Differential Privacy: The Pursuit of Protections by Default

case study

A DISCUSSION WITH MIGUEL GUEVARA, DAMIEN DESFONTAINES, JIM WALDO, AND TERRY COATTA

Over the past decade, calls for better measures to protect sensitive, personally identifiable information have blossomed into what politicians like to call a "hot-button issue." Certainly, privacy violations have become rampant and people have grown keenly aware of just how vulnerable they are. When it comes to potential remedies, however, proposals have varied widely, leading to bitter, politically charged arguments. To date, what's chiefly come of that have been bureaucratic policies that satisfy almost no one—and infuriate many.

Now, into this muddled picture comes differential privacy. First formalized in 2006, it's an approach based on a mathematically rigorous definition of privacy that allows formalization and proof of the guarantees against re-identification offered by a system. While differential privacy has been accepted by theorists for some time, its implementation has turned out to be subtle and tricky, with practical applications only now starting to become available. To date, differential privacy has been adopted by the U.S. Census Bureau, along with a number of technology companies, but what this means and how these organizations have implemented their systems remains a mystery to many.

It's also unlikely that the emergence of differential privacy signals an end to all the difficult decisions and tradeoffs, but it does signify that there now are measures of privacy that

can be quantified and reasoned about—and then used to apply suitable privacy protections.

A milestone in the effort to make this capability generally available came in September 2019 when Google released an open-source version of the differential privacy library that the company has used with many of its core products.

In the exchange that follows, two of the people at Google who were central to the effort to release the library as open source—Damien Desfontaines, privacy software engineer; and Miguel Guevara, who leads Google's differential privacy product development effort—reflect on the engineering challenges that lie ahead, as well as what remains to be done to achieve their ultimate goal of providing privacy protection by default. They are joined in this discussion by Jim Waldo, Harvard's CTO who recently co-chaired a National Academies study on privacy, and Terry Coatta, the CTO of Marine Learning Systems.

JIM WALDO I'd love to hear how you characterize differential privacy, since most of the descriptions I've heard so far are either so loose as to be meaningless or so formal as to be difficult to follow.

MIGUEL GUEVARA I think about it in the context of other privacy technologies, many of which are policy- and heuristics-driven. That can make you feel good, but it's very hard to reason about a lot of those technologies, whereas differential privacy gives you a tangible way to reason about what's happening with the privacy of the underlying data and to quantify how much privacy has been lost there.

Having that ability is powerful for data curators. It also allows us to imagine a world where users possess

that same sort of control over their own data and, by way of some adjustments to their applications, will have the ability to determine how much privacy they can have. So, the basic idea behind differential privacy is to give individuals the ability to make these sorts of decisions in a rational and informed manner.

JW That's a nice way to characterize the goals of differential privacy. But now I'm going to ask you to get a little more concrete and talk about how you intend to meet those goals.

DAMIEN DESFONTAINES What's most characteristic about differential privacy is that when you generate statistics—that is, some aggregated information about a set of people—you purposely add noise to the results of that computation. This is how you attain the guarantee of differential privacy: by ensuring that someone looking at the results of that computation will not be able to get information about the individuals whose data has been included as part of the dataset.

What I mean by *noise* simply has to do with sampling a random number of data points from a distribution. Ideally, that random number can be kept quite small—between -10 and 10 for a count, for example. For statistics on a larger scale, the noise you add should not greatly impact the quality of your data. Then, as Miguel indicated, differential privacy also lets you quantify the tradeoffs between privacy and precision for a dataset. The amount of noise you add to the data is what allows you to quantify just how private the dataset will be. Which is to say, the more noise you add, the less precise your statistics will be. At the same time, your privacy guarantees will also become that much

stronger.

JW So, the core idea is that when you query the data, the answer has some noise added to it, and this gives you control over privacy because the more noise you add to the data, the more private it becomes—with the tradeoff being that the amount of precision goes down as the noise goes up.

DD That's right.

JW How is this now being used inside Google?

MG It's mostly used by a lot of internal tools. From the start, we saw it as a way to build tooling that could be used to address some core internal use cases. The first of those was a project where we helped some colleagues who wanted to do some rapid experimentation with data. We discovered that, much of the time, a good way to speed access to the data underlying a system is to add a privacy layer powered by differential privacy. That prompted us to build a system that lets people query underlying data and obtain differentially private results.

Differential privacy made it possible for Google to produce the COVID-19 Community Mobility Reports.

