DATA, PRIVACY AND THE INDIVIDUAL

FORMAL VS EMPIRICAL APPROACHES TO DATA ANONYMITY

PAUL FRANCIS
MAX PLANCK INSTITUTE FOR SOFTWARE SYSTEMS

NOVEMBER 2019

FORMAL VERSUS EMPIRICAL APPROACHES TO DATA ANONYMITY

Paul Francis

Max Planck Institute For Software Systems

ABSTRACT

The focus of data anonymity research by computer scientists is almost completely on methods with formal guarantees of anonymity, especially differential privacy. The usefulness of mechanisms with formal guarantees, however, has so far been disappointing. This article argues that computer scientists should be open to and encouraged to work on empirical data anonymization mechanisms as well—in much the same way that researchers work on both formal and empirical approaches to crypto. This article describes differential privacy and explains its benefits and shortcomings. It also describes a recently developed empirical data anonymization mechanism called Diffix, and describes how transparency and programs that incentivize white-hat attacks, such as bounty programs, can build understanding and confidence in empirical approaches. The article concludes that there is a need for both formal and empirical research on data anonymity.

Reference to this paper should be made as follows:
Francis, P. (2019) "Formal versus Empirical Approaches to Data Anonymity", *Data, Privacy and the Individual.*

INTRODUCTION

The European General Data Protection Regulation (GDPR) says that the principles of data protection need not apply to anonymous data, referring to anonymization as: *personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable*

The EU Article 29 Data Protection Working Party Opinion 05/2014 on Anonymization Techniques defines criteria for determining whether an anonymization technique is indeed anonymous. While there is substantial room for interpretation in these documents, the following two points seem clear: 1) there are tremendous benefits to anonymizing data; in terms of both reduced liability and increased safety, and 2) the bar set by the GDPR is quite high.

In October 2018, there was an academic workshop at the ACM CCS 2018 conference called "Theory and Practice of Differential Privacy." Here is an excerpt from the workshop's description:

Differential privacy provides strong worst-case guarantees [...] but is also flexible enough to allow for a wide variety of data analyses to be performed with a high degree of utility [...] it has also now been deployed in products at government agencies such as the U.S. Census Bureau and companies like Apple and Google.

Ever since Apple announced its use of differential privacy in 2016, the tech media frequently mentions it—a search for "differential privacy" on Google news finds around 200,000 articles, with 30,000 of them mentioning Apple, nearly all of them in a positive light. From these statements, differential privacy sounds like a very good technology for implementing anonymization.

The current reality, however, does not fully match expectations. I work in close research collaboration with the startup Aircloak, which markets an anonymization technology called Diffix (not differential privacy), and therefore have had numerous conversations with organizations looking to anonymize data. By way of full disclosure, I am the primary inventor of Diffix and a co-founder of Aircloak. Other than the well-publicized but limited use cases at Apple, Google, and Uber, I have yet to find an organization that is satisfied with differential privacy even though several have made a serious effort. Later in this article, I'll discuss existing implementations, but the bottom line is that there are very few, if any, that offer both acceptable utility and strong anonymity guarantees.

The vast majority of 'privacy-protected' data for analytics is only weakly protected. Typically, little more is done than removing personally identifying information like names and addresses, more-or-less along the lines for instance of what HIPPA suggests. This type of weak anonymization is called pseudonymization by the GDPR, and has been dramatically shown to be vulnerable. For example, Sweeney was able to identify the

Governor of Massachusetts in a pseudonymized medical dataset. When data is strongly anonymized, it is generally for specific data with a specific use case in mind, for instance the release of census data.

In short, weak anonymization is widely used, strong anonymization based on formal models is rarely usable, and most strong anonymization is custom-built for a specific use case. The research agenda of academic computer science (CS) is almost exclusively focused on formal approaches, and among these differential privacy dominates (150,000 hits on Google Scholar since 2017).

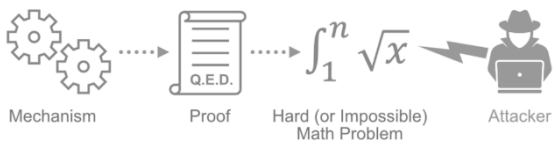
In spite of the tone of this introduction, the purpose of this article is not to argue against differential privacy. The purpose is to argue for more acceptance and focus in CS on informal or empirical research in data anonymity. The informal approach is anathema to many if not most academic researchers. As a reviewer recently said about a rejected paper on empirical data anonymity: *'By not having formal guarantees, the approach proposed reenters the arms race that frameworks like differential privacy ended.'* I believe that this arms race is both inevitable and manageable.

