the following conditions hold:*
   (a) Conditioned on any privacy breach of length ℓ, the min-entropy of the utility vector is at least ℓ.
   (b) Every $DB \in \mathcal{D}$ has a privacy breach of length ℓ.

3. $\Pr[\mathcal{B}(\mathcal{D}, San(DB))$ wins$] \leq \mu$ *for all interactive Turing machines $\mathcal{B}$, where μ is a suitably small constant. The probability is taken over the coin flips of $\mathcal{B}$ and the privacy mechanism* San()*, as well as the choice of $DB \in_R \mathcal{D}$.*

Intuitively, Part (2a) implies that we can extract ℓ bits of randomness from the utility vector, which can be used as a one-time pad to hide any privacy breach of the same length. (For the full proof, ie, when not necessarily all of w is learned by the adversary/user, we will need to strengthen Part (2a).) Let ℓ_0 denote the least ℓ satisfying (both clauses of) Part 2. We cannot assume that ℓ_0 can be found in finite time; however, for any tolerance γ let n_γ be as in Part 1, so all but a γ fraction of the support of $\mathcal{D}$ is strings of length at most n_γ. For any fixed γ it is possible to find an $\ell_\gamma \leq \ell_0$ such that ℓ_γ satisfies both clauses of Assumption 2(2) on all databases of length at most n_γ. We can assume that γ is hard-wired into all our machines, and that they all follow the same procedure for computing n_γ and ℓ_γ. Thus, Part 1 allows the more powerful order of quantifiersd in the statement of the theorem; without it we would have to let $\mathcal{A}$ and $\mathcal{A}^*$ depend on $\mathcal{D}$ (by having ℓ hard-wired in). Finally, Part 3 is a nontriviality condition.

The strategy for $\mathcal{X}$ and $\mathcal{A}$ is as follows. On input $DB \in_R \mathcal{D}$, $\mathcal{X}$ randomly chooses a privacy breach y for DB of length $\ell = \ell_\gamma$, if one exists, which occurs with probability at least $1 - \gamma$. It also computes the utility vector, w. Finally, it chooses a seed s and uses a strong randomness extractor to obtain from w

[5] Although this case is covered by the more general case, in which not all of w need be learned, it permits a simpler proof that exactly captures the height example.

an ℓ-bit almost-uniformly distributed string r [16, 17]; that is, $r = \mathrm{Ext}(s, w)$, and the distribution on r is within statistical distance ϵ from U_ℓ, the uniform distribution on strings of length ℓ, even given s and y. The auxiliary information will be $z = (s, y \oplus r)$.

Since the adversary learns all of w, from s it can obtain $r = \mathrm{Ext}(s, w)$ and hence y. We next argue that $\mathcal{A}^*$ wins with probability (almost) bounded by μ, yielding a gap of at least $1 - (\gamma + \mu + \epsilon)$.

Assumption 2(3) implies that $\Pr[\mathcal{A}^*(\mathcal{D}) \text{ wins}] \leq \mu$. Let d_ℓ denote the maximum, over all $y \in \{0,1\}^\ell$, of the probability, over choice of $DB \in_R \mathcal{D}$, that y is a privacy breach for DB. Since $\ell = \ell_\gamma$ does not depend on DB, Assumption 2(3) also implies that $d_\ell \leq \mu$.

By Assumption 2(2a), even conditioned on y, the extracted r is (almost) uniformly chosen, independent of y, and hence so is $y \oplus r$. Consequently, the probability that $\mathcal{X}$ produces z is essentially independent of y. Thus, the simulator's probability of producing a privacy breach of length ℓ for the given database is bounded by $d_\ell + \epsilon \leq \mu + \epsilon$, as it can generate simulated "auxiliary information" with a distribution within distance ϵ of the correct one.

The more interesting case is when the sanitization does not necessarily reveal all of w; rather, the guarantee is only that it always reveal a vector w' within Hamming distance κ/c of w for constant c to be determined[6]. The difficulty with the previous approach is that if the privacy mechanism is randomized then the auxiliary information generator may not know which w' is seen by the adversary. Thus, even given the seed s, the adversary may not be able to extract the same random pad from w' that the auxiliary information generator extracted from w. This problem is solved using *fuzzy extractors* [10].

