eBOOK

# How to Enhance Privacy in Data Science

2019

IMMUTA

# Table of Contents

Note: The information contained in this presentation is not intended to be and should not be construed to be legal advice. Organizations should not rely on the information herein, and they should obtain legal advice from their own legal counsel or other professional legal services provider.

# Executive Summary

In the last two decades, the ability to collect personal information on individuals has opened up a new frontier, fueling innovation and enabling companies and organizations to deliver better, more personalized services at scale. But innovation carries risks and this new frontier is rife with them, often calling for vast amounts of personal information to be digested into rich analytical products: reports, data sets, machine learning models. These data products — which provide decision support, power applications, enrich Application Program Interfaces (APIs), and pave the way to new kinds of services — require a wealth of personal data. The use of personal data leads to increased risk of data leakage, misappropriation, and loss of trust. Organizations must proactively develop controls and processes to guard personal data in order to have sustained success in a data-driven world.

It is simply no longer enough to safeguard access to raw inputs. While this remains absolutely essential, questions such as: "What information can be inferred about an individual from the behavior of this model, or from the output of our API?," and "What are the risks posed by publication of this data-set?" are relevant and equally troubling to privacy-conscious individuals — who wish to keep their secrets — and for the organizations who may bear ultimate responsibility in the event of a breach.

This document examines methods for transforming data in a manner that protects the privacy of individuals while preserving utility — some realizable benefit of data. These kinds of transformations enable organizations to release data to either internal or external consumers who, in turn, are free to examine and further analyze the data with lessened ability to attack the privacy of individuals. Specifically, we highlight the following anonymization techniques:

- **De-identification:** A process of replacing individual identifiers and, more generally, sensitive attributes with less meaningful, non-sensitive, placeholder values.
- **$k$-Anonymization:** A constraint on a dataset that ensures that no individual can be singled out from $k$-1 others given knowledge of quasi-identifying attributes such as zip-code, birth date, or biological sex. We also discuss two refinements of $k$-anonymization, $l$-diversity, and $t$-closeness.
- **Differential Privacy:** An advanced family of techniques that mathematically limit the ability of an outsider to make confident inferences about analysis input from analysis output. Analytical products produced via differential privacy enable participating individuals to credibly deny their participation in the input.
- **Local Differential Privacy:** An advanced family of techniques enabling participating individuals to credibly deny the contents of their records.

For each of the techniques, we detail its privacy-enhancing objectives, methods of implementation, and the circumstances in which it can be undermined. Examples illustrate how each transformation impacts the semantic and statistical value of the data and how the technique might be used in practice.

Our hope is that this document will provide you, the reader, with an overview of the challenges and opportunities of privacy-aware analytics. Further, this document can help equip data analysts and scientists with a framework for understanding how to implement anonymization techniques within their data projects.

INTRODUCTION

# The utility-privacy tradeoff

In the last decade, a great emphasis has been placed on extracting value from data. In the retail, travel, healthcare, finance, entertainment, and insurance industries, value is realized by collecting and leveraging data to enhance a consumer's experience. Personalization efforts such as these often require predicting preferences and behavior which, in turn, requires collecting and leveraging personal data.

> “ Data is the pollution problem of the information age, and protecting privacy is the environmental challenge.
>
> — **Bruce Schneier,** Cyber Security Expert

As an organization's knowledge of its customers becomes more detailed and intimate, the greater the *utility* — the realizable benefit — of its data.

While organizations and individuals are often adept at recognizing utility, the concept of *privacy* is a bit nebulous. In her 1975 article, “The Right to Privacy,” Judith Jarvis Thompson noted:

> Perhaps the most striking thing about the right to privacy is that nobody seems to have a very clear idea what it is.[1]

In business usage, the term privacy is often limited to matters related to privacy-centric legislation and/or compliance. While governance remains critical to the continued functioning of many organizations, casting privacy as something achieved solely through strict adherence to policy often proves short-sighted. Recent events have demonstrated that organizations wishing to act ethically and maintain the trust of their customers would be wise to serve as responsible data stewards.

But what is privacy in essence? One answer may be found in the definition provided by the International Association of Privacy Professionals (IAPP), which defines information privacy as “the right to have some control over how your personal information is collected and used.”[2] It follows immediately that organizations wishing to respect individual privacy must cede some control to those individuals.

