SCIENTIFIC AMERICAN™

Permanent Address: **http://www.scientificamerican.com/article.cfm?id=privacy-by-the-numbers-a-new-approach-to-safeguarding-data**

Privacy by the Numbers: A New Approach to Safeguarding Data

A mathematical technique called "differential privacy" gives researchers access to vast repositories of personal data while meeting a high standard for privacy protection

By Erica Klarreich and Quanta Magazine | Monday, December 31, 2012 | 16 comments

From Simons Science News (find original story here)

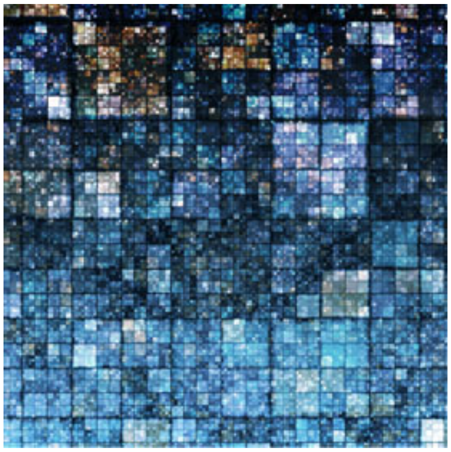
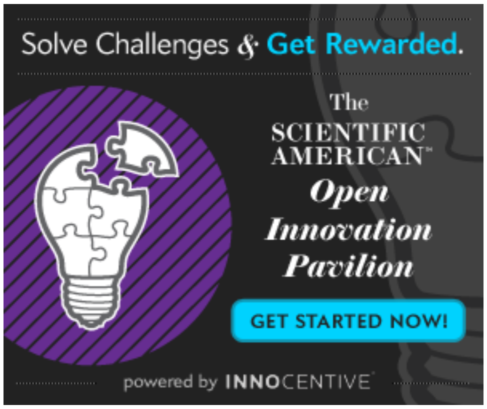


Image: DARPA



In 1997, when Massachusetts began making health records of state employees available to medical researchers, the government removed patients' names, addresses, and Social Security numbers. William Weld, then the governor, assured the public that identifying individual patients in the records would be impossible.

Within days, an envelope from a graduate student at the Massachusetts Institute of Technology arrived at Weld's office. It contained the governor's health records.

Although the state had removed all obvious identifiers, it had left each patient's date of birth, sex and ZIP code. By cross-referencing this information with voter-registration records, Latanya Sweeney was able to pinpoint Weld's records.

Sweeney's work, along with other notable privacy breaches over the past 15 years, has raised questions about the security of supposedly anonymous information.

"We've learned that human intuition about what is private is not especially good," said Frank McSherry of Microsoft Research Silicon Valley in Mountain View, Calif. "Computers are getting more and more sophisticated at pulling individual data out of things that a naive person might think are harmless."

As awareness of these privacy concerns has grown, many organizations have clamped down on their sensitive data, uncertain about what, if anything, they can release without jeopardizing the privacy of individuals. But this attention to privacy has come at a price, cutting researchers off from vast repositories of potentially invaluable data.

Medical records, like those released by Massachusetts, could help reveal which genes increase the risk of developing diseases like Alzheimer's, how to reduce medical errors in hospitals or what treatments are most effective against breast cancer. Government-held information from Census Bureau surveys and tax returns could help economists devise policies that best promote income equality or economic growth. And data from social media websites like Facebook and Twitter could offer sociologists an unprecedented look at how ordinary people go about their lives.

The question is: How do we get at these data without revealing private information? A body of work a decade in the making is now starting to offer a genuine solution.

“Differential privacy,” as the approach is called, allows for the release of data while meeting a high standard for privacy protection. A differentially private data release algorithm allows researchers to ask practically any question about a database of sensitive information and provides answers that have been “blurred” so that they reveal virtually nothing about any individual’s data — not even whether the individual was in the database in the first place.