This article focuses on two examples of data anonymization technologies, one with mathematical guarantees of anonymity (differential privacy), and one without mathematical guarantees (Diffix).

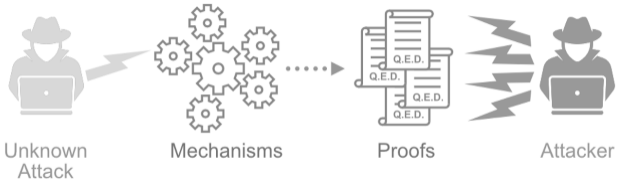
BACKGROUND: CRYPTO

It is informative to compare differential privacy and Diffix with two commonly used crypto technologies, one with a mathematical foundation (Elliptic Curve Cryptography (ECC), a public-key encryption scheme), and one without (Advanced Encryption Standard (AES), a symmetric-key encryption scheme). Public-key encryption is used to establish shared keys, and symmetric key encryption uses those keys to encrypt the actual data being encrypted.

The art of cryptography—allowing a trusted party to read a message while preventing an untrusted party from reading the same message—goes back thousands of years. During this time, attackers and defenders have been locked in an arms race, with the attackers sometimes having the upper hand—as when the Allies broke the German Enigma cypher system during WWII—and with defenders appearing to have the upper hand in modern times.



The holy grail of cryptography is to have a system that is mathematically proven to be unbreakable even when the attacker knows how the system works, as illustrated in the figure to the right. There is a complete proof that a given mechanism reduces to a known hard or impossible math problem. To effectively break the mechanism, an attacker must solve the math problem efficiently (much better than brute force)—which nobody in history has been able to do. A brute force attack is one in which the attacker simply tries every possible key. This is akin to trying every possible combination on a lock. With the exception of side-channel attacks and implementation errors (which all systems are subject to), ECC meets this ideal.



By contrast, if the mechanism is too complex, then it cannot be reduced to a hard math problem. Instead, the best that can be done is to provide proofs or strong evidence that specific known attacks fail, as illustrated in the figure to the right. The risk associated with these systems is that there may be unknown attacks. AES is this kind of system.

One might reasonably ask: if we have encryption schemes that reduce to hard math, why do we bother with encryption schemes that do not? The answer is simple: performance. ECC is very slow compared to AES. Users would rather live with the uncertainty of AES than put up with the overhead of ECC. The request for candidate algorithms for AES issued by the U.S. National Institute of Standards and Technology (NIST) in 1997 is a case in point. The document lays out the requirements for the U.S. standard for symmetric key encryption, and yet the document is only eight pages long. Nowhere does the document require formal guarantees. By contrast, the document is quite clear about the need for high performance, second only to the need for an algorithm that can defend against all known attacks.

All security mechanisms have something they *enable* and something they *prevent*. Symmetric key crypto (AES) *enables* a trusted party to read a message at very little computational cost. It *prevents* an untrusted party from reading the message without a prohibitive computational cost. Our experience with AES tells us that it is reasonable to accept more risk in order to satisfy the enabling requirement. Of course all things being equal, less risk is better than more: if it were the case that formally guaranteed encryption schemes were fast enough, or say nearly as fast as AES, then certainly it would be the preferred solution. If this were possible, NIST would have probably demanded it.

WHAT DOES DATA ANONYMIZATION ENABLE?

It is clear what anonymization should prevent, at least informally. It prevents an untrusted party from learning about individuals in the data. What does it enable? It allows an untrusted party to learn *as much as possible about the data* (while of course still preventing what it needs to prevent). In short, **anonymization should enable analytics** by an untrusted party.

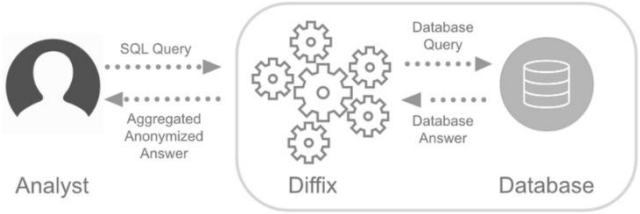
There are thousands of analytic tasks, and data scientists make a career out of learning how to do them. This page lists 16 R manuals of less than 100 pages, and 24 manuals of more than 100 pages. A PostgreSQL comprehensive manual is 3437 pages long. By contrast, the NIST AES request needed only 8 pages to specify its requirements, and even then AES is too complex for formal guarantees. The utility side of data anonymization is much more complex. Computer scientists in academia are eschewing empirical data anonymity research, meaning that a large talent pool is not adequately addressing an important societal need. Yet research on technologies that carry some risk can be valuable.