Efforts to collect and extract value from personal data place utility in opposition to privacy. Maximizing utility requires extracting all potential value from data — and the higher the fidelity, the better. This data could be seemingly banal, such as someone's preferred brand of cottage cheese, or highly sensitive, such as which drugs someone has been prescribed. Both bits of data have value to both benign and malicious observers. Achieving a balance between these oppositional forces may seem insurmountable, but new techniques enable organizations to trade privacy for utility and vice-versa. For instance, intentionally injecting a small amount of noise into a data set may only marginally affect model performance, but substantially lower the risk that the model leaks some of the personal information on which it was trained.

The goal of this document is to provide an introduction for understanding how organizations can begin to balance the utility of personal data with privacy protections. We describe techniques which enhance privacy and detail some of their tradeoffs. Along the way, we will highlight some of the ethical and legal responsibilities of data-driven organizations.

[1] https://www.jstor.org/stable/pdf/2265075.pdf

[2] https://iapp.org/about/what-is-privacy/

# The role of the organization in preserving privacy

Privacy-conscientious organizations need to consider the myriad of ways their actions may impact the privacy of their *data subjects* — the individuals whose data they possess. Well-meaning organizations may diligently hide obvious personally identifiable information (PII) such as names, addresses, medical conditions, and the like, but still leak information that can compromise the privacy of customers. At first glance, it may seem like an organization that does not maintain any sensitive information has met its obligations — and indeed it may satisfy regulatory requirements. However, this is not sufficient when an organization values the ability of an individual to, say, keep secrets, since the revelation of any personal information could be combined with outside information to enable a third party to infer something private.

## HIPAA

Early legislative attempts at privacy enhancement focused on the data itself, specifically on requirements surrounding access and control and on the visibility of PII. The Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule[3], first proposed in 1999, restricted the purposes under which PII could be used. If used for research, for example, a so-called *limited dataset* is required for most purposes. HIPAA supports two routes to produce limited datasets: *Safe Harbor* and *Expert Determination*. Safe Harbor requires the redaction or obfuscation of identifying attributes, including 18 categories of PII. While this provided some basis for protecting privacy, the protections often prove fragile in practice, particularly when paired with external data. The second approach is *Expert Determination*. Under this approach, an individual trained in statistical methodology determines that the risk of reidentification of individuals is low — although there are no standard prescriptions on how this might be done or scored.

## GDPR

The European Union's General Data Protection Regulation (GDPR) advances a broader-reaching conception of privacy. Under the GDPR, rights flow from a framework where subjects are conceived to be owners of their data and, as such, retain rights over that data even in the hands of a third party. The rights afforded to individuals include: the right to erasure, the right to be informed, and the right to restrict processing (among others). Further, prescriptive measures are not sufficient and the companies employing specific techniques are not assured safe harbor. As a consequence, companies are forced to ensure the maintenance of these rights, have magnified responsibility to make sure data remains difficult to re-identify, and have more liability in the event of a breach.

[3] https://www.hipaajournal.com/when-was-hipaa-enacted/

These regulations are both meant to increase the level of protection and control that individuals have over their personal metadata and to reduce the risk of violation of their personal privacy. Getting ahead of the trajectory of legislation requires that organizations think about privacy in an adversarial context. It is not enough to follow a recipe and, instead, organizations must consider how their actions may erode privacy. This is a strange position. For instance, organizations are generally not privy to their users' secrets. Yet, organizations that are called upon to protect privacy in effect are asked to safeguard access to information that they do not directly possess.  For example, when experts determine re-identification risk under HIPAA, they are asked to consider what demographic information can be used to link a privatized record with an individual.[4]

**This situation presents several unique considerations, including:**

- What general policies should an organization adopt to protect the privacy of its data subjects?
- Is it possible to estimate potential impacts to the data subject upon release of personal information?
- Can we limit the advantage conferred to an untrusted outsider observing actions that we take that are informed by personal data?

[4] https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html

# What can the data scientist do?

What is the data scientist's role in all of this? What actions can a data scientist take to enhance privacy while still realizing utility? While the organization may put in place practices, policies, and decision-making processes governing data, individual data scientists can often exert control over the preparation and analysis of data, and the publication of subsequent data products. Before releasing data, then, data scientists need to assess the potential negative impact of their work on the privacy of those who contributed the data.

