A Firm Foundation for Private Data Analysis

Cynthia Dwork
Microsoft Research
dwork@microsoft.com

1. PRIVATE DATA ANALYSIS

In the information realm, loss of privacy is usually associated with failure to control access to information, to control the flow of information, or to control the purposes for which information is employed. *Differential privacy* arose in a context in which ensuring privacy is a challenge even if all these control problems are solved: privacy-preserving statistical analysis of data.

The problem of *statistical disclosure control* – revealing accurate statistics about a set of respondents while preserving the privacy of individuals – has a venerable history, with an extensive literature spanning statistics, theoretical computer science, security, databases, and cryptography (see, for example, the excellent survey [1], the discussion of related work in [2] and the *Journal of Official Statistics 9*(2), dedicated to confidentiality and disclosure control). This long history is a testament the importance of the problem. Statistical databases can be of enormous social value; they are used for apportioning resources, evaluating medical therapies, understanding the spread of disease, improving economic utility, and informing us about ourselves as a species.

The data may be obtained in diverse ways. Some data, such as census, tax, and other sorts of official data, are compelled; others are collected opportunistically, for example, from traffic on the internet, transactions on Amazon, and search engine query logs; other data are provided altruistically, by respondents who hope that sharing their information will help others to avoid a specific misfortune, or more generally, to increase the public good. Altruistic data donors are typically promised their individual data will be kept confidential – in short, they are promised "privacy." Similarly, medical data and legally compelled data, such as census data, tax return data, have legal privacy mandates. In our view, ethics demand that opportunistically obtained data should be treated no differently, especially when there is no reasonable alternative to engaging in the actions that generate the data in question.

The problems remain: even if data encryption, key management, access control, and the motives of the data curator are all unimpeachable, what does it mean to preserve privacy, and how can it be accomplished?

1.1 "How" is Hard

Let us consider a few common suggestions and some of the difficulties they can encouter.

Large Query Sets. One frequent suggestion is to disallow queries about a specific individual or small set of individuals. A well-known differencing argument demonstrates the inadequacy of the suggestion. Suppose it is known that Mr. X is in a certain medical database. Taken together, the answers to the two large queries "How many people in the database have the sickle cell trait?" and "How many people, not named X, in the database have the sickle cell trait?" yield the sickle cell status of Mr. X. The example also shows that encrypting the data, another frequent suggestion (oddly), would be of no help at all. The privacy compromise arises from correct operation of the database.

In *query auditing* each query to the database is evaluated in the context of the query history to determine if a response would be disclosive; if so, then the query is refused. For example, query auditing might be used to interdict the pair of queries about sickle cell trait just described. This approach is problematic for several reasons, among them that query monitoring is computationally infeasible [15] and that the refusal to respond to a query may itself be disclosive [14].

We think of a database as a collection of *rows*, with each row containing the data of a different respondent. In *subsampling* a subset of the rows is chosen at random and released. Statistics can then be computed on the subsample and, if the subsample is sufficiently large, these may be representative of the dataset as a whole. If the size of the subsample is very small compared to the size of the dataset, this approach has the property that every respondent is unlikely to appear in the subsample. However, this is clearly insufficient: Suppose appearing in a subsample has terrible consequences. Then every time subsampling occurs *some* individual suffers horribly.

In *input perturbation*, either the data or the queries are modified before a response is generated. This broad category encompasses a generalization of subsampling, in which the curator first chooses, based on a secret, random, function of the query, a subsample from the database, and then returns the result obtained by applying the query to the subsample [4]. A nice feature of this approach is that repeating the same query yields the same answer, while semantically equivalent but syntactially different queries are made on essentially unrelated subsamples. However, an outlier may may only be protected by the unlikelihood of being in the

subsample.

In what is traditionally called *randomized response*, the data themselves are randomized once and for all and statistics are computed from the noisy responses, taking into account in the distribution on the perturbation [22]. The term "randomized response" comes from the practice of having the respondents to a survey flip a coin and, based on the outcome, answering an invasive yes/no question or answering a more emotionally neutral one. In the computer science literature the choice governed by the coin flip is usually between honestly reporting one's value and responding randomly, typically by flipping a second coin and reporting the outcome. Randomized response was devised for the setting in which the individuals do not trust the curator, so we can think of the randomized responses as simply being published. Privacy comes from the uncertainty of how to interpret a reported value. The approach becomes untenable for complex data.

Adding random noise to the output has promise, and we will return to it later; here we point out that if done naïvely this approach will fail. To see this, suppose the noise has mean zero and that fresh randomness is used in generating every response. In this case, if the same query is asked repeatedly, then the responses can be averaged, and the true answer will eventually emerge. This is disastrous: an adversarial analyst could exploit this to carry out the difference attack described above. The approach cannot be "fixed" by recording each query and providing the same response each time a query is re-issued. There are several reasons for this. For example, syntactically different queries may be semantically equivalent, and, if the query language is sufficiently rich, then the equivalence problem itself is undecidable, so the curator cannot even test for this.

Problems with noise addition arise even when successive queries are completely unrelated to previous queries [5]. Let us assume for simplicity that the database consists of a single – but very sensitive – bit per person, so we can think of the database as an n-bit Boolean vector $d = (d_1, \ldots, d_n)$. This is an abstraction of a setting in which the database rows are quite complex, for example, they may be medical records, but the attacker is interested in one specific field, such as HIV status. The abstracted attack consists of issuing a string of queries, each described by a subset S of the database rows. The query is asking how many 1's are in the selected rows. Representing the query as the n-bit characteristic vector of the set S, with 1's in all the positions corresponding to rows in S and 0's everywhere else, the true answer to the query is the inner product $A(S) = \sum_{i=1}^{n} d_i S_i$. Suppose the privacy mechanism responds with $A(S)$ + random noise. How much noise is needed in order to preserve privacy?