After we started to see a lot of success there, we decided to scale that system—to the point where we're now building systems capable of dealing with data volumes at Google scale, while also finding ways to serve end users, as well as internal ones. For example, differential privacy made it possible for Google to produce the COVID-19 Community Mobility Reports [used by public health officials to obtain aggregated, anonymized insights from health-care data that can then be used to chart disease movement trends over time by geography as well as by locales (such as grocery stores, transit stations, and workplaces)]. There's also a business feature in Google

Maps that shows you how busy a place is at any given point in time. Differential privacy makes that possible as well. Basically, differential privacy is used by infrastructure systems at Google to enable both internal analysis and some number of end-user features.

JW As I understand it, there's a third variable. There's how accurate things are and how much noise you add—and then there's the number of queries you allow. Do you take all three of those into account?

MG It really depends on the system. In theory, you can have an infinite number of queries. But there's a critical aspect of differential privacy called the privacy budget—each time you use a query, you use some part of your budget. So, let's say that every time you issue a query, you use half of your remaining budget. As you continue to issue more queries, the amount of noise you introduce into your queries will just increase.

With one of our early systems, we overcame this by doing something you're hinting at, which was to limit the number of queries users could make. That was so we wouldn't exhaust the budget too fast and would still have what we needed to provide meaningful results for our users.

DD There's also a question that comes up in the literature having to do with someone using an engine to run arbitrary queries over a dataset—typically whenever that person does not have access to the raw data. In such use cases, budget tracking becomes very important. Accordingly, we've developed systems with this in mind, using techniques like sampling, auditing, and limiting the number of queries that can be run. On the other hand, with many common applications, you know what kind of query you

want to run on the data: For a busyness graph displayed on Google Maps, for example, a handful of predetermined queries might be used daily to generate the required data so you don't have to provide a higher privacy budget for future queries as yet unknown. Instead, you'll know in advance which queries are going to be issued, so you'll also know how much noise needs to be added.

TERRY COATTA It seems a corollary of this might be: If you have a dataset against which you intend to perform ad hoc queries but don't know in advance what the nature of those queries might be, differential privacy in some sense limits the utility of that dataset. That is, there are only so many ad hoc queries that can be served before you've effectively exhausted your ability to query anymore against that dataset.

MG OK, but I guess I would frame this in terms of use cases. What we've discovered is that when you look at the sorts of use cases you're suggesting, people tend to be interested in looking only at broad statistical trends. Say some company just introduced its product in Country X and now wants to see how many users are using operating system 1 versus operating system 2. At that level, differential privacy provides really good results from a statistical perspective.

But then there's another use case, which is what I believe Damien was talking about. Let's say that, for this same example, you discover that the critical variables for your analysis happen to be country, age, and income. You can just set up a query accordingly and then run that every day or every few days without consuming any additional privacy budget simply because you're going to be using

those data points only once every so many days.

JW It seems that many of the examples you're offering are gross in the sense that there are fairly large numbers of entities in the datasets on one side or the other of a comparison—meaning that adding a small amount of noise really shouldn't cause an issue. But I wonder about queries around outliers. Say, if I wanted to find the number of people in some particular country who were still running Windows XP or maybe were still using OS/2, a little bias in those numbers would probably cause a real difference in the outcomes. When do you think it's appropriate—or inappropriate—to use a query that is differentially private?

MG In general, I think differential privacy is very good for describing broad statistical trends in terms of how thousands of people do X things each day. The Community Mobility Reports that Google has been producing to track COVID-19 infection trends is a good example. There are other use cases where you can look at some very particular abuse or spam trends indicating specific attack vectors. If you end up doing some very granular queries on that, you'll find that—while it's theoretically possible to accomplish this with differential privacy—the relative impact of the noise will be so huge that the results you'll get will be almost useless.

As a general rule, I'd say that while differential privacy is good for doing broad population analysis, it's not so good at figuring out how one or two people are behaving since, by definition, that's the very thing differential privacy is designed to protect against.

TC A couple of times already we've made reference to the amount of privacy that might be "lost," whereas the layman

concept of privacy is more Boolean—that is, it's either private or not. So, it's interesting to talk about it here as a quantitative measure. What does that actually mean?

DD The notion of privacy as something Boolean is misleading from the start. Even outside of differential privacy, you always need to ask yourself questions like: How can we make this feature work while collecting as little data as possible? What level of protection should we apply to the data we store? How can we request user consent in an understandable, respectful way? And so on.

The notion of privacy as something Boolean is misleading from the start.

None of these questions is Boolean. Even in adversarial contexts, where the answer *seems* to be Boolean, it isn't. For example: Is the attacker going to be able to intercept and re-identify data? The answer is either yes or no.