A useful rubric for evaluating privacy impact is the *"motivated intruder test."*[5] The test goes like this: imagine there is someone who wants to get your data, and they have significant time, energy, and resources. How hard would it be for them to get the data? If they were successful, what would they have? Ultimately, every data scientist should be able to answer the question, "If an outsider gained access to your database, what have I done to protect the personal information of my data subjects?"

A key tool for the data scientist, then, are **anonymization techniques.**[6] Although a variety of techniques exist, each of which deserve thorough treatment, the ones that we address in this document can be immediately useful to any practicing data scientist. Specifically, these are operations that alter private data or its analysis in such a way as to safeguard analytical products for release into untrusted settings where attackers may attempt to compromise the privacy of those individuals. We discuss the motivations, mechanisms, and weaknesses of three of these techniques:

- **De-identification:** A process of replacing individual identifiers and, more generally, sensitive attributes with less meaningful, non-sensitive, placeholder values.
- **$k$-Anonymization:** A constraint on a data-set that ensures that no individual can be singled out from $k$-1 others given knowledge of quasi-identifying attributes such as zip-code, birth date, or biological sex. This document will discuss two refinements of $k$-Anonymization: *l*-Diversity and *t*-Closeness: *l*-diversity requires that each distinct cohort also contain at least *l* distinct sensitive attributes. *t*-Closeness requires that the underlying distributions of sensitive attributes must be statistically similar to the overall distribution of the dataset.
- **Differential Privacy:** An advanced family of techniques that mathematically limit the ability of an outsider to make confident inferences about analysis input from analysis output. Analytical products produced via differential privacy afford participating individuals credible deniability regarding their participation in the input.
- **Local Differential Privacy:** An advanced family of techniques enabling participating individuals to credibly deny the contents of their records.

We conclude the document by briefly discussing privacy-enhancing technologies and practices, as well as other decisions a data science team might make when enhancing the privacy of their data workflow.

[5] https://ico.org.uk/media/1061/anonymisation-code.pdf

[6] These techniques are a subset of privacy-enhancing technologies, also known as PETs. The techniques discussed herein pertain to transformations of data in such a way as to mitigate the ability of an attacker to deanonymize input data, whereas PETs are more general and include things like fully-homomorphic encryption which would not protect the publication of results.

# De-identification

De-identification is the process of removing or obscuring personally identifying information (PII) within a dataset. PII attributes are defined by the National Institute for Standards and Testing (NIST) as:

> Any information about an individual maintained by an agency, including (1) any information that can be used to distinguish or trace an individual's identity, such as name, social security number, date and place of birth, mother's maiden name, or biometric records; and (2) any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information.[7]

The definition separates PII attributes into two categories: Identifiers and Quasi-identifiers. Identifiers are defined as attributes that identify an individual with high specificity. These can include name, social security number, driver's license number, passport ID, and many others. Quasi-identifiers do not provide the same level of specificity but, when combined with other attributes, quasi-identifiers may still be used to identify an individual with high specificity.

HIPAA's Safe Harbor approach to de-identifying a dataset enumerates 18 attributes of PII which must be obscured for a data set to be reduced.[8] The list includes both Identifiers, such as name and passport numbers or email addresses, as well as quasi-identifiers like birth dates, admission dates, or IP addresses. While these represent a solid foundation, the list of potential quasi-identifiers has only increased since HIPAA was enacted: even seemingly innocuous data such as calibration statistics on a phone may be used to identify a specific device.[9]

## Glossary of terms

- **Dataset:** A collection of records and attributes. Often represented as tables, where rows are records and columns are attributes.
- **Record:** A collection of attributes of some common object. This object could be a book, car, person, event, etc.
- **Attribute:** A specific feature of an object. For example the author of a book, the model of a car, the age of a person, or the time of an event.
- **Identifier:** Attributes that, by themselves, identify an individual with high specificity, such as name, social security number, or insurance policy number.
- **Quasi-identifier:** Attributes that, when combined with others, can identify an individual with high specificity, such as birth date, zip code, and gender.

[7] https://www.nist.gov/publications/guide-protecting-confidentiality-personally-identifiable-information-pii

[8] https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html#standard

[9] Zhang, Jiexin, Alastair R. Beresford, and Ian Sheret. "SENSORID: Sensor Calibration Fingerprinting for Smartphones."