OVERVIEWS OF DIFFIX AND DIFFERENTIAL PRIVACY

With this background in place, let's take a brief look at Diffix and differential privacy.

Diffix

Diffix is an anonymization algorithm that is deployed between an analyst and the database that the analyst wishes to query (see figure to right). Diffix acts on each query (Structured Queried Language, or SQL). It parses the query and rejects or modifies those that could be used in known attacks. It interacts with the raw data, and suppresses or distorts the answer so as to satisfy anonymity. Anonymization in Diffix consists of a cohesive package of mechanisms, some decades old and some new. Old mechanisms include suppressing answers that pertain to too few users, adding "noise" by distorting answers, and changing extreme values contributed by individuals so that they match those of other individuals. A key new mechanism is *layered sticky noise*, which ties noise values to components of the query and the answer so that repeated queries cannot be used to remove noise.

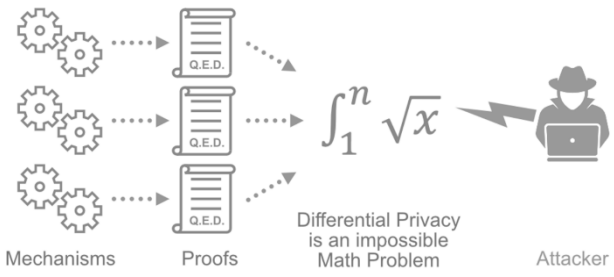


Our design philosophy for Diffix is to enable as much analytics as possible, with as little distortion as possible, while defending against known attacks. We measure the effectiveness of an attack according to the criteria of the EU Article 29 Working Party opinion: singling-out, linkability, and inference. The kinds of analytics that Diffix enables as of this writing are relatively basic. Aggregates such as count, sum, average, and standard deviation are computed across multiple attributes, leading to histograms, heat maps, probability distributions, and so on. We hope to develop machine learning in the future.

Diffix is certainly not a panacea. The amount of noise can be high when individual users contribute disproportionally to an answer, and in some cases it is hard for an analyst to know how much noise there is. Large amounts of data can be suppressed. The limitations that Diffix places on SQL are at best irritating, and at worst can prevent certain queries altogether. One cannot, for instance, simply drop Diffix into an existing SQL-based analytics application and expect it to work as is. While Diffix is head-and-shoulders more usable than alternatives of arguably comparable anonymity, there is still much to improve.

Differential Privacy

At its core, **differential privacy** is a mathematical model. The model expresses the abstract concept that if there are two databases that differ by only one user, the two databases are to a greater or lesser extent statistically indistinguishable from each other. The indistinguishability property holds even if an attacker knows everything about the data except whether this one user is included or not. The idea is that if an attacker can't even tell if a user is in the data, he or she will not be able to determine anything else about the user. In essence, differential privacy ***is*** the hard math problem. Even more, it is an *impossible* math problem, because it expresses uncertainty derived from randomness. One can no more "solve" differential privacy math than one can solve a coin toss.



Of course math alone doesn't protect data. There must be a mechanism and a proof that the mechanism reduces down to the math. There are by now hundreds of published differentially-private mechanisms. Each one aims to accomplish an analytic task, for instance building a histogram, taking an average, releasing micro-data, or generating a machine learning model. Invariably these mechanisms involve some form of random

perturbation, be it adding noise (i.e. a random number taken from a Laplacian distribution with mean zero) to a computed answer, adding noise to individual values in the database, or removing random users from the dataset.

A key characteristic of differential privacy is that each new release of information reduces the amount of uncertainty that an analyst has about the data. Or put another way, each new release of information increases the mathematical measure of privacy loss. This relationship between the number of queries and privacy loss results in the notion of a *privacy budget*—a limit on how much information can be released from a dataset. By way of analogy, suppose you are given two coins, a true coin with a 50% probability of landing heads, and a biased coin with a 51% probability of landing heads. (This is analogous to the abstract concept of two databases differing by one user.) Say you wish to determine which coin is which. With a single toss of one coin, you really have no idea which is which. If the coin lands heads, you might guess that that coin is the biased coin, but you'll be very uncertain about that. If, however, you can toss the coin hundreds of times, you can increase your certainty. If after 10000 tosses you get 5097 heads, you can be almost certain that that coin is the biased coin. The differential privacy budget in essence prevents attackers from having too many coin tosses. If attackers are allowed to learn too much precise information, the privacy of individuals in the database will be compromised.