“The idea is that if you allow your data to be used, you incur no additional risk,” said Cynthia Dwork of Microsoft Research Silicon Valley. Dwork introduced the concept of differential privacy in 2005, along with McSherry, Kobbi Nissim of Israel’s Ben-Gurion University and Adam Smith of Pennsylvania State University.

Differential privacy preserves “plausible deniability,” as Avrim Blum of Carnegie Mellon University likes to put it. “If I want to pretend that my private information is different from what it really is, I can,” he said. “The output of a differentially private mechanism is going to be almost exactly the same whether it includes the real me or the pretend me, so I can plausibly deny anything I want.”

This privacy standard may seem so high as to be unattainable — and indeed, there is no useful differentially private algorithm that gives out *exactly* the same information regardless of whether it includes the real you or the pretend you. But if we allow algorithms that give out *almost* exactly the same information in the two cases, then useful and efficient algorithms do exist. This “almost” is a precisely calibrated parameter, a measurable quantification of privacy. Individuals or social institutions could decide what value of this parameter represents an acceptable loss of privacy, and then differentially private algorithms could be chosen that guarantee that the privacy loss is less than the selected parameter.

Privacy experts have developed a wide assortment of specialized differentially private algorithms to handle different kinds of data and questions about the data. Although much of this work is technical and difficult for nonexperts to penetrate, researchers are starting to build standardized computer languages that would allow nonexperts to release sensitive data in a differentially private way by writing a simple computer program.

One real-world application already uses differential privacy: a Census Bureau project called*OnTheMap*, which gives researchers access to agency data. Also, differential privacy researchers have fielded preliminary inquiries from Facebook and the federally funded iDASH center at the University of California, San Diego, whose mandate in large part is to find ways for researchers to share biomedical data without compromising privacy.

“Differential privacy is a promising and exciting technology,” said Aaron Roth, a computer scientist at the University of Pennsylvania.

Needle in a Haystack

It might seem that a simpler solution to the privacy problem would be to release only “aggregate” information — statements about large groups of people. But even this approach is susceptible to breaches of privacy.

Suppose you wanted to ascertain whether this writer has diabetes and you knew I belonged to a health database. You could find this out simply by subtracting the answers to two aggregate-level questions: “How many people in the database have diabetes?” and “How many people in the database not named Erica Klarreich have diabetes?”

Clearly, these two questions, when combined, violate my privacy. But it’s not always easy to spot which combinations of questions would constitute privacy breaches. Spotting such combinations is, in its full generality, what computer scientists call an “NP-hard” problem, which means that there is probably no efficient computer algorithm that could catch all such attacks.

And when the attacker has access to outside information about individuals in the database, extracting private information from aggregate statistics becomes even easier.

In 2008, a research team demonstrated the dangers of releasing aggregate information from genome-wide association studies, one of the primary research vehicles for uncovering links between diseases and particular genes. These studies typically involve sequencing the genomes of a test group of 100 to 1,000 patients who have the same disease and then calculating the average frequency in the group of something on the order of 100,000 different mutations. If a mutation appears in the group far more frequently than in the general population, that mutation is flagged as a possible cause or contributor to the disease.

The research team, led by Nils Homer, then a graduate student at the University of California at Los Angeles, showed that in many cases, if you know a person's genome, you can figure out beyond a reasonable doubt whether that person has participated in a particular genome-wide test group. After Homer's paper appeared, the National Institutes of Health reversed a policy, instituted earlier that year, that had required aggregate data from all NIH-funded genome-wide association studies to be posted publicly.

Perhaps even more surprisingly, researchers showed in 2011 that it is possible to glean personal information about purchases from Amazon.com's product recommendation system, which makes aggregate-level statements of the form, "Customers who bought this item also bought A, B and C." By observing how the recommendations changed over time and cross-referencing them with customers' public reviews of purchased items, the researchers were able in several cases to infer that a particular customer had bought a particular item on a particular day — even before the customer had posted a review of the item.