De-identification through masking entails altering the attributes of a table so that individual records can't be easily traced back to the original individual. Data can be made less sensitive by replacing the true value with a masked value. A simple example is replacing someone's true name with an alias, such as **Jane Doe**. The motivation to mask PII is to obscure the true value while still preserving structure within the data. For example, if a single individual's name was always replaced with the same unique alias, records could still be grouped together and relationships preserved, even though the true value would be hidden.

## Mechanism

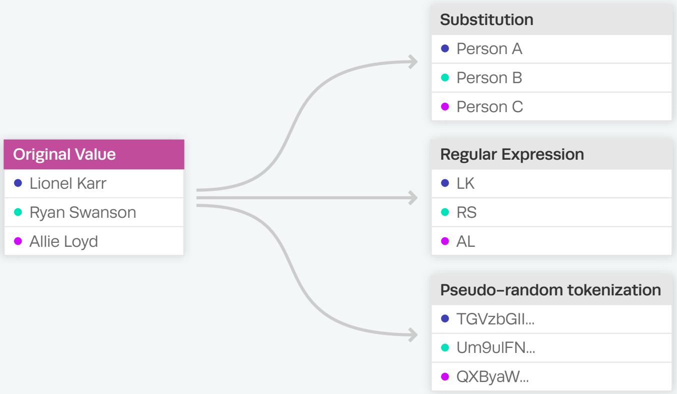
Numerous masking mechanisms can be used to obscure the data, and the choice of which masking rules to use must balance between privacy and utility. Some common ones include:

- **Substitution:** Substitution simply substitutes one value for another, typically using a codebook. This process is straightforward to implement but can be difficult to maintain as new values are observed, and is quickly compromised if an attacker can associate a small group of original values with the substituted value.

- **Regular Expression:** Regular Expression replaces a portion of an input string. Examples of this rule include removing all but the last digits of a social security number, keeping the first six digits of a credit card number, or, as shown below, revealing only the first name of an individual. A regular expression rule has some attractive features in that it can preserve some analytically useful information. For example, a rule could remove the house number from an address, preserving the street name. This would provide a small degree of anonymity to the data, while preserving some location information. However regular expression rules tend to preserve some real data in a string, increasing the risk of re-identification.

- **Pseudo-random tokenization:** Pseudo-random tokenization replaces the original value by a token which appears to be unrelated to the input string. Such a process makes it difficult for an attacker to guess the original value if they only have observed the masked value.

Figure 1 shows examples of masking data under each of these mechanisms.



Original Value
- Lionel Karr
- Ryan Swanson
- Allie Loyd

Substitution
- Person A
- Person B
- Person C

Regular Expression
- LK
- RS
- AL

Pseudo-random tokenization
- TGVzbGlI...
- Um9ulFN...
- QXByaW...

**Figure 1.** Examples of masking processes: Substitution, in which each value is recoded to a monotonically increasing value; Regular Expression, in which only the first and last letters of a name are used; and Pseudo-random tokenization, in which a hash function is used to generate a hashed string value.

## How to choose a masking function

When there is some analytic value within semantics of an input value, regular expressions may be appropriate. But often this is not the case. Using the example shown in Figure 1, the statistical utility of the label "Lionel Karr" is that it can be used to associate a series of observations about an actual person, not that its representation is "Lionel Karr." In those circumstances, another non-sensitive label, which preserves the necessary associations, is appropriate. In these circumstances, a pseudo-random tokenization provides a good process to mask data.

### Masking can be implemented with functional transformation. A few properties of a good masking functions are:

- **Hard to invert:** It should be computationally difficult for an attacker to figure out how to reverse the tokenization function. This property reduces the risk for revealing the true value given the masked value.
- **Preserves some useful structure:** Masking functions should approximately or exactly preserve some useful structure of the input. For example, masking should preserve counting statistics in a dataset, so that all records for an individual can still be grouped together, even when the individual's PII is obscured. Another example is random projection, where points in high dimensional space are projected randomly onto a lower dimensional subspace, approximately preserving interpoint distances.
- **One input produces one output:** Each true value should produce a distinct masked value. Randomly salted, collision-resistant hash functions are a quick and easy way to obtain such functions with high probability.

> **The process of salting** — appending or prepending the value with a secret randomized string — is essential when using a cryptographic hash function for tokenization.

Figure 2 shows examples of these properties.