Since we have not yet defined privacy, let us consider the easier problem of avoiding blatant "non-privacy," defined as follows: the system is blatantly non-private if an adversary can construct a candidate database that agrees with the real database D in, say, 99% of the entries. An easy consequence of the following theorem is that a privacy mechanism adding noise with magnitude always bounded by, say, $n/401$ is blatantly non-private against an adversary that can ask all 2^n possible queries [5]. There is nothing special about 401; any number exceeding 400 would work.

THEOREM 1. **[5]** *Let $\mathcal{M}$ be a mechanism that adds noise bounded by E. Then there exists an adversary that can reconstruct the database to within $4E$ positions.*

Blatant non-privacy with $E = n/401$ follows immediately from the theorem, as the reconstruction will be accurate in all but at most $4E = n \cdot \frac{4}{401} < n/100$ positions.

PROOF. Let d be the true database. The adversary can attack in two phases:

1. **Estimate the number of 1's in all possible sets:** Query $\mathcal{M}$ on all subsets $S \subseteq [n]$.
2. **Rule out "distant" databases:** For every candidate database $c \in \{0,1\}^n$, If, for any $S \subseteq [n]$, $|\sum_{i \in S} c_i - \mathcal{M}(S)| > E$, then rule out c. If c is not ruled out, then output c and halt.

Since $\mathcal{M}(S)$ never errs by more than E, the real database will not be ruled out, so this simple (but inefficient!) algorithm will output *some* database; let us call it c. We will argue that the number of positions in which c and d differ is at most $4 \cdot E$.

Let I_0 be the indices in which $d_i = 0$, that is, $I_0 = \{i \mid d_i = 0\}$. Similarly, define $I_1 = \{i \mid d_i = 1\}$. Since c was not ruled out, $|\mathcal{M}(I_0) - \sum_{i \in I_0} c_i| \leq E$. However, by assumption $|\mathcal{M}(I_0) - \sum_{i \in I_0} d_i| \leq E$. It follows from the triangle inequality that c and d differ in at most $2E$ positions in I_0; the same argument shows that they differ in at most $2E$ positions in I_1. Thus, c and d agree on all but at most $4E$ positions. □

What if we consider more realistic bounds on the number of queries? We think of $\sqrt{n}$ as an interesting threshold on noise, for the following reason: if the database contains n people drawn uniformly at random from a population of size $N \gg n$, and the fraction of the population satisfying a given condition is p, then we expect the number of rows in the database satisfying p to be roughly $np \pm \Theta(\sqrt{n})$, by the properties of the hypergeometric distribution. That is, the sampling error is on the order of $\sqrt{n}$. We would like that the noise introduced for privacy is smaller than the sampling error, ideally $o(\sqrt{n})$. Unfortunately, noise of magnitude $o(\sqrt{n})$ is blatantly non-private against a series of $n \log^2 n$ *randomly generated* queries [5], no matter the distribution on the noise. Several strengthenings of this pioneering result are now known. For example, if the entries in S are chosen independently according to a standard normal distribution, then blatant non-privacy continues to hold even against an adversary asking only $\Theta(n)$ questions, and even if more than a fifth of the responses have arbitrarily wild noise magnitudes, provided the other responses have noise magnitude $o(\sqrt{n})$ [8].

These are not just interesting mathematical exercises. We have been focussing on *interactive* privacy mechanisms, distinguished by the involvement of the curator in answering each query. In the *noninteractive* setting the curator publishes some information of arbitrary form, and the data are not used further. Research statisticians like to "look at the data," and we have frequently been asked for a method of generating a "noisy table" that will permit highly accurate answers to be derived for computations that are not specified at the outset. The noise bounds say this is impossible: no such table can safely provide very accurate answers to too many weighted subset sum questions; otherwise the table could be used in a simulation of the interactive mechanism,

and an attack could be mounted against the table. Thus, even if the analyst only requires the responses to a small number of unspecified queries, the fact that the table can be exploited to gain answers to other queries is problematic.

In the case of "internet scale" data sets, obtaining responses to, say, $n \geq 10^8$ queries is infeasible. What happens if the curator permits only a sublinear number of questions? This inquiry led to the first algorithmic results in differential privacy, in which it was shown how to maintain privacy against a sublinear number of *counting* queries, that is, queries of the form "How many rows in the database satisfy property P?" by adding noise of order $o(\sqrt{n})$ – less than the sampling error – to each answer [11]. The cumbersome privacy guarantee, which focused on the question of what an adversary can learn about a row in the database, is now known to imply a natural and still very powerful relaxation of differential privacy.

1.2 "What" is Hard

Newspaper horror stories about "anonymized" and "de-identified" data typically refer to non-interactive approaches in which certain kinds of information in each data record have been suppressed or altered. A famous example is AOL's release of a set of "anonoymized" search query logs. People search for many "obviously" disclosive things, such as their full names ("vanity searches"), their own social security numbers (to see if their numbers are publicly available on the web, possibly with a goal of detecting assess the threat of identity theft), and even the combination of mother's maiden name and social security number. AOL carefully redacted such obviously disclosive "personally identifiable information," and each user id was replaced by a random string. However, search histories can be very idiosyncratic, and a New York Times reporter correctly traced such an "anonymized" search history to a specific resident of Georgia.