The budget places a bound on privacy loss (or, equivalently, uncertainty). With a given budget, an attacker cannot cause more privacy loss than that which the budget allows. The budget is also a tunable parameter, by convention expressed by the Greek character epsilon (ε). An epsilon of zero means no privacy loss whatsoever (and also no data whatsoever), and an epsilon of infinity means no bound on privacy loss. The choice of epsilon is of critical importance.

There is a general concensus that epsilon less than one is very safe, and epsilon well above one is, not unsafe necessarily, but rather meaningless. Epsilon is not a measure of *actual* privacy loss, it is a measure of *worst-case scenario possible* privacy loss. Let's look at another analogy. Suppose that we have a measure of the worst-case bound on gas mileage, call it zeta (ζ). In other words, given the zeta of a car, I am guaranteed that the gas mileage is not less than zeta. Now suppose the zeta for a given compact car is 40 mpg. That is excellent! But now suppose instead that zeta is 0.5 mpg. Maybe it is so low because the car manufacturer wanted to take into account the worst possible conditions, for instance driving up a steep incline in gravel. We can be pretty sure that under normal road conditions, the car would have much better gas mileage, so zeta=0.5mpg tells us almost nothing. Likewise in differential privacy, epsilon=100 tells us almost nothing.

ANALOGIES BETWEEN CRYPTO AND DATA ANONYMITY

My premise is that Diffix is to AES what differential privacy is to ECC. I make this comparison to support the argument that, just as research on both formal and informal

crypto is important, so is it for both formal and informal data anonymity. Let's explore the extent to which these analogies make sense.

Diffix is like AES in that both lack mathematical guarantees, and both are evaluated against a set of known attacks. The similarity, however, ends there. AES is far simpler than Diffix. AES (and other symmetric key crypto) is backed by decades of research by hundreds of researchers. Its properties have been rigorously studied and are very well understood. The confidence that there are no catastrophic unknown attacks is high.

By contrast, Diffix has been under development for less than three years, has evolved substantially in that time, and is studied by a small handful of people. We are using an anonymization bounty program to motivate attacks on the system, but the amount of participation pales compared to AES, which is used worldwide.

Differential privacy is like ECC in that both have mathematical guarantees. One important difference is that requirements for ECC are relatively simple. As a result, the mechanisms are simple, the proofs are correspondingly simple and few, and therefore confidence that the proofs are correct is high.

Differential privacy mechanisms, on the other hand, can be complex or subtle, and so can their proofs be. Furthermore there are so many proofs that they can't all be thoroughly scrutinized. A paper by Lyu et.al., for instance, shows that multiple versions of a particular variant of differential privacy had incorrect proofs. It is unfortunately also quite easy to build an implementation that doesn't match the proven mechanism. In the provocatively titled blog "Uber's Differential Privacy … Probably Isn't", an early differential privacy researcher finds that the implementation of the differentially-private system built by researchers at the University of Berkeley does not match the theory. This mismatch is not surprising. Proofs are hard to get right and implementations are hard to get right. But any technology that requires a new proof and implementation for each of hundreds of analytic tasks is likely to often get it wrong.

A second important difference is that ECC (and other public-key crypto systems) demonstrably satisfies its utility goal. It is widely used. By contrast, differential privacy is struggling. There are many experts who would disagree with this statement, so it merits discussion.

USABILITY OF DIFFERENTIAL PRIVACY

The Harvard Privacy Tools Project has a differential privacy implementation called PSI. Setting epsilon to 0.5, we could build user-count histograms of 3 columns and take the mean of two more. The tool estimated the 95% error on the mean to be 3%, and on the counts to be ±60 (around 5% to 10% for most of the histogram bars). These are reasonable and useful answers, but those 5 queries exhausted the privacy budget. Were one to adhere

strictly to differential privacy, that database (a small portion of the California Demographic Dataset) could never be queried again. A dataset with more rows could have allowed more queries at similar relative error levels. The privacy budget, however, will always set a limit to the number of queries one can do, thereby preventing one from getting some or even most of the analytic value from the dataset.