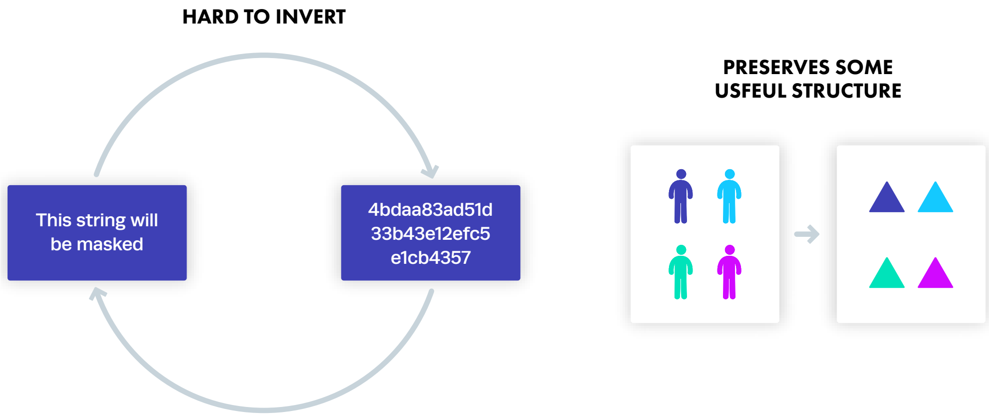
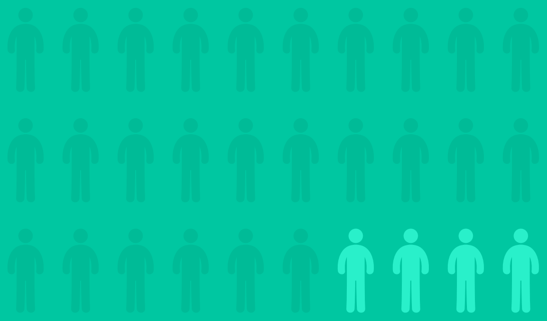

HARD TO INVERT
This string will be masked
4bdaa83ad51d 33b43e12efc5 e1cb4357
PRESERVES SOME USFEUL STRUCTURE

**Figure 2.** (Left) An example of a difficult to invert mask. Given the observed mask "4bdaa83...," an observer will have a difficult time deducing the input "This string will be masked." (Right) Here individual shapes have been "masked" as triangles, but they still have a correspondent. Their color is still there, and they still have the same number of counts in the data set.

*Cryptographic hashing* is the recommended method for tokenizing data. Cryptographic hash functions are deterministic functions that produce a uniform output over a fixed domain given some input domain of arbitrary size. Cryptographic hashing functions are extremely difficult to invert, i.e. pre-image resistant.

The process of *salting* — appending or prepending the value with a secret randomized string — is essential when using a cryptographic hash function for tokenization. The necessity of salting becomes clear in the context of so-called *dictionary attacks*: if both the hash function and set of possible true values are known in advance, an attacker may choose to build a lookup table that converts hash values back into their true values. Salting the original value prevents the attacker from being able to build this lookup table, since they would have to guess the value of the salt as well.



87%

of the population is uniquely identifiable by Zip Code, Birth Date and Sex!

## Weaknesses

De-identification through masking comes with a downside: it is vulnerable to *link attacks*, or the ability to connect a de-identified dataset to an identifiable dataset using seemingly innocuous quasi-identifier values. Generally, a dataset's vulnerability to link attacks increases as more public data is available.

A famous link attack was the re-identification of former Massachusetts governor Bill Weld's medical records. In 1997, Massachusetts General Hospital released about 15,000 de-identified medical records in which names and patient IDs had been stripped from the database. Despite the precautions, Harvard researcher, Latanya Sweeney was able to link publicly available voter information to these anonymized medical records.[10] Medical records were re-identified by linking zip code, birth date, and sex, present in both voter rolls and anonmyized medical records, as shown in Figure 3.