In all these cases, the privacy measures that had been taken seemed adequate, until they were breached. But even as the list of privacy failures ballooned, a different approach to data release was in the making, one that came with an a priori privacy guarantee. To achieve this goal, researchers had gone back to basics: Just what does it mean, they asked, to protect privacy?

Two-World Privacy

If researchers study a health database and discover a link between smoking and some form of cancer, differential privacy will not protect a public smoker from being labeled with elevated cancer risk. But if a person's smoking is a secret hidden in the database, differential privacy will protect that secret.

"'Differential' refers to the difference between two worlds — one in which you allow your sensitive data to be included in the database and one in which you don't," McSherry said. The two worlds cannot be made to work out exactly the same, but they can be made close enough that they are effectively indistinguishable. That, he said, is the goal of differential privacy.

Differential privacy focuses on information-releasing algorithms, which take in questions about a database and spit out answers — not exact answers, but answers that have been randomly altered in a prescribed way. When the same question is asked of a pair of databases (*A* and *B*) that differ only with regard to a single individual (Person *X*), the algorithm should spit out essentially the same answers.

More precisely, given any answer that the algorithm could conceivably spit out, the probability of getting that answer should be almost exactly the same for both databases; that is, the ratio of these two probabilities should be bounded by some number *R* close to 1. The closer *R* is to 1, the more difficult it will be for an attacker to figure out whether he is getting information about database *A* or database *B* and the better protected Person *X* will be. After all, if the attacker can't even figure out whether the information he is getting includes Person *X*'s data, he certainly can't figure out what Person *X*'s data is.

(Differential privacy researchers usually prefer to speak in terms of the logarithm of *R*, which they denote ε. This parameter puts a number on how much privacy leaks out when the algorithm is carried out: The closer ε is to 0, the better the algorithm is at protecting privacy.)

To get a sense of how differentially private algorithms can be constructed, let's look at one of the simplest such algorithms. It focuses on a scenario in which a questioner is limited to "counting queries"; for example: "How many people in the database have property P?"

Suppose the true answer to one such question is 157. The differentially private algorithm will "add noise" to the true answer; that is, before returning an answer, it will add or subtract from 157 some number, chosen randomly according to a predetermined set of probabilities. Thus, it might return 157, but it also might return 153, 159 or even 292. The person who asked the question knows which probability distribution the algorithm is using, so she has a rough idea of how much the true answer has likely been distorted (otherwise the answer the algorithm spat out would be completely useless to her). However, she doesn't know which random number the algorithm actually added.

The particular probability distribution being used must be chosen with care. To see what kind of distribution will ensure differential privacy, imagine that a prying questioner is trying to find out whether I am in a database. He asks, "How many people named Erica Klarreich are in the database?" Let's say he gets an answer of 100. Because Erica Klarreich is such a rare name, the questioner knows that the true answer is almost certainly either 0 or 1, leaving two possibilities:

(a) The answer is 0 and the algorithm added 100 in noise; or

(b) The answer is 1 and the algorithm added 99 in noise.

To preserve my privacy, the probability of picking 99 or 100 must be almost exactly the same; then the questioner will be unable to distinguish meaningfully between the two possibilities. More precisely, the ratio of these two probabilities should be at most the preselected privacy parameter *R*. And this should be the case with regard to not only 99 and 100 but also any pair of consecutive numbers; that way, no matter what noise value is added, the questioner won't be able to figure out the true answer.

A probability distribution that achieves this goal is the Laplace distribution, which comes to a sharp peak at 0 and gradually tapers off on each side. A Laplace distribution has exactly the property we need: There is some number *R* (called the width of the distribution) such that for any two consecutive numbers, the ratio of their probabilities is *R*.