Definition 1. *An* $(\mathcal{M}, m, \ell, t, \epsilon)$ fuzzy extractor *is given by procedures (*Gen,Rec*).*

1. Gen *is a randomized generation procedure. On input* $w \in \mathcal{M}$ *outputs an "extracted" string* $r \in \{0,1\}^\ell$ *and a public string* p*. For any distribution* W *on* $\mathcal{M}$ *of min-entropy* m*, if* $(R, P) \leftarrow \mathrm{Gen}(W)$ *then the distributions* (R, P) *and* (U_ℓ, P) *are within statistical distance* ϵ*.*
2. Rec *is a deterministic reconstruction procedure allowing recovery of* $r = R(w)$ *from the corresponding public string* $p = P(w)$ *together with any vector* w' *of distance at most* t *from* w*. That is, if* $(r, p) \leftarrow \mathrm{Gen}(w)$ *and* $||w - w'||_1 \leq t$ *then* $\mathrm{Rec}(w', p) = r$*.*

In other words, $r = R(w)$ looks uniform, even given $p = P(w)$, and $r = R(w)$ can be reconstructed from $p = P(w)$ and any w' sufficiently close to w.

We now strenthen Assumption 2(2a) to say that the entropy of the source $\mathrm{San}(W)$ (vectors obtained by interacting with the sanitization mechanism, all of distance at most κ/c from the true utility vector) is high even conditioned on any privacy breach y of length ℓ **and** $P = \mathrm{Gen}(W)$.

[6] One could also consider privacy mechanisms that produce good approximations to the utility vector with a certain probability for the distribution $\mathcal{D}$, where the probability is taken over the choice of $DB \in_R \mathcal{D}$ and the coins of the privacy mechanism. The theorem and proof hold *mutatis mutandis*.

Assumption 3 *For some ℓ satisfying Assumption 2(2b), for any privacy breach $y \in \{0,1\}^\ell$, the min-entropy of $(\mathrm{San}(W)|y)$ is at least $k+\ell$, where k is the length of the public strings p produced by the fuzzy extractor*[7].

Strategy when w need not be fully learned: For a given database DB, let w be the utility vector. This can be computed by $\mathcal{X}$, who has access to the database. $\mathcal{X}$ simulates interaction with the privacy mechanism to determine a "valid" w' close to w (within Hamming distance κ/c). The auxiliary information generator runs $\mathrm{Gen}(w')$, obtaining $(r = R(w'), p = P(w'))$. It computes n_γ and $\ell = \ell_\gamma$ (as above, only now satisfying Assumptions 3 and 2(2b) for all $DB \in D$ of length at most n_γ), and uniformly chooses a privacy breach y of length ℓ_γ, assuming one exists. It then sets $z = (p, r \oplus y)$.

Let w'' be the version of w seen by the adversary. Clearly, assuming $2\kappa/c \leq t$ in Definition 1, the adversary can reconstruct r. This is because since w' and w'' are both within κ/c of w they are within distance $2\kappa/c$ of each other, and so w'' is within the "reconstruction radius" for any $r \leftarrow \mathrm{Gen}(w')$. Once the adversary has reconstructed r, obtaining y is immediate. Thus the adversary is able to produce a privacy breach with probability at least $1 - \gamma$. It remains to analyze the probability with which the simulator, having access only to z but not to the privacy mechanism (and hence, not to any w'' close to w), produces a privacy breach.

In the sequel, we let $\mathcal{B}$ denote the *best* machine, among all those with access to the given information, at producing producing a privacy breach ("winning").

By Assumption 2(3), $\Pr[\mathcal{B}(\mathcal{D}, \mathrm{San}(DB)) \text{ wins}] \leq \mu$, where the probability is taken over the coin tosses of the privacy mechanism and the machine $\mathcal{B}$, and the choice of $DB \in_R \mathcal{D}$. Since $p = P(w')$ is computed from w', which in turn is computable from $\mathrm{San}(DB)$, we have

$$p_1 = \Pr[\mathcal{B}(\mathcal{D}, p) \text{ wins}] \leq \mu$$

where the probability space is now also over the choices made by Gen(), that is, the choice of $p = P(w')$. Now, let U_ℓ denote the uniform distribution on ℓ-bit strings. Concatenating a random string $u \in_R U_\ell$ to p cannot help $\mathcal{B}$ to win, so

$$p_2 = \Pr[\mathcal{B}(\mathcal{D}, p, u) \text{ wins}] = p_1 \leq \mu$$

where the probability space is now also over choice of u. For any fixed string $y \in \{0,1\}^\ell$ we have $U_\ell = U_\ell \oplus y$, so for all $y \in \{0,1\}^\ell$, and in particular, for all privacy breaches y of DB,

$$p_3 = \Pr[\mathcal{B}(\mathcal{D}, p, u \oplus y) \text{ wins}] = p_2 \leq \mu.$$