[10] https://fpf.org/wp-content/uploads/The-Re-identification-of-Governor-Welds-Medical-Information-Daniel-Barth-Jones.pdf

Even though the publicly released, anonymized medical records had no "name" column in them, the records did have patient's zip codes, birthdays, and genders, which the openly available voter ID list had as well. By linking the two datasets, Sweeney et al. were able to narrow down the possible records related to the governor to only a handful. In fact, it turns out that 87% of the population is uniquely identifiable by zip code, birthday, and sex. This attack was a major factor in the de-identification provisions of HIPAA's privacy rule, first published in 2000 and modified in 2002.[11]

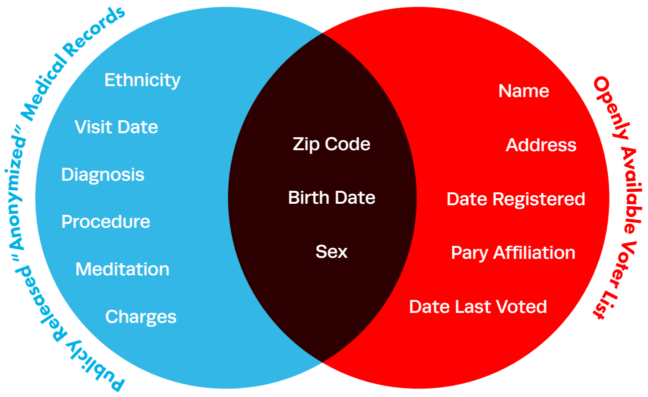


**Figure 3.** Common attributes between the de-identified medical records and public voter roles used to re-identify records from Massachusetts General Hospital.[12]

# Conclusions

De-identification using masking should be a first line of defense in most data projects. It can be safely used in many cases without making statistical sacrifices, since the underlying distributions for categorical values will be unchanged. There are many strong and lightweight cryptographic hashing functions available, with implementations across databases and programming languages — making it a low-cost operation to execute.

However, data scientists should be careful to remember that masking provides fragile privacy protections to underlying data. While guidelines such as HIPAA's Safe Harbor regulations provide a good starting place, the data landscape has changed immensely in the intervening years, meaning seemingly benign codes and values (for example, phone calibration statistics) may actually be as identifying as a person's name and address.

[11] https://www.hhs.gov/hipaa/for-professionals/index.html
[12] https://epic.org/privacy/reidentification/Sweeney_Article.pdf

# $k$-Anonymization

In response to the Massachusetts General Hospital link attack, Sweeney et al. developed the concept of "$k$–anonymization."[13] $k$–anonymization requires the quasi–identifiers of every record be identical to at least $k$ other records.

Each set of records with identical quasi–identifiers can be considered a cohort. The motivating notion of privacy in this technique is "safety in groups:" if no individual record can be unambiguously linked to a specific individual, but it is part of a group of other records that look similar, then the individual can be said to be afforded privacy. Figure 4 shows the concept of $k$–anonymization. In this example, there is a population of 10 distinct individuals, shown with 10 distinct hues. The $k$–anonymization technique groups similar individuals into three equivalence classes, depicted on the right with three distinct hues (blue, red, and green). The $k$–anonymization concept constrains each equivalence class to have at least $k$ individuals. Figure 4 is therefore an example of 2–anonymization.

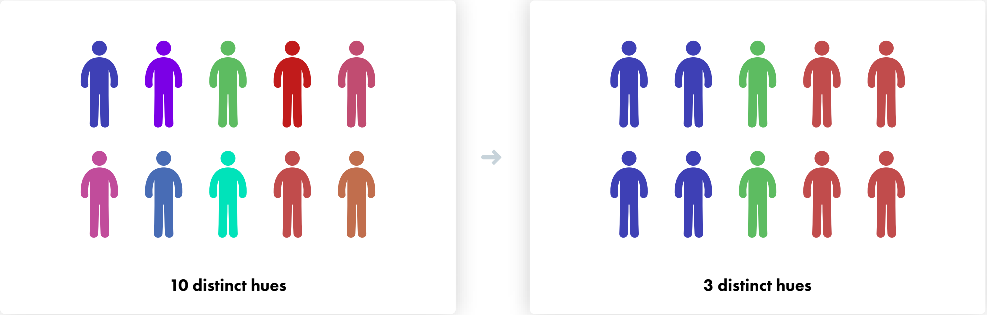


**Figure 4.** Notional 2–anonymization

## Mechanism

$k$–anonymity can be achieved by the application of two techniques: **generalization** and **suppression**. Generalization is the idea that a specific value is substituted for a more general one. For example, rolling up zip code to municipality or replacing someone's specific age with an age bracket. Suppression is the process of removing an attribute's value entirely from a cohort. An example of this would be simply removing gender distinctions from a dataset.