In a *linkage attack*, released data are linked to other databases or other sources of information. We use the term *auxiliary information* to capture information about the respondents *other* than that which is obtained through the (interactive or non-interactive) statistical database. Any priors, beliefs, or information from newspapers, labor statistics, and so on, all fall into this category.

In a notable demonstration of the power of auxiliary information, medical records of the governor of Massachusetts were identified by linking voter registration records to "anonymized" Massachusetts Group Insurance Commission (GIC) medical encounter data, which retained the birthdate, sex, and zip code of the patient [21].

Despite this exemplary work, it has taken several years to fully appreciate the importance of taking auxiliary information into account in privacy-preserving data release. Sources and uses of auxiliary information are endlessly varied. As a final example, it has been proposed to modify search query logs by mapping *all* terms, not just the user ids, to random strings. In *token-based hashing* each query is tokenized, and then an uninvertible hash function is applied to each token. The intuition is that the hashes completely obscure the terms in the query. However, using a statistical analysis of the hashed log and *any* (unhashed) query log, for example, the released AOL log discussed above, the anonymization can be severely compromised, showing that token-based hashing is unsuitable for anonymization [16].

As we will see next, there are deep reasons for the fact that auxiliary information plays such a prominent role in these examples.

2. DALENIUS'S DESIDERATUM

In 1977 the statistician Tore Dalenius articulated an "*ad omnia*" (as opposed to *ad hoc*) privacy goal for statistical databases: anything that can be learned about a respondent from the statistical database should be learnable without access to the database. Although informal, this feels like the "right" direction. The breadth of the goal captures all the common intuitions for privacy. In addition, the definition only holds the database accountable for whatever "extra" is learned about an individual, beyond that which can be learned from other sources. In particular, an extrovert who posts personal information on the web may destroy her own privacy, and the database should not be held accountable.

Formalized, Dalenius' goal is strikingly similar to the gold standard for security of a cryptosystem against a passive eavesdropper, defined 5 years later. *Semantic security* captures the intuition that the encryption of a message reveals no information about the message. This is formalized by comparing the ability of a computationally efficient adversary, having access to both the ciphertext and any auxiliary information, to output (anything about) the plaintext, to the ability of a computationally efficient party having access *only* to the auxiliary information (and not the ciphertext), to achieve the same goal [12]. Abilities are measured by probabilities of success, where the probability space is over the random choices made in choosing the encryption keys, the ciphertexts, and by the adversaries. Clearly, if this difference is very, very tiny, then in a rigorous sense the ciphertext leaks (almost) no information about the plaintext.

The formal definition of semantic security is a pillar of modern cryptography. It is therefore natural to ask whether a similar property, such as Dalenius' goal, can be achieved for statistical databases. But there is an essential difference in the two problems. Unlike the eavesdropper on a conversation, the statistical database attacker is also a user, that is, a legitimate consumer of the information provided by the statistical database (not to mention the fact that she may also be a respondent in the database).

Many papers in the literature attempt to formalize Dalenius' goal (in some cases unknowingly) by requiring that the adversary's prior and posterior views about an individual (*i.e.*, before and after having access to the statistical database) shouldn't be "too different," or that access to the statistical database shouldn't change the adversary's views about any individual "too much." The difficulty with this approach is that if the statistical database teaches us anything at all, then it *should* change our beliefs about individuals. For example, suppose the adversary's (incorrect) prior view is that everyone has 2 left feet. Access to the statistical database teaches that almost everyone has one left foot and one right foot. The adversary now has a very different view of whether or not any given respondent has two left feet. But has privacy been compromised?

The last hopes for Dalenius' goal evaporate in light of the following parable, which again involves auxiliary information. Suppose we have a statistical database that teaches average heights of population subgroups, and suppose further that it is infeasible to learn this information (perhaps for financial reasons) in any other way (say, by conducting a

new study). Finally, suppose that one's true height is considered sensitive. Given the auxiliary information "Turing is two inches taller than the average Lithuanian woman," access to the statistical database teaches Turing's height. In contrast, anyone without access to the database, knowing only the auxiliary information, learns much less about Turing's height.

A rigorous impossibility result generalizes this argument, extending to essentially any notion of privacy compromise, *assuming the statistical database is useful.* The heart of the attack uses extracted randomness from the statistical database as a one-time pad for conveying the privacy compromise to the adversary/user [6].

Turing did not have to be a member of the database for the attack described above to be prosecuted against him. More generally, the things that statistical databases are designed to teach can, sometimes indirectly, cause damage to an individual, even if this individual is not in the database.

In practice, statistical databases are (typcially) created to provide some anticipated social gain; they teach us something we could not (easily) learn without the database. Together with the attack against Turing described above, and the fact that he did not have to be a member of the database for the attack to work, this suggests a new privacy goal: minimize the increased risk to an individual incurred by joining (or leaving) the database. That is, we move from comparing an adversary's prior and posterior views of an individual to comparing the risk to an individual when included in, versus when not included in, the database. This makes sense. A privacy guarantee that limits risk incurred by joining therefore encourages participation in the dataset, increasing social utility. This is the starting point on our path to *differential privacy.*

3. DIFFERENTIAL PRIVACY

Differential privacy will ensure that ability of an adversary to inflict harm (or good, for that matter) – of any sort, to any set of people – should be essentially the same, independent of whether any individual opts in to, or opts out of, the dataset. We will do this indirectly, simultaneously addressing all possible forms of harm and good, by focussing on the probability of any given output of a privacy mechanism and how this probability can change with the addition or deletion of any row. Thus, we will concentrate on pairs of databases (D, D') differing only in one row, meaning one is a subset of the other and the larger database contains just one additional row. Finally, to handle worst case pairs of databases, our probabilities will be over the random choices made by the privacy mechanism.