But you still need to think about other questions like: What is the attacker capable of? What are we trying to defend against? What's the worst-case scenario? This is to say, even without the formal concept of differential privacy, the notion of privacy in general is far from Boolean. There always are shades of gray.

What differential privacy does to achieve data anonymization is to quantify the tradeoffs in a formal, mathematical way. This makes it possible to move beyond these shades-of-gray assessments to apply a strong attack model where you have an attacker armed with arbitrary background knowledge and computational resources—which represents the worst possible case—and yet you're still able to get strong, quantifiable guarantees. That's the essence of differential privacy, and it's by far the best thing we have right now in terms of quantifying and measuring privacy against utility for data anonymization.

Powerful as differential privacy may be, it's also highly abstract. Getting users and developers alike to build confidence around its ability to protect personally identifiable information has proved to be challenging.

In an ongoing effort, various approaches are being tried to help people make the connection between the mathematics of differential privacy and the realization of actual privacy protection goals. Progress in this regard is not yet up to Google scale.

And yet, Google has a clear, vested interest in building public confidence in the notion that it and other large aggregators of user data are fully capable of provably anonymizing the data they utilize. Finding a way to convey that in a convincing manner to the general public, however, remains an unsolved problem.

JW When it comes to users who are worried about privacy, I doubt you'll be able to ease those concerns much by telling them you've set epsilon to some particular value. How do you translate the significance of that into something users can understand?

MG Honestly, I don't think we've done a great job of communicating this to users. We've been more focused on raising awareness. But this issue you raise is an important one since there are just so many misconceptions about anonymization out in the world right now. Many people believe that, to anonymize data, you just remove an entire identifier from a dataset. So, our first step is to make sure everyone realizes that does *not* qualify as proper or strong anonymization.

Then, once we get to that stage where users have personal privacy as their mindset, one of the biggest priorities for those of us in the privacy research community needs to become exactly what you're saying: How can we help people see the connection between what we're doing mathematically and what they've come to expect in terms of protections for their own personal privacy?

We've already done a bit of user research that has allowed us to really talk with users, and what I've learned is that, whenever we're able to show people how their personal data can be hidden behind the crowd and protected by random information, they definitely come around to expressing more confidence. Clearly, however, there's still a huge challenge ahead for us in terms of learning how to talk about these mathematical techniques and the guarantees they confer in ways that feel more tangible to end users.

DD The other side of this is that gaining a better understanding of the users' privacy concerns is part of what informs policy. Some of their questions are entirely orthogonal to the use of differential privacy. For example: Who among my family and friends and colleagues can see what I just shared online? How long will my data be kept?

When the time comes, we need to be able to offer differential privacy as an answer to the different, more specific question: How is my data protected whenever Google shares aggregated data publicly?

JW Maybe you ought to describe what you've developed for Google to make differential privacy a little easier for the average programmer to use.

MG The first critical thing to point out is that we've

developed a SQL engine that produces differential privacy results. The core idea behind that was, since a lot of analysts are already familiar with SQL, it would be best just to augment that syntax with a couple of differentially private operations. Essentially, that means someone can do an anon count and produce a differential privacy count from that and, similarly, do an anon sum and produce a differential privacy sum.

Some of the other pieces we've built are geared more toward a data-operation framework that processes a lot of data. You can think of them as Apache Beam-type frameworks that let us turn regular operations—primarily counts and sums—into differential privacy operations that teams then can use to produce their data in a manner that better protects privacy. [Apache Beam is an open-source, unified model for defining both batch and streaming data-parallel processing pipelines.]

JW How broadly is this used within Google and in what context?

DD Probably the most visible user-facing examples are a few features in Google Maps that are powered by differential privacy. Then there also are the COVID-19 Community Mobility Reports mentioned earlier. We use differential privacy internally as well to help analysts access data in a safe, anonymized way, and to power internal dashboards that let developers monitor how their products are being used. Basically, at a high level, any time a team wants to do something with sensitive data that calls for the data to be handled in an anonymous manner—for example, to retain the data longer so that data-protection requirements that might otherwise call

for encryption or tight access controls can instead be relaxed—we encourage them to use differential privacy.

TC But I can easily imagine users shooting themselves in the foot when using differential privacy. For example, I might issue some queries against the database, get some results back, and think I actually know what those results mean. What I might fail to recognize is that there's so much noise in those results that they actually don't mean anything at all. What does Google's differential privacy library do to help people avoid this trap?

MG Results that contain more noise than you realize can be a real problem from a usability perspective. In fact, one of the things our internal users continually ask us is: Where should we stop trusting the data?