There is one Laplace distribution for each possible width; thus, we can tinker with the width to get the Laplace distribution that gives us the exact degree of privacy we want. If we need a high level of privacy, the corresponding distribution will be comparatively wide and flat; numbers distant from 0 will be almost as probable as numbers close to 0, ensuring that the data are blurred by enough noise to protect privacy.

Inevitably, tension exists between privacy and utility. The more privacy you want, the more Laplace noise you have to add and the less useful the answer is to researchers studying the database. With a Laplace distribution, the expected amount of added noise is the reciprocal of Ɛ; so, for example, if you have chosen a privacy parameter of 0.01, you can expect the algorithm's answer to be blurred by about 100 in noise.

The larger the dataset, the less a given amount of blurring will affect utility: Adding 100 in noise will blur an answer in the hundreds much more than an answer in the millions. For datasets on the scale of the Internet — that is, hundreds of millions of entries — the algorithm already provides good enough accuracy for many practical settings, Dwork said.

And the Laplace noise algorithm is only the first word when it comes to differential privacy. Researchers have come up with a slew of more sophisticated differentially private algorithms, many of which have a better utility-privacy trade-off than the Laplace noise algorithm in certain situations.

"People keep finding better ways of doing things, and there is still plenty more room for improvement," Dwork said. When it comes to more moderate-sized datasets than the Internet, she said, “there are starting to be algorithms out there for many tasks.”

With a differentially private algorithm, there’s no need to analyze a question carefully to determine whether it seeks to invade an individual’s privacy; that protection is automatically built into the algorithm’s functioning. Because prying questions usually boil down to small numbers related to specific people and non-prying questions examine aggregate-level behavior of large groups, the same amount of added noise that renders answers about individuals meaningless will have only a minor effect on answers to many legitimate research questions.

With differential privacy, the kinds of issues that plagued other data releases — such as attackers cross-referencing data with outside information — disappear. The approach’s mathematical privacy guarantees do not depend on the attacker having limited outside information or resources.

“Differential privacy assumes that the adversary is all-powerful,” McSherry said. “Even if attackers were to come back 100 years later, with 100 years’ worth of thought and information and computer technology, they still would not be able to figure out whether you are in the database. Differential privacy is future-proofed.”

A Fundamental Primitive

So far, we have focused on a situation in which someone asks a single counting query about a single database. But the real world is considerably more complex.

Researchers typically want to ask many questions about a database. And over your lifetime, snippets of your personal information will probably find their way into many different databases, each of which may be releasing data without consulting the others.

Differential privacy provides a precise and simple way to quantify the cumulative privacy hit you sustain if researchers ask multiple questions about the databases to which you belong. If you have sensitive data in two datasets, for example, and the curators of the two datasets release those data using algorithms whose privacy parameters are Ɛ1 and Ɛ2, respectively, then the total amount of your privacy that has leaked out is at most Ɛ1+ Ɛ2. The same additive relationship holds if a curator allows multiple questions about a single database. If researchers ask m questions about a database and each question gets answered with privacy parameter Ɛ, the total amount of privacy lost is at most mƐ.

So, in theory, the curator of a dataset could allow researchers to ask as many counting queries as he wishes, as long as he adds enough Laplace noise to each answer to ensure that the total amount of privacy that leaks out is less than his preselected privacy “budget.”

And although we have limited our attention to counting queries, it turns out that this restriction is not very significant. Many of the other question types that researchers like to ask can be recast in terms ofcounting queries. If you wanted to generate a list of the top 100 baby names for 2012, for example, you could ask a series of questions of the form, “How many babies were given names that start with A?” (or Aa, Ab or Ac), and work your way through the possibilities.

“One of the early results in machine learning is that almost everything that is possible in principle to learn can be learned through counting queries,” Roth said. “Counting queries are not isolated toy problems, but a fundamental primitive” — that is, a building block from which many more complex algorithms can be built.