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

$$\Pr[\mathcal{K}(D) \in S] \quad \leq \quad \exp(\varepsilon) \times \Pr[\mathcal{K}(D') \in S], \qquad (1)$$

where the probability space in each case is over the coin flips of $\mathcal{K}$.

The multiplicative nature of the guarantee implies that an output whose probability is zero on a given database must also have probability zero on any neighboring database, and hence, by repeated application of the definition, on any other database. Thus, Definition 1 trivially rules out the subsample-and-release paradigm discussed above: for an individual x not in the dataset, the probability that x's data are sampled and released is obviously zero; the multiplicative nature of the guarantee ensures that the same is true for an individual whose data *are* in the dataset.

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

Definition 2 extends naturally to group privacy. repeated application of the definition bounds the ratios of probabilities of outputs when a collection C of participants opts in or opts out by a factor of $e^{|C|\varepsilon}$. Of course, the point of the statistical database is to disclose aggregate information about large groups (while simultaneously protecting individuals), so we should expect privacy bounds to disintegrate with increasing group size.

The parameter ε is public, and its selection is a social question. We tend to think of ε as, say, 0.01, 0.1, or in some cases, $\ln 2$ or $\ln\ 3$.

Sometimes, for example, in the census, an individual's participation is known, so hiding presence or absence makes no sense; instead we wish to hide the values in an individual's row. Thus, we can (and sometimes do) extend "differing in at most one row" to mean having symmetric difference at most 1 to capture both possibilities. However, we will continue to use Definition 2.

Returning to randomized response, we see that it yields ε-differential privacy for a value of ε that depends on the universe from which the rows are chosen and the probability with which a random, rather than non-random, value is contributed by the respondent. As an example, suppose each row consists of a single bit, and that the respondent's instructions are to first flip an unbiased coin to determine whether he will answer randomly or truthfully. If heads (respond randomly), then the respondent is to flip a second unbiased coin and report the outcome; if tails, the respondent answers truthfully. Fix $b \in \{0, 1\}$. If the true value of the input is b, then b is output with probability 3/4. On the other hand, if the true value of the input is $1 - b$, then b is output with probability 1/4. The ratio is 3, yielding $(\ln 3)$-differential privacy.

Suppose n respondents each employ randomized response independently, but using coins of known, fixed, bias. Then, given the randomized data, by the properties of the binomial distribution the analyst can appproximate the true answer to the question "How many respondents have value b?" to within an expected error on the order of $\Theta(\sqrt{n})$. As we will see, it is possible to do much better – obtaining *constant* expected error, independent of n.

Generalizing in a different direction, suppose each row now has two bits, each one randomized independently, as described above. While each bit remains $(\ln 3)$-differentially private, their logical-AND enjoys less privacy. That is, consider a privacy mechanism in which each bit is protected by this exact method of randomized response, and consider the

query: "What is the logical-AND of the bits in the row of respondent i (after randomization)?" If we consider the two extremes, one in which respondent i has data 11 and the other in which respondent i has data 00, we see that in the first case the probability of output 1 is 9/16, while in the second case the probability is 1/16. Thus, this mechanism is at best $(\ln 9)$-differentially private, not $\ln 3$. Again, it is possible to do much better, even while releasing the entire 4-element histogram, also known as a *contingency table*, with only constant expected error in each cell.

4. ACHIEVING DIFFERENTIAL PRIVACY

Achieving differential privacy revolves around hiding the presence or absence of a single individual. Consider the query "How many rows in the database satisfy property P?" The presence or absence of a single row can affect the answer by at most 1. Thus, a differentially private mechanism for a query of this type can be designed by first computing the true answer and then adding random noise according to a distribution with the following property:

$$\forall z, z' \; s.t. \; |z - z'| = 1 : \; \Pr[z] \leq e^{\varepsilon} \Pr[z']. \tag{2}$$

To see why this is desirable, consider any feasible response r. For any m, if m is the true answer and the response is r then then the random noise must have value $r - m$; similarly, if $m - 1$ is the true answer and the response is r, then the random noise must have value $r - m + 1$. In order for the response r to be generated in a differentially private fashion, it suffices for

$$e^{-\varepsilon} \leq \frac{\Pr[\text{noise} = r - m]}{\Pr[\text{noise} = r - m + 1]} \leq e^{\varepsilon}.$$

In general we are interested in vector-valued queries; for example, the data may be points in $\mathbf{R}^d$ and we wish to carry out an analysis that clusters the points and reports the location of the largest cluster.

DEFINITION 3. *[7]For $f : \mathcal{D} \to \mathbf{R}^d$, the L_1 sensitivity of f is*

$$\begin{aligned} \Delta f &= \max_{D, D'} \|f(D) - f(D')\|_1 \qquad (3) \\ &= \max_{D, D'} \sum_{i=1}^{d} |f(D)_i - f(D')_i| \end{aligned}$$

for all D, D' differing in at most one row.