Imagine that you issue a query and then get back a table that, say, gives you different counts. At some point, those counts will have more noise than real data in them. One way we try to address that is by providing confidence intervals in the results, with the hypothesis being that, if the confidence interval is very small relative to the value, then there's very little noise—meaning users can trust that result. If the confidence interval is very broad, then users can infer there's a lot of noise. And then, yes, they can stop trusting the data at that point.

DD In the specific use case of the COVID-19 Community Mobility Reports, which contain data that researchers and policymakers use to make hard decisions about social distancing and that sort of thing, we don't want them to derive the wrong conclusions from the data just because they don't really understand the noise-addition process. We did a couple of things to help avoid that. One is that

we decided to publish only the data where the confidence intervals seemed tight enough. That is, if the added noise had more than a 10 percent chance of leading to numbers that were more than 10 percent off, we didn't release that data. Instead, we'd say, "What we have isn't accurate enough, so no data is available for this metric."

The second thing we did was to document the whole process just as precisely as we could in a whitepaper that's been published online [Differentially Private SQL with Bounded User Contributions; https://arxiv.org/abs/1909.01917]. Referring to this, anyone doing complex statistical analyses on the data should have what they need to account for the uncertainty contributed by the noise.

JW Of course, any machine-learning algorithm also has a certain confidence interval. What is the relationship between the confidence intervals you're able to get out of a differentially private query on the data and what the machine-learning folks then manage to do with that data? Or have you not connected the two as yet?

DD There are various ways to combine differential privacy and machine learning, and we have a lot of researchers working on that very thing—in particular, by increasing the accuracy of machine-learning models while making them safe through the use of differential privacy. We've also published an open-source library [TensorFlow Privacy] that incorporates some of these techniques as part of training models for machine learning.

We're now experimenting to understand better how machine-learning models trained on sensitive datasets can inadvertently memorize information from the

original training data, while also working to see how differential privacy might be used to quantify that. One challenge is that the epsilon parameters we get by way of these methods are typically quite high, sometimes to the point where it's hard to interpret the relevant guarantees. Empirically, however, it also seems that even these difficult-to-interpret guarantees generally prove successful in mitigating attacks. Let's just say this is proving to be a fascinating and fruitful field of research.

TC Have you run into any complications in trying to combine differential privacy with other privacy-protecting technologies? I ask, since differential privacy clearly isn't going to solve all our problems.

MG I think we're just too early in our efforts to advance protections to know what all the possibilities are, but there are some encouraging signs. I've heard that some people are trying to use differential privacy with federated learning to train models in a provably private way. I've also heard that differential privacy is being used together with homomorphic encryption to share data between two parties such that both parties then can produce results that don't reveal any individual patterns or any group of patterns.

JW One of the interesting things I've observed about differential privacy is that there's been about a 10-year lag between the theoretical foundations and the first practical applications, which are only now becoming available. What has made this so difficult to implement?

DD We were quite surprised by some of the difficulties we encountered. Fundamentally, I don't think the math is all that hard. The basic results and techniques are relatively

simple, and it doesn't really take much time or effort to get a reasonable understanding of the theory behind them. But it turns out that transforming all that theory into practice has proved difficult and has required more time and thought than we anticipated.

There are a few reasons for this. One is that the literature makes some assumptions about the type of data you'd be looking to anonymize, and we discovered—in practice—that this is mostly wrong. An example is the assumption that each record of the dataset corresponds to a single user. This owes to the fact that the main use case presented in much of the literature relates to medical data—with one record per patient. But, of course, when you're working with datasets like logs of user activities, place visits, or search queries, each user ends up contributing much more than just a single record in the dataset. So, it took some innovations and optimizations to account for this in building some better tooling for our purposes.

In differential privacy theory, you add a random number from a continuous distribution to a statistic with arbitrary precision.

Something else that contributed to the unforeseen difficulties was that, even though the math is relatively simple, implementing it in a way that preserves the guarantees is tricky. It's a bit like RSA (Rivest-Shamir-Adleman) in cryptography—simple to understand, yet naïve implementations will encounter serious issues like timing attacks. In differential privacy theory, you add a random number from a continuous distribution to a statistic with arbitrary precision. To do that with a computer, you need to use floating-point numbers, and the ways these are represented come with a lot of subtle issues. For example, the bits of least precision in the noisy number can leak information about the original number if you're not careful.

In many ways, the release of an open-source version of Google's differential privacy library creates a whole raft of new challenges. Now there's an education program to roll out; users and developers to be supported; new tools to be built; external contributions to be curated, vetted, and tested... indeed, a whole new review process to put into place and an even broader undertaking to tackle in the form of organizing an external community of developers.