The ARX Data Anonymization Tool has a feature for generating differentially-private microdata. Running this feature with an epsilon of 2 on a dataset with 16 columns and 5369 rows generates an anonymized dataset with no content whatsoever. Every value was replaced by a '*' symbol.

In the above two examples, I did not look for specific cases that perform exceptionally poorly, nor did I look for cases that perform exceptionally well. These are just two examples that seem reasonable to try that resulted in limited or no utility. The following are examples of specific use cases that do demonstrate some utility.

Apple's reported implementation of differential privacy allows for counting the number of users with certain values for given attributes (i.e. how many users used certain emoticons, the number of users who typed certain terms, etc.). The mechanism fails to find infrequently used values, and has absolute errors in the hundreds or thousands. Nevertheless, it suits the specific use case: simple counting over very large datasets. Google has also demonstrated utility for a similar kind of use case.

The US census bureau has stated that it will use differential privacy for disclosure of 2020 census data. The public release of census data is a potential sweet spot for differential privacy, because the census bureau can carefully engineer how the privacy budget is used so as to maximize utility. A paper published by the Institute for Social Research and Data Innovation (ISRDI), however, argues that using differential privacy is overkill, and may severely compromise the utility of the census data. This is an important test case for differential privacy and it will be very interesting to see how it plays out.

If differential privacy has thus far proved of limited use, then why does it get so much attention? Differential privacy provides tremendous value in ways that have nothing to do with its usability per se. First, it has and continues to provide value to academic researchers. In general, security papers involving guarantees and proofs are easier to publish than those without. Differential privacy is particularly good in this respect, in part because it is such a compelling idea intellectually, and in part because there are so many use cases to explore and therefore so many proofs to write. Second, it provides value to marketing departments, as amply demonstrated by Apple.

In addition, both marketers and researchers like to interpret facts about their products/research in a positive light. As a result, it is hard to separate the hype from the reality. For instance, it is widely reported that Apple uses differential privacy to collect personal data from its devices. The privacy statement from Apple (taken from my iPhone)

has the phrase "subject to privacy preserving techniques such as differential privacy." This means that they are using other privacy preserving techniques as well. Apple does not reveal what percentage of their personal data is protected with differential privacy.

Perhaps because differential privacy is now a marketing "brand", several companies have started offering products labeled as differentially private. One such company, Immuta, describes their mechanism here. The description states that a budget is not practical, so to get around this obstacle, Immuta:

"can capture the fact the question has already been answered, and instead of asking the room [sic] to think of a new random number, we give the same noisy response we previously calculated."

This basic sticky noise concept was tried as early as 1980, and is similar to the sticky part of Diffix' layered sticky noise. From our experience, sticky noise requires substantial complexity to defend against known attacks, and is almost certainly not provable as differential privacy.

A relevant question to accurately assess differential privacy is ***when a mechanism can be legitimately be described as differentially private.*** For instance:

- Only when epsilon is below a certain threshold.
- Only when the value of epsilon is proven.
- Only when the proof has been certified as correct.
- Only when the assumptions made in the proof are regarded as reasonable by a committee of experts, or
- Only when the assumptions are shown to hold in the deployed setting.

Today there exist no standards for when to label a mechanism as differentially private, nor is there any kind of industry forum for oversight.

Assumptions are an important part of proofs. For instance, I may have a proof that a message written on a piece of paper is secret on the assumption that attackers cannot see through the walls of a metal box. On the other hand, I may have a proof that the message is secret on the assumption that attackers are blind. Clearly the first proof is more general and useful than the second.

The Apple case provides an insightful example of how assumptions play out in the world of differential privacy. In 2016 Apple announced that they use differential privacy. They did not offer details—how the mechanism works or what epsilon they use. In 2017 a team of researchers reverse engineered Apple's client implementation, and reported that Apple's epsilon was essentially unbounded. In a press article describing the research, WIRED magazine describes Apple's response outlining a number of reasons why the researchers' conclusions about epsilon are wrong. These include that Apple doesn't correlate different data or even know how it could be done, and that it throws away user

IP addresses. Later Apple published its epsilons (between 2 and 8, which is pretty good but worse than the strong threshold of epsilon=1) and its mechanism.