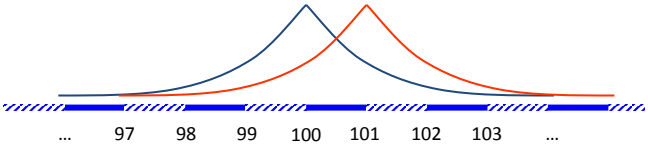
In particular, when $d = 1$ the sensitivity of f is the maximum difference in the values that the function f may take on a pair of databases that differ in only one row. This is the difference our noise must be designed to hide. For now, let us focus on the case $d = 1$.

The Laplace distribution with parameter b, denoted $\text{Lap}(b)$, has density function $P(z|b) = \frac{1}{2b} \exp(-|z|/b)$; its variance is $2b^2$. Taking $b = 1/\varepsilon$ we have that the density at z is proportional to $e^{-\varepsilon|z|}$. This distribution has highest density at 0 (good for accuracy), and for any z, z' such that $|z - z'| \leq 1$ the density at z is at most e^{ε} times the density at z', satisfying the condition in Equation 2. It is also symmetric about 0, and this is important. We cannot, for example, have a distribution that only yields non-negative noise. Otherwise the only databases on which a counting query could return a response of 0 would be databases in which no row satisfies the query. Letting D be such a database, and letting $D' = D \cup \{r\}$ for some row r satisfying the query, the pair D, D' would violate ε-differential privacy. Finally, the distribution gets flatter as ε decreases. This is correct: smaller ε means better privacy, so the noise density should be less "peaked" at 0 and change more gradually as the magnitude of the noise increases.

There is nothing special about the cases $d = 1, \Delta f = 1$:

THEOREM 4. *[7] For $f : \mathcal{D} \to \mathbf{R}^d$, the mechanism $\mathcal{K}$ that adds independently generated noise with distribution $\text{Lap}(\Delta f/\varepsilon)$ to each of the d output terms enjoys ε-differential privacy.*

Before proving the theorem, we illustrate the situation for the case of a counting query ($\Delta f = 1$) when $\varepsilon = \ln 2$ and the true answer to the query is 100. The distribution on the outputs (in gray) is centered at 100. The distribution on outputs when the true answer is 101 is shown in orange.



PROOF. (Theorem 4.) The proof is a simple generalization of the reasoning we used to illustrate the case of a single counting query.

Consider any subset $S \subseteq \text{Range}(\mathcal{K})$, and let D, D' be any pair of databases differing in at most one row. When the database is D, the probability density at any $r \in S$ is proportional to $\exp(-||f(D) - r||_1(\varepsilon/\Delta f))$. Similarly, when the database is D', the probability density at any $r \in \text{Range}(\mathcal{K})$ is proportional to $\exp(-||f(D') - r||_1(\varepsilon/\Delta f))$.

We have

$$\begin{aligned} \frac{e^{-||f(D)-r||(\varepsilon/\Delta f)}}{e^{-||f(D')-r||(\varepsilon/\Delta f)}} &= \frac{e^{||f(D')-r||(\varepsilon/\Delta f)}}{e^{||f(D)-r||(\varepsilon/\Delta f)}} \\ &= e^{(||f(D')-r||-||f(D)-r||)(\varepsilon/\Delta f)} \\ &\leq e^{(||f(D')-f(D)||)(\varepsilon/\Delta f)} \end{aligned}$$

where the inequality follows from the triangle inequality. By definition of sensitivity, $||f(D') - f(D)||_1 \leq \Delta f$, and so the ratio is bounded by $\exp(\varepsilon)$. Integrating over S yields ε-differential privacy. □

Given any query sequence $f_1, \ldots, f_m$, ε-differential privacy can be achieved by running $\mathcal{K}$ with noise distribution $\text{Lap}(\sum_{i=1}^{m} \Delta f_i/\varepsilon)$ on *each* query, even if the queries are chosen adaptively, with each successive query depending on the answers to the previous queries. In other words, by allowing the quality of each answer to deteriorate in a controlled way with the sum of the sensitivities of the queries, we can maintain ε-differential privacy.

With this in mind, let us return to some of the suggestions we considered earlier. Recall that using the specific randomized response strategy described above, for a single Boolean attribute, yielded error $\Theta(\sqrt{n})$ on databases of size n and $(\ln 3)$-differential privacy. In contrast, using Theorem 4 with the same value of ε, noting that $\Delta f = 1$ yields a variance of $2(1/\ln 3)^2$, or an expected error of $\sqrt{2}/\ln 3$. More generally, to obtain ε-differential privacy we get an expected error of $\sqrt{2}/\varepsilon$. Thus, our expected error magnitude is constant, independent of n.

What about two queries? The sensitivity of a sequence of two counting queries is 2. Applying the theorem with

$\Delta f/\varepsilon = 2/\varepsilon$, adding independently generated noise distributed as $\text{Lap}(2/\varepsilon)$ to each true answer yields ε-differential privacy. The variance is $2(2/\varepsilon)^2$, or standard deviation $2\sqrt{2}/\varepsilon$. Thus, for any desired ε we can achieve ε-differential privacy by increasing the expected magnitude of the errors as a function of the total sensitivity of the two-query sequence. This holds equally for

- Two instances of the *same query*, addressing the repeated query problem;
- One count for each of two different bit positions, for example, when each row consists of two bits;
- A pair of queries of the form: "How many rows satisfy property P?" and "How many rows satisfy property Q?" (where possibly $P = Q$); and
- An arbitrary pair of queries.