Let W denote the distribution on utility vectors and let $\mathrm{San}(W)$ denote the distribution on the versions of the utility vectors learned by accessing the

[7] A good fuzzy extractor "wastes" little of the entropy on the public string. Better fuzzy extractors are better for the adversary, since the attack requires ℓ bits of residual min-entropy after the public string has been generated.

database through the privacy mechanism. Since the distributions $(P, R) = \mathrm{Gen}(W')$, and (P, U_ℓ) have distance at most ϵ, it follows that for any $y \in \{0,1\}^\ell$

$$p_4 = \Pr[\mathcal{B}(\mathcal{D}, p, r \oplus y) \text{ wins}] \leq p_3 + \epsilon \leq \mu + \epsilon.$$

Now, p_4 is an upper bound on the probability that the *simulator* wins, given $\mathcal{D}$ and the auxiliary information $z = (p, r \oplus y)$, so

$$\Pr[\mathcal{A}^*(\mathcal{D}, z) \text{ wins}] \leq p_4 \leq \mu + \epsilon\,.$$

An $(\mathcal{M}, m, \ell, t, \epsilon)$ fuzzy extractor, where $\mathcal{M}$ is the distribution $\mathrm{San}(W)$ on utility vectors obtained from the privacy mechanism, m satisfies: for all ℓ-bit strings y which are privacy breaches for some database $D \in DB$, $\mathrm{H}_\infty(W'|y) \geq m$; and $t < \kappa/3$, yields a gap of at least

$$(1 - \gamma) - (\mu + \epsilon) = 1 - (\gamma + \mu + \epsilon)$$

between the winning probabilities of the adversary and the simulator. Setting $\Delta = 1 - (\gamma + \mu + \epsilon)$ proves Theorem 1.

We remark that, unlike in the case of most applications of fuzzy extractors (see, in particular, [10, 11]), in this proof we are not interested in hiding partial information about the source, in our case the approximate utility vectors W', so we don't care how much min-entropy is used up in generating p. We only require sufficient residual min-entropy for the generation of the random pad r. This is because an approximation to the utility vector revealed by the privacy mechanism is not itself disclosive; indeed it is by definition safe to release. Similarly, we don't necessarily need to maximize the tolerance t, although if we have a richer class of fuzzy extractors the impossibility result applies to more relaxed privacy mechanisms (those that reveal worse approximations to the true utility vector).

4 Differential Privacy

As noted in the example of Terry Gross' height, an auxiliary information generator with information about someone not even in the database can cause a privacy breach to this person. In order to sidestep this issue we change from absolute guarantees about disclosures to relative ones: any given disclosure will be, within a small multiplicative factor, just as likely whether or not the individual participates in the database. As a consequence, there is a nominally increased risk to the individual in participating, and only nominal gain to be had by concealing or misrepresenting one's data. Note that a bad disclosure can still occur, but our guarantee assures the individual that it will not be the presence of her data that causes it, nor could the disclosure be avoided through any action or inaction on the part of the user.

Definition 2. *A randomized function $\mathcal{K}$ gives* ϵ-differential privacy *if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$,*

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

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

This definition extends to group privacy as well. A collection of c participants might be concerned that their collective data might leak information, even when a single participant's does not. Using this definition, we can bound the dilation of any probability by at most $\exp(\epsilon c)$, which may be tolerable for small c. Note that we specifically aim to disclose aggregate information about large groups, so we should expect privacy bounds to disintegrate with increasing group size.

5 Achieving Differential Privacy

We now describe a concrete interactive privacy mechanism achieving ϵ-differential privacy[8]. The mechanism works by adding appropriately chosen random noise to the answer $a = f(X)$, where f is the *query function* and X is the database; thus the query functions may operate on the entire database at once. It can be simple – eg, "Count the number of rows in the database satisfying a given predicate" – or complex – eg, "Compute the median value for each column; if the Column 1 median exceeds the Column 2 median, then output a histogram of the numbers of points in the set S of orthants, else provide a histogram of the numbers of points in a different set T of orthants."