[13] https://dataprivacylab.org/dataprivacy/projects/kanonymity/paper3.pdf

Generalizing numeric data often involves partitioning the numeric domain into discrete, non-overlapping regions. Consider the domain of possible ages, notionally from 0 to greater than 120 years. A generalization scheme would be to partition the domain of ages into brackets [0, 15) years, [15, 24) years, [24, 35) years, [35, 45) years, [45, 55) years, [55, 70) years, and [70, ∞) years. In this example, age resolution is compromised in order to gain greater anonymity. One of the benefits of such a process is its elasticity.  In high density areas, such as between 15 and 55, the domain can be carved up into fine partitions. At the tails of a distribution, where there are a smaller number of observations, the partitioning can become coarser.

Generalizing strings or categorical entities can be achieved through the use of a generalization hierarchy.[14] A generalization hierarchy associates entities with some common characteristic. As entities lower on the hierarchy are replaced with parent values, the disclosed attribute is semantically consistent, but less specific. Figure 5 shows an example of using generalization hierarchy in a medical context. Treatment strings from a set of five medical records are shown. It could be the case that hospitals use certain types of strings more often than other hospitals, so there could be correlations by doctors about which doctors are using which strings. As you move up the chain, you lose specificity about the treatment, but you reduce the likelihood that a single attribute can lead to its original record. For example, by removing dosing or replacing the specific brand of drug with its generic treatment name, each record will have less specificity.

Generalizing date and time data can involve a mixture of both of these approaches. Date and times could be generalized by either reducing time resolution to the day, month, year, decade, etc. Alternatively, the time of the day, day of the week, season of the year could be extracted.

When taken to its extreme, generalization becomes a technique known as "suppression," in which all records have the same value for a given attribute, thus suppressing all differences between records. This is a common technique for dealing with high-cardinality columns, such as indexes or unique identifiers.

Implementing a $k$-anonymization scheme in situations where a static dataset will be released, such as when publishing a dataset, is tractable. For practical purposes, consider taxi pickup locations within New York City. A simple $k$-anonymization scheme would be to truncate the coordinates (measured in latitude and longitude) to a fixed precision, anywhere from a few meters to several kilometers. After truncation, delete any records from regions that have fewer than $k$ rides.

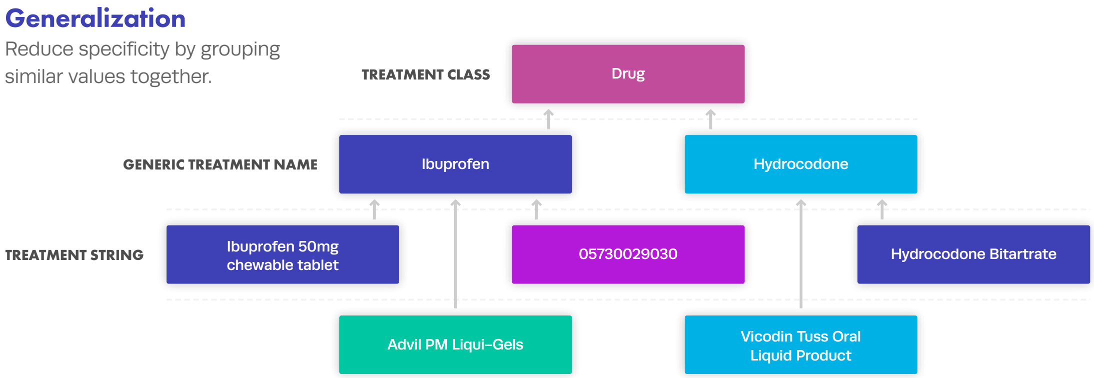


**Figure 5.** Hierarchical generalization

[14] https://www.sciencedirect.com/topics/computer-science/generalization-hierarchy

Figure 6 compares examples of $k$–anonymization. The map on the left shows the density of rides with a resolution of 0.8 m$^2$. The figure on the right shows this same data bucketed with a precision of 32.5 m$^2$. In both $k$ is set to 10. Applying this condition means that several regions with fewer than 10 pickups have been deleted from the data. These rides from the low density regions are salvaged in the lower resolution map on the left, at the cost of reducing the resolution. While there is macroscopically little impact on the underlying distribution, analysts must decide the required resolution for the end analysis.