However, the theorem also shows we can sometimes do better. The logical-AND count we discussed above, even though it involves two different bits in each row, still only has sensitivity 1: the number of 2-bit rows whose entries are both 1 can change by at most one with the addition or deletion of a single row. Thus, this more complicated query can be answered in an ε-differentially private fashion using noise distributed as $\text{Lap}(1/\varepsilon)$; we don't need to use the distribution $\text{Lap}(2/\varepsilon)$.

Histogram Queries.

The power of Theorem 4 really becomes clear when considering *histogram queries*, defined as follows. If we think of the rows of the database as elements in a universe X, then a histogram query is a partitioning of X into an arbitrary number of disjoint regions $X_1, X_2, \ldots, X_d$. The implicit question posed by the query is: "For $i = 1, 2, \ldots, d$, how many points in the database are contained in X_i?" For example, the database may contain the annual income for each respondent, the query is a partitioning of incomes into ranges: $\{[0, 50K), [50K, 100K), \ldots, \geq 500K\}$. In this case $d = 11$, and the question is asking, for each of the d ranges, how many respondents in the database have annual income in the given range. This looks like d separate counting queries, but the entire query actually has sensitivity $\Delta f = 1$. To see this, note that if we remove one row from the database, then only one cell in the histogram changes, and that cell only changes by 1; similarly for adding a single row. So Theorem 4 says that ε-differential privacy can be maintained by perturbing each cell with an independent random draw from $\text{Lap}(1/\varepsilon)$. Returning to our example of two-bit rows, we can pose the 4-ary histogram query requesting, for each pair of literals v_1v_2, the number of rows with value v_1v_2, adding noise of order $1/\varepsilon$ to each of the four cells.

When Noise Makes No Sense.

There are times when the addition of noise for achieving privacy makes no sense. For example, the function f might map databases to strings, strategies, or trees, or it might be choosing the "best" among some specific, not necessarily continuous, set of real-valued objects. The problem of optimizing the output of such a function while preserving ε-differential privacy requires additional technology.

Assume the curator holds a database D and the goal is to produce an object y. The *exponential mechanism* [18] works as follows. We assume the existence of a *utility function u(D,y)* that measures the quality of an output y, given that the database is D. For example, the data may be a set of labeled points in $\mathbf{R}^d$ and the output y might be a d-ary vector describing a $(d-1)$-dimensional hyperplane that attempts to classify the points, so that those labeled with $+1$ have non-negative inner product with y and those labeled with -1 have negative inner product. In this case the utility would be the number of points correctly classified, so that higher utility corresponds to a better classifier. The exponential mechanism, $\mathcal{E}$, outputs y with probability proportional to $\exp(u(D, y)\varepsilon/\Delta u)$ and ensures ε-differential privacy. Here Δu is the sensitivity of the utility function bounding, for all adjacent databases (D, D') and potential outputs y, the difference $|u(D, y) - u(D', y)|$. In our example, $\Delta u = 1$. The mechanism assigns most mass to the best classifier, and the mass assigned to any other drops off exponentially in the decline in its utility for the current data set – hence the name "exponential mechanism."

When Sesitivity is Hard to Analyze.

The Laplace and exponential mechanisms provide a differentially private interface through which the analyst can access the data. Such an interface can be useful even when it is difficult to determine the sensitivity of the desired function or query sequence; it can also be used to run an iterative algorithm, composed of easily analyzed steps, for as many iterations as a given privacy budget permits. This is a powerful observation; for example, using only noisy sum queries, it is possible to carry out many standard datamining tasks, such as singular value decompositions, finding an ID3 decision tree, clustering, learning association rules, and learning anything learnable in the statistical queries learning model, frequently with good accuracy, in a privacy-preserving fashion [2]. This approach has been generalized to yield a publicly available codebase for writing programs that ensure differential privacy [17].

k-Means Clustering.

As an example of "private programming" [2], consider k-means clustering, described first in its usual, non-private form. The input consists of points $p_1, \ldots, p_n$ in the d-dimensional unit cube $[0, 1]^d$. Initial candidate means $\mu_1, \ldots, \mu_k$ are chosen randomly from the cube and updated as follows:

1. Partition the samples $\{p_i\}$ into k sets $S_1, \ldots, S_k$, associating each p_i with the nearest μ_j.
2. For $1 \leq j \leq k$, set $\mu'_j = \sum_{i \in S_j} p_i/|S_j|$, the mean of the samples associated with μ_j.

This update rule is typically iterated until some convergence criterion has been reached, or a fixed number of iterations have been applied.

Although computing the nearest mean of any one sample (Step 1) would breach privacy, we observe that to compute an average among an unknown set of points it is enough to compute their sum and divide by their number. Thus, the computation only needs to expose the approximate cardinalities of the S_j, not the sets themselves. Happily, the k candidate means implicitly define a histogram query, since they partition the space $[0, 1]^d$ according to their Voronoi cells, and so the vector $(|S_1|, \ldots, |S_k|)$ can be released with

very low noise in each coordinate. This gives us a differentially private approximation to the denominators in Step 2. As for the numerators, the sum of a subset of the p_i has sensitivity at most d, since the points come from the bounded region $[0,1]^d$. Even better, the sensitivity of the d-ary function that returns, for each of the k Voronoi cells, the d-ary sum of the points in the cell is at most d: adding or deleting a single d-ary point can affect at most one sum, and that sum can change by at most 1 in each of the d dimensions. Thus, using a query sequence with total sensitivity at most $d+1$, the analyst can compute a new set of candidate means by dividing, for each μ_j, the approximate sum of the points in S_j by the approximation to the cardinality $|S_j|$.