But that's just what comes with the territory whenever there are grand aspirations. The goals of Google's differential privacy team happen to be quite ambitious indeed.

TC It's great you've released this open-source library that provides for the implementation of much of the really subtle mathematical computation at the heart of differential privacy. But do you also have a lot of unit tests to make sure this isn't going to go off the rails?

DD One of the other things we open sourced along with the library was a testing framework, specifically built to verify differential privacy guarantees. But unit tests are a little difficult for that type of library. By design, differential privacy randomizes its outputs, so you can't simply check to make sure the value returned is the one you were expecting. The testing framework, on the other hand, gives you a way to empirically verify the formal property of differential privacy by generating lots of outputs and applying statistical tests. We published a description of one of our methods in the whitepaper I referred to earlier.

Anyway, yes, we agree: Testing is super-important, and special statistical techniques must be used to complement unit testing and manual auditing.

JW In looking over your open-source page, I see a few languages are supported—one seemingly better than the others. Do you plan to expand this to other languages? Or are you going to focus more on adding new algorithms? Do you think you'll manage to do a bit of both?

MG The languages supported are those we use in production at Google: Go, C, and Java. In time, we hope to offer the same set of features for each of those three languages. You'll probably soon see an experimental folder that will contain some new things like those higher-level, data-processing frameworks I mentioned earlier. There also will be some open-source things to help with the accounting for privacy budgets over a set of queries. We're definitely looking to extend our open-source library, and the things people will find there are mostly the same things we use internally, meaning we have a lot of confidence in them.

TC What if people outside of Google encounter difficulties when using the technology? After all, it's not as if they can walk down the hall to talk to the person who wrote whatever it is they're having an issue with.

MG We try to answer people's questions on the repository to the degree possible. Anyone can check the comments posted there and the issues submitted there. Our goal, actually, is to be as supportive as possible.

JW It looks at this point as though this is mostly a read-only open-source repository for people outside of Google. Do you have any plans to expand the implementation team to include people from outside?

DD In time we'd like to open it up to external contributions. At first, our C++ library didn't seem to generate a lot of

external contributor interest. For one thing, the number of people who work on differential privacy isn't huge, and C++ isn't widely used in the open-source community. Still, more recently, we've witnessed a real growth in interest, both for differential privacy in general and for our work in particular. Folks at OpenMined, for example, wrote a Python wrapper for our work and are working on Java tooling based on our libraries. We hope to attract even more people as we start to publish more in Java and Go—in particular around end-to-end tools like Privacy on Beam.

JW Whenever the time comes for you to start taking in external contributions, it should make for an interesting vetting process since this is fairly subtle stuff.

DD Exactly. Much remains to be determined in terms of what we'll need to do in the way of testing, mathematical proofs, ensuring code quality, and the rest of it prior to accepting any contribution into the repository.

MG We'll need to make sure the differential privacy mechanisms are actually doing what they're supposed to be doing—which means there would need to be some sort of review process. We're just not sure what that process ought to look like yet.

TC How widely deployed do you expect differential privacy ultimately to be?

MG It could have the sort of reach encryption has currently. In the same way that many people now use encryption by default, I'd like to see a world where people use differential privacy by default prior to analyzing datasets. That should just be a standard best practice. That's because privacy protections then would become commonplace.

There's another aspect of differential privacy we haven't talked about yet, and that's the ability to collect data in a differentially private way. So, here again, going back to that crypto analogy, I'd like to see a world where, by default, data applications collect data *only* in a differentially private manner—perhaps allowing exceptions only for specific use cases.

DD I agree with Miguel. The biggest barrier to achieving differential privacy today is not the math or a lack of theoretical research. Instead, we need more implementations and some dedicated effort to make differential privacy easier to use. Once we have that, people will be able to readily add differential privacy whenever they're publishing the results of data analysis or statistical studies. Then, maybe differential privacy will become a standard best practice rather than just a curiosity.

Should similar efforts around local differential privacy also prove successful, that too could become a best practice for data collection—at least, that's a long-term goal of ours. The only thing that stands between us and achieving that goal is more implementation, usability, and outreach work—as opposed to more research breakthroughs.

TC In terms of this becoming the default way of doing data analysis, how long will it be before a differentially private data service becomes something I can just sign up for on the Google engine or AWS (Amazon Web Services)?

MG A lot of the foundational pieces already exist on the Google site, so I don't think it should take that long. My optimistic estimate would be one year. A pessimistic

estimate would be more like three years. But I sure hope it doesn't take that long before we're able to offer default services that deliver differential privacy for end users in a more intuitive manner.

CONTENTS