Note that the complex query above (1) outputs a vector of values and (2) is an adaptively chosen sequence of two vector-valued queries, where the choice of second query depends on the *true answer* to the first query. Although complex, it is soley a function of the database. We handle such queries in Theorem 4. The case of an adaptively chosen series of questions, in which subsequent queries depend on the *reported* answers to previous queries, is handled in Theorem 5. For example, suppose the adversary first poses the query "Compute the median of each column," and receives in response noisy versions of the medians. Let M be the reported median for Column 1 (so M is the true median plus noise). The adversary may then pose the query: "If M exceeds the true median for Column 1 (ie, if the added noise was positive), then ... else ..." This second query is a function not only of the database but also of the *noise* added by the privacy mechanism in responding to the first query; hence, it is adaptive to the behavior of the mechanism.

5.1 Exponential Noise and the $L1$-Sensitivity

We will achieve ϵ-differential privacy by the addition of random noise whose magnitude is chosen as a function of the largest change a single participant

[8] This mechanism was introduced in [12], where analagous results were obtained for the related notion of ϵ-indistinguishability. The proofs are essentially the same.

could have on the output to the query function; we refer to this quantity as the *sensitivity* of the function[9].

Definition 3. *For $f : \mathcal{D} \to R^d$, the* $L1$-sensitivity *of f is*

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

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

For many types of queries Δf will be quite small. In particular, the simple counting queries ("How many rows have property P?") have $\Delta f \leq 1$. Our techniques work best – ie, introduce the least noise – when Δf is small. Note that sensitivity is a property of the function alone, and is independent of the database.

The privacy mechanism, denoted $\mathcal{K}_f$ for a query function f, computes $f(X)$ and adds noise with a scaled symmetric exponential distribution with variance σ^2 (to be determined in Theorem 4) in each component, described by the density function

$$\Pr[\mathcal{K}_f(X) = a] \propto \exp(-\|f(X) - a\|_1/\sigma) \tag{3}$$

This distribution has independent coordinates, each of which is an exponentially distributed random variable. The implementation of this mechanism thus simply adds symmetric exponential noise to each coordinate of $f(X)$.

Theorem 4. *For $f : \mathcal{D} \to R^d$, the mechanism $\mathcal{K}_f$ gives $(\Delta f/\sigma)$-differential privacy.*

Proof. Starting from (3), we apply the triangle inequality within the exponent, yielding for all possible responses r

$$\Pr[\mathcal{K}_f(D_1) = r] \leq \Pr[\mathcal{K}_f(D_2) = r] \times \exp(\|f(D_1) - f(D_2)\|_1/\sigma)\ . \tag{4}$$

The second term in this product is bounded by $\exp(\Delta f/\sigma)$, by the definition of Δf. Thus (1) holds for singleton sets $S = \{a\}$, and the theorem follows by a union bound.

Theorem 4 describes a relationship between Δf, σ, and the privacy differential. To achieve ϵ-differential privacy, one must choose $\sigma \geq \epsilon/\Delta f$.

The importance of choosing the noise as a function of the sensitivity of the entire complex query is made clear by the important case of *histogram queries*, in which the domain of data elements is partitioned into some number k of classes, such as the cells of a contingency table of gross shoe sales versus geographic regions, and the true answer to the query is the k-tuple of the exact number of database points in each class. Viewed naïvely, this is a set of k queries, each of sensitivity 1, so to ensure ϵ-differential privacy it follows from k applications of

[9] It is unfortunate that the term *sensitivity* is overloaded in the context of privacy. We chose it in concurrence with *sensitivity analysis*.

Theorem 4 (each with $d = 1$) that it suffices to use noise distributed according to a symmetric exponential with variance k/ϵ in each component. However, for any two databases D_1 and D_2 differing in only one element, $||f(D_1) - f(D_2)||_1 = 1$, since only one cell of the histogram changes, and that cell only by 1. Thus, we may apply the theorem once, with $d = k$ and $\Delta f = 1$, and find that it suffices to add noise with variance $1/\epsilon$ rather than d/ϵ.

Adaptive Adversaries We begin with deterministic query strategies F specified by a set of query functions f_ρ, where $f_\rho(X)_i$ is the function describing the ith query given that the first $i-1$ (possibly vector-valued) responses have been $\rho_1, \rho_2, \ldots, \rho_{i-1}$. We require that $f_\rho(X)_i = f_{\rho'}(X)_i$ if the first $i-1$ responses in ρ and ρ' are equal. We define the sensitivity of a query strategy $F = \{f_\rho : \mathcal{D} \to (R^+)^d\}$ to be the largest sensitivity of any of its possible functions, ie: $\Delta F = \sup_\rho \Delta f_\rho$.