If we run the algorithm for a fixed number N of iterations we can use the noise distribution $\mathrm{Lap}((d+1)N/\varepsilon)$ to obtain ε-differentialy privacy. If we don't know the number of iterations in advance we can increase the noise parameter as the computation proceeds. There are many ways to do this. For example, we can answer in the first iteration with parameter $(d+1)(\varepsilon/2)$, in the next with parameter $(d+1)(\varepsilon/2)$, in the next with parameter $(d+1)(\varepsilon/4)$, and so on, each time using up half of the remaining "privacy budget."

5. GENERATING SYNTHETIC DATA

The idea of creating a synthetic data set whose statistics closely mirror those of the original data set, but which preserves privacy of individuals, was proposed in the statistics community no later than 1993 [20]. The lower bounds on noise discussed at the end of Section 1.1 imply that no such data set can safely provide very accurate answers to too many weighted subset sum questions, motivating the interactive approach to private data analysis discussed herein. Intuitively, the advantage of the interactive approach is that only the questions actually asked receive responses.

Against this backdrop, the non-interactive case was revisited from a learning theory perspective, challenging the interpretation of the noise lower bounds as a limit on the number of queries that can be answered privately [3]. This work, described next, has excited interest in solutions yielding noise in the range $[\omega(\sqrt{n}), o(n)]$.

Let X be a universe of data items and $\mathcal{C}$ be a *concept class* consisting of functions $c : X \to \{0,1\}$. We say $x \in X$ *satisfies* a concept $c \in C$ if and only if $c(x) = 1$. A concept class can be extremely general; for example, it might consist of all rectangles in the plane, or all Boolean circuits containing a given number of gates.

Given a sufficiently large database $D \in X^n$, it is possible to privately generate a synthetic database that maintains approximately correct fractional counts for *all* concepts in $\mathcal{C}$ (there may be infinitely many!). That is, letting S denote the synthetic database produced, with high probability over the choices made by the privacy mechanism, for every concept $c \in \mathcal{C}$, the fraction of elements in S that satisfy c is approximately the same as the fraction of elements in D that satisfy c.

The minimal size of the input database depends on the quality of the approximation, the logarithm of the cardinality of the universe X, the privacy parameter ε, and the *Vapnick-Chervonenkis dimension* of the concept class $\mathcal{C}$ (for finite $|C|$ this is at most $\log_2 |\mathcal{C}|$). The synthetic dataset, chosen by the exponential mechanism, will be a set of $m = O(\mathrm{VCdim}(\mathcal{C})/\gamma^2)$, elements in X (γ governs the maximum permissible inaccuracy in the fractional count.) Letting D denote the input dataset and $\hat{D}$ a candidate synthetic dataset, the utility function for the exponential mechanism is given by

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

6. PAN-PRIVACY

Data collected by a curator for a given purpose may be subject to "mission creep" and legal compulsion, such as a subpoena. Of course, we could analyze data and then throw it away, but can we do something even stronger, never storing the data in the first place? Can we strengthen our notion of privacy to capture the "never store" requirement?

These questions suggest an investigation of differentially private streaming algorithms with small state – much too small to store the data. However, nothing in the definition of a streaming algorithm, even one with very small state, precludes storing a few individual data points. Indeed, popular techniques from the streaming literature, such as Count-Min Sketch and subsampling, do precisely this. In such a situation, a subpoena or other intrusion into the local state will breach privacy.

A *pan-private* algorithm is private "inside and out," remaining differentially private even if its internal state becomes visible to an adversary [9]. To understand the pan-privacy guarantee, consider click stream data. These data are generated by individuals, and an individual may appear many times in the stream. Pan-privacy requires that any two streams differing only in the information of a single individual should produce very similar distributions on the *internal states* of the algorithm *and on its outputs*, even though the data of an individual are interleaved arbitrarily with other data in the stream.

As an example, consider the problem of *density estimation*. Assuming, for simplicity, that the data stream is just a sequence of IP addresses in a certain range, we wish to know what fraction of the set of IP addresses in the range actually appears in the stream. A solution inspired by randomized response can be designed using the following technique [9].

Define two probability distributions, D_0 and D_1, on the set $\{0,1\}$. D_0 assigns equal mass to zero and to one. D_1 has a slight bias towards 1; specifically, 1 has mass $1/2 + \varepsilon/4$, while 0 has mass $1/2 - \varepsilon/4$.

Let X denote the set of all possible IP addresses in the range of interest. The algorithm creates a table, with a one-bit entry b_x for each $x \in X$, initialized to an independent random draw from D_0. So initially the table is roughly half zeroes and half ones.

In an atomic step, the algorithm receives an element from the stream, changes state, and discards the element. When processing $x \in X$, the algorithm makes a fresh random draw from D_1, and stores the result in b_x. This is done no matter how many times x may have appeared in the past. Thus, for any x appearing at least once, b_x will be distributed according to D_1. However, if x never appears, then the entry for x is the bit drawn according to D_0 during the initialization of the table.