The big difference between what Apple claims and what the researchers claim is due to the difference in assumptions. The researchers did not take into account additional mechanisms that happen within Apple's data centers because they looked only at the client implementation. Apple assumes (doesn't prove) that mechanisms such as dropping the IP address and changing the client identifier daily allow the overall system to be differentially private. Whether this is true or not is a matter of opinion, though to my knowledge no differential privacy experts have offered their views on this case.

TOWARDS HIGH-ASSURANCE EMPIRICAL DATA ANONYMITY

The reviewer quoted in the introduction believes that differential privacy has ended the arms race, or at least this thought can be implied from the way the comment was worded. This can't be true, given how seldom differential privacy is used. The arms race goes on with practitioners relying mainly on access control and encryption to ensure that only trusted parties have access to data that is usually only weakly anonymized.

One way to improve understanding and strength of empirical data anonymity is to encourage white-hat attacks on transparent mechanisms. The NIST AES competition is one example of how this was done for crypto. In late 2017 and early 2018, Aircloak ran a bounty program for data anonymity. This program was notable not only because it was the first ever bounty program for data anonymity, but also because it included a measure of anonymity that was used to determine the effectiveness of attacks and resulting payout.

The Aircloak bounty program was transparent: a detailed description of Diffix was openly published, and participants could view the source code on request (though none did). Participants needed to register to get access to the deployment, and in order to make priority claims on planned attacks. Although no signed documents were required, participants informally agreed to give Aircloak up to 12 months to fix discovered vulnerabilities before going public.

The measure of anonymity used to determine payout was based on the three EU Article 29 Working Party criteria for data anonymity. The three criteria—singling-out, inference, and linkability—are generally regarded as strong criteria for anonymity. The payout was designed such that even attacks that would not be regarded as serious threats could still be counted as successful and receive a payout. To mimic the fact that attackers often have external knowledge about data, the program allowed attackers to have partial knowledge of the data, though with greater knowledge leading to smaller payouts. Attackers could make an unlimited number of SQL queries to one of a number of real datasets via the

Aircloak data anonymization interface, and could supply their own datasets if they wished, so long as the data was real.

The challenge attracted around 30 attackers, two teams of which were successful (see here and here). Both attacks exploited the ability to formulate queries about specific sets of users. Aircloak implemented a defense that restricts the types of queries that can be made on columns in which a large proportion of values uniquely identify users. These specific new restrictions turned out not to be particularly onerous, though it certainly may be the case that fixes for future vulnerabilities may result in serious new limitations.

The experience with the bounty program was overall positive, and additional rounds are planned.

NIST also has a prize challenge on data anonymity, the Differential Privacy Synthetic Data Challenge. Unlike the AES competition, this challenge is not to demonstrate privacy—other than validating the correctness of the proof, the challenge assumes privacy due to the differential privacy guarantees. Rather, the challenge is to demonstrate the utility of a synthetic data set for several machine learning operations for several values of epsilon (0.1, 1, and 10).

The juxtaposition of these two challenges illustrates the thesis of this article nicely: the formal mechanisms lack utility, the informal mechanisms lack assurance of protection, and both are offering incentives to move towards utility and protection. Both approaches are legitimate and should be encouraged.

The measure of anonymity used for the Aircloak challenge can be generalized to apply to any anonymization scheme, including differential privacy. My research group has started the Open GDA Score Project to establish a General Data Anonymization (GDA) Score. The project can be used as a repository for tools, attacks, datasets, and apples-to-apples measurements of all kinds of anonymization technologies. The starting point for this project is a method we developed for measuring both anonymity and utility. As with the Aircloak Challenge, the method is based on implemented attacks on real anonymized datasets. This approach has the drawback that the measure is only as good as the attacks themselves, but the hope is that, through community participation, we can build a near-comprehensive library of attacks and utility measures with which to evaluate anonymization technologies.

CONCLUSION

Differential privacy is a beautiful theory. If it could be made to provide adequate utility while maintaining small epsilon, corresponding complete proofs, and reasonable assumptions, it would certainly be a privacy breakthrough. It remains to be seen to what extent we can make use of differential privacy. So long as differential privacy is not

developed enough to be widely used, industry will continue to use empirical approaches to anonymity, and these efforts should be better supported by the academic privacy research community.

Based on the research outlined above, the following recommendations are suggested:

- That program committees encourage submission of empirical approaches to data anonymity.
- That research funding agencies support empirical approaches to data anonymity.
- That standards be established regarding the use of the term “differential privacy”, these standards encompassing the value of epsilon, the rigor with which the value is proven, and the strength of the corresponding assumptions.