Theorem 5. *For query strategy $F = \{f_\rho : \mathcal{D} \to R^d\}$, the mechanism $\mathcal{K}_F$ gives $(\Delta F/\sigma)$-differential privacy.*

Proof. For each $\rho \in (R^+)^d$, the law of conditional probability says

$$\Pr[\mathcal{K}_F(X) = \rho] = \prod_{i \le d} \Pr[\mathcal{K}_F(X)_i = \rho_i | \rho_1, \rho_2, \ldots \rho_{i-1}] \tag{5}$$

With $\rho_1, \rho_2, \ldots, \rho_{i-1}$ fixed, $f_\rho(X)_i$ is fixed, and the distribution of $\mathcal{K}_F(X)_i$ is simply the random variable with mean $f_\rho(X)_i$ and exponential noise with variance σ^2 in each component. Consequently,

$$\Pr[\mathcal{K}_F(X) = \rho] \propto \prod_{i \le d} \exp(-\|f_\rho(X)_i - \rho_i\|_1/\sigma) \tag{6}$$

$$= \exp(-\|f_\rho(X) - \rho\|_1/\sigma) \tag{7}$$

As in Theorem 4, the triangle inequality yields $(\Delta F/\sigma)$-differential privacy.

The case of randomized adversaries is handled as usual, by fixing a "successful" coin sequence of a winning randomized strategy.

Acknowledgements Kobbi Nissim introduced me to the topic of interactive privacy mechanisms. The impossibility result is joint work with Moni Naor, and differential privacy was motivated by this result. The definition, the differential privacy mechanism, and Theorems 4 and 5 are joint work with Frank McSherry. The related notion of ϵ-indistinguishable privacy mechanisms was investigated by Kobbi Nissim and Adam Smith, who were the first to note that histograms of arbitrary complexity have low sensitivity. This example was pivotal in convincing me of the viability of our shared approach.

References

[1] N. R. Adam and J. C. Wortmann, Security-Control Methods for Statistical Databases: A Comparative Study, *ACM Computing Surveys* 21(4): 515-556 (1989).

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. ACM SIGMOD International Conference on Management of Data*, pp. 439–450, 2000.

[3] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: The SuLQ framework. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 128–138, June 2005.

[4] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee. Toward privacy in public databases. In *Proceedings of the 2nd Theory of Cryptography Conference*, pages 363–385, 2005.

[5] S. Chawla, C. Dwork, F. McSherry, and K. Talwar. On the utility of privacy-preserving histograms. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, 2005.

[6] T. Dalenius, Towards a methodology for statistical disclosure control. *Statistik Tidskrift 15*, pp. 429–222, 1977.

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

[8] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 202–210, 2003.

[9] D. Dobkin, A.K. Jones, and R.J. Lipton, Secure databases: Protection against user influence. *ACM Trans. Database Syst. 4*(1), pp. 97–106, 1979.

[10] Y. Dodis, L. Reyzin and A. Smith, Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Proceedings of EUROCRYPT 2004*, pp. 523–540, 2004.

[11] Y. Dodis and A. Smith, Correcting Errors Without Leaking Partial Information, In *Proceedings of the 37th ACM Symposium on Theory of Computing*, pp. 654–663, 2005.

[12] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Theory of Cryptography Conference*, pages 265–284, 2006.

[13] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Advances in Cryptology: Proceedings of Crypto*, pages 528–544, 2004.

[14] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 211–222, June 2003.

[15] S. Goldwasser and S. Micali, Probabilistic encryption. *Journal of Computer and System Sciences 28*, pp. 270–299, 1984; prelminary version appeared in *Proceedings 14th Annual ACM Symposium on Theory of Computing*, 1982.

[16] N. Nisan and D. Zuckerman. Randomness is linear in space. *J. Comput. Syst. Sci.*, 52(1):43–52, 1996.

[17] Ronen Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77:67–95, 2002.

[18] Sweeney, L., Weaving technology and policy together to maintain confidentiality. J Law Med Ethics, 1997. 25(2-3): p. 98-110.

[19] L. Sweeney, Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10* (5), 2002; 571-588.