As with randomized response, the density in X of the items in the stream can be approximated from the number of 1's in the table, taking into account the expected fraction of "false positives" from the initialization phase and the "false negatives" when sampling from D_1. Letting θ denote the fraction of entries in the table with value 1, the output is

$4(\theta - 1/2)/\varepsilon + \mathrm{Lap}(1/\varepsilon|X|)$.

Intuitively, the internal state is differentially private because, for each $b \in \{0,1\}$, $e^{-\varepsilon} \leq \Pr_{\mathcal{D}_1}[b]/\Pr_{\mathcal{D}_0}[b] \leq e^{\varepsilon}$; privacy for the output is ensured by the addition of Laplacian noise. Over all, the algorithm is 2ε-differentially pan-private.

7. CONCLUSIONS

The differential privacy frontier is expanding rapidly, and there is insufficient space here to list all the interesting directions currently under investigation by the community. We identify a few of these.

The Geometry of Differential Privacy. Sharper upper and lower bounds on noise required for achieving differential privacy against a sequence of linear queries can be obtained by understanding the geometry of the query sequence [13]. In some cases dependencies among the queries can be exploited by the curator to markedly improve the accuracy of the responses. Generalizing this investigation to the non-linear and interactive cases would be of significant interest.

Algorithmic Complexity. We have so far ignored questions of computational complexity. Many, but not all, of the techniques described here have efficient implementations. For example, there are instances of the synthetic data generation problem that, under standard cryptographic assumptions, have no polynomial time implementation [10]. It follows that there are cases in which the exponential mechanism has no efficient implementation. When can this powerful tool be implemented efficiently, and how?

An Alternative to Differential Privacy? Is there an alternative, "*ad omnia*," guarantee that composes automatically, and permits even better accuracy than differential privacy? Can cryptography be helpful in this regard [19]?

The work described herein has, for the first time, placed private data analysis on a strong mathematical foundation. The literature connects differential privacy to decision theory, economics, robust statistics, geometry, additive combinatorics, cryptography, complexity theory learning theory, and machine learning. Differential privacy thrives because it is natural, it is not domain-specific, and it enjoys fruitful interplay with other fields. This flexibility gives hope for a principled approach to privacy in cases, like private data analysis, where traditional notions of cryptographic security are inappropriate or impracticable.

8. REFERENCES

[1] N. R. Adam and J. Wortmann. Security-control methods for statistical databases: A comparative study. *ACM Computing Surveys*, 21:515–556, 1989.

[2] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: The SuLQ framework. In *Proc. 24th ACM Symposium on Principles of Database Systems*, pages 128–138, 2005.

[3] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *Proc. 40th ACM SIGACT Symposium on Thoery of Computing*, pages 609–618, 2008.

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

[5] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *Proc. 22nd ACM Symposium on Principles of Database Systems*, pages 202–210, 2003.

[6] C. Dwork. Differential privacy. In *Proc. 33rd International Colloquium on Automata, Languages and Programming (ICALP)(2)*, pages 1–12, 2006. See also: C. Dwork and M. Naor, On the difficulties of disclosure prevention in statistical databases, in submission.

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

[8] C. Dwork, F. McSherry, and K. Talwar. The price of privacy and the limits of lp decoding. In *Proc. 39th ACM Symposium on Theory of Computing*, pages pp. 85–94, 2007.

[9] C. Dwork, M. Naor, T. Pitassi, G. Rothblum, and S. Yekhanin. Pan-private streaming algorithms. In *Proc. 1st Symposium on Innovations in Computer Science*, 2010.

[10] C. Dwork, M. Naor, O. Reingold, G. Rothblum, and S. Vadhan. When and how can privacy-preserving data release be done efficiently? In *Proc. 41st International ACM Symposium on Theory of Computing*, pages 381–390, 2009.

[11] C. Dwork and K. Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Advances in Cryptology – CRYPTO'04*, pages 528–544, 2004.

[12] S. Goldwasser and S. Micali. Probabilistic encryption. *JCSS*, 28:270–299, 1984.

[13] M. Hardt and K. Talwar. On the geometry of differential privacy. arXiv:0907.3754v2, 2009.

[14] K. Kenthapadi, N. Mishra, and K. Nissim. Simulatable auditing. In *Proc. 24th ACM Symposium on Principles of Database Systems*, pages 118–127, 2005.

[15] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Auditing boolean attributes. In *Proc. 19th ACM Symposium on Principles of Database Systems*, pages 86–91, 2000.

[16] R. Kumar, J. Novak, B. Pang, and A. Tomkins. On anonymizing query logs via token-based hashing. In *Proc. WWW 2007*, pages 629–638, 2007.

[17] F. McSherry. Privacy integrated queries (codebase). available on Microsoft Research downloads website. See also pages 19-30, Proc. SIGMOD 2009.

[18] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Proc. 48th Annual Symposium on Foundations of Computer Science*, 2007.

[19] I. Mironov, O. Pandey, O. Reingold, and S. Vadhan. Computational differential privacy. In *Advances in Cryptology – CRYPTO'09*, pages 126–142, 2009.

[20] D. Rubin. Discussion: Statistical disclosure limitation. *Journal of Official Statistics*, 9:462–468, 1993.

[21] L. Sweeney. Weaving technology and policy together to maintain confidentiality. *J. Law Med. Ethics*, 25:98–110, 1997.

[22] S. Warner. Randomized response: a survey technique for eliminating evasive answer bias. *JASA*, pages 63–69, 1965.