But there’s a catch. The more questions we want to allow, the less privacy each question is allowed to use up from the privacy budget and the more noise has to be added to each answer. Consider the baby names question. If we decide on a total privacy budget Ɛ of 0.01 and there are 10,000 names to ask about, each question’s individual privacy budget is only Ɛ/10,000, or 0.000001. The expected amount of noise added to each answer will be 10,000/Ɛ, or 1,000,000 — an amount that will swamp the true answers.

In other words, the naive approach of adding Laplace noise to each question independently is limited in terms of the number of questions to which it can provide useful answers. To deal with this, computer scientists have developed an arsenal of more powerful primitives — algorithmic building blocks which, by taking into account the particular structure of a database and problem type, can answer more questions with more accuracy than the naive approach can.

For example, In 2005, Smith noticed that the baby names problem has a special structure: removing one person's personal information from the database changes the answer for only one of the 10,000 names in the database. Because of this attribute, we can get away with adding only 1/Ɛ in Laplace noise to each name answer, instead of 10,000/Ɛ, and the outcome will stay within our Ɛ privacy budget. This algorithm is a primitive that can be applied to any "histogram" query — that is, one asking how many people fall into each of several mutually exclusive categories, such as first names.

When Smith told Dwork about this insight in the early days of differential privacy research, "something inside me went, 'Wow!'" Dwork said. "I realized that we could exploit the structure of a query or computation to get much greater accuracy than I had realized."

Since that time, computer scientists have developed a large library of such primitives. And because the additive rule explains what happens to the privacy parameter when algorithms are combined, computer scientists can assemble these building blocks into complex structures while keeping tabs on just how much privacy the resulting algorithms use up.

"One of the achievements in this area has been to come up with algorithms that can handle a very large number of queries with a relatively small amount of noise," said Moritz Hardt of IBM Research Almadenin San Jose, Calif.

To make differential privacy more accessible to nonexperts, several groups are working to create a differential privacy programming language that would abstract away all the underlying mathematics of the algorithmic primitives to a layer that the user doesn't have to think about.

"If you're the curator of a dataset, you don't have to worry about what people are doing with your dataset as long as they are running queries written in this language," said McSherry, who has created one preliminary such language, calledPINQ. "The program serves as a proof that the query is OK."

A Nonrenewable Resource

Because the simple additive Ɛ rule gives a precise upper limit on how much total privacy you lose when the various databases you belong to release information in a differentially private way, the additive rule turns privacy into a "fungible currency," McSherry said.

For example, if you were to decide how much total lifetime privacy loss would be acceptable to you, you could then decide how you want to "spend" it — whether in exchange for money, perhaps, or to support a research project you admire. Each time you allowed your data to be used in a differentially private data release, you would know exactly how much of your privacy budget remained.

Likewise, the curator of a dataset of sensitive information could decide how to spend whatever amount of privacy she had decided to release — perhaps by inviting proposals for research projects that would describe not only what questions the researchers wanted to ask and why, but also how much privacy the project would use up. The curator could then decide which projects would make the most worthwhile use of the dataset's predetermined privacy budget. Once this budget had been used up, the dataset could be closed to further study.

"Privacy is a nonrenewable resource," McSherry said. "Once it gets consumed, it is gone."

The question of which value of Ɛ represents an acceptable privacy loss is ultimately a problem for society, not for computer scientists — and each person may give a different answer. And although the prospect of putting a price on something as intangible as privacy may seem daunting, a relevant analog exists.

“There’s another resource that has the same property — the hours of your life,” McSherry said. “There are only so many of them, and once you use them, they’re gone. Yet because we have a currency and a market for labor, as a society we have figured out how to price people’s time. We could imagine the same thing happening for privacy.”

Reprinted with permission from Simons Science News, an editorially independent division of SimonsFoundation.org *whose mission is to enhance public understanding of science by covering research developments and trends in mathematics and the computational, physical and life sciences.*