In nearly all cases, both generalization and suppression remove information from data by reducing the granularity of the attribute. As such, how data is generalized should be tailored to each analytic project. For example, imagine other methods for $k$–anonymizing the previously mentioned taxi trip dataset. Two reasonable mechanisms might be:

- Extracting only the dates of trips, removing times.
- Extracting the hour of trips, removing any date information

If an analyst is attempting to forecast hourly demand, they would be well served to extract out *hour of day*. In contrast if they wanted to estimate seasonal variations, only the date is needed, not the time.

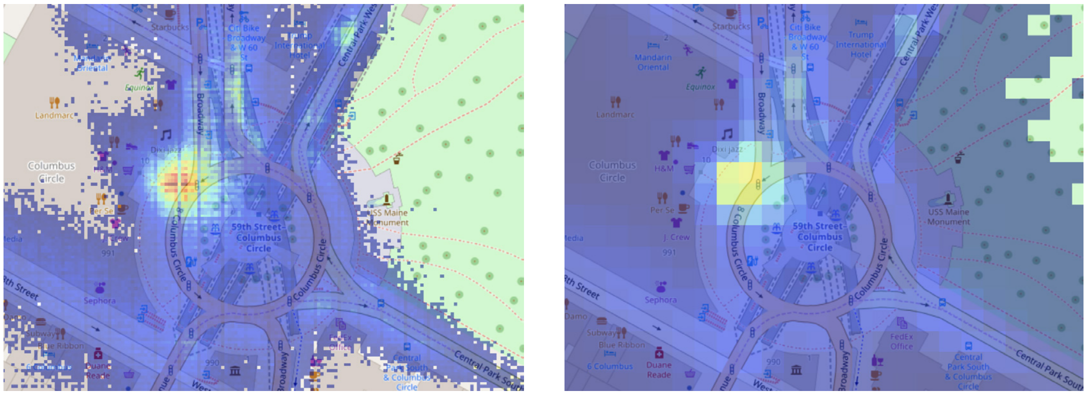


**Figure 6.** Contrast $k$–anonymization schemes. (Left) High resolution map while suppressing low density regions. (Right) Lower resolution with less suppression of low density regions.

# Weaknesses

$k$–anonymization has two central weaknesses. First, it is difficult to attain on high dimensional data or in the presence of outliers, since finding an optimal $k$–anonymization scheme, when $k$ is greater than 2, is an NP–hard[15] problem. Second, and more crucially, the privacy protections are not robust when combined with external information.

For data sources with a large number of PII attributes, each new attribute can erode anonymity. If each attribute is roughly orthogonal, individual records can quickly become unique. This can be seen with the example of birth date, gender, and zip code. Gender will divide the population to equally–sized cohorts, and zip codes will contain on

[15] http://www.aladdin.cs.cmu.edu/papers/pdfs/y2004/kanonim.pdf

average roughly 8,000 individuals. The addition of birth date breaks nearly all anonymity afforded to individuals. As such, whenever a new attribute is added to a dataset, $k$-anonymity must be re-evaluated.

While $k$-anonymization provides greater protection against record linkability than the de-identification-through-masking approach, the notion of "safety in numbers" may not protect against the leakage of sensitive information. To illustrate, consider the following example: a researcher at Big State University is developing a new neuro-prosthetic device and recruits mobility-restricted patients for her clinical trial. These patients use the device to complete a series of mobility-related tests, and undergo a battery of cognitive tests, such as IQ and cognitive flexibility. The scientist would like to publish the dataset along with a research paper, and proceeds to $k$-anonymize the reduced dataset, such that each individual record has at least two other records with identical quasi-identifiers.

Figure 7 shows a portion of a fictional dataset. It contains information about the participants' *age* (rounded to decade), *number of limbs*, IQ bucket, and mobility score. In this context, age and number of limbs are considered quasi-identifiers. Despite not being able to trace a specific record back to the individual, the researcher has leaked some private information about her participants: if an attacker learns that a person in their 50s with three limbs participated in the study, they know with certainty that he/she had a low IQ score!

| Age | Limbs | IQ | Mobility |
|---|---|---|---|
| ... | ... | ... | ... |
| ● 50 | ● 3 | Low | 52 |
| ● 50 | ● 3 | Low | 34 |
| ● 50 | ● 3 | Low | 41 |
| ● 40 | ● 4 | High | 23 |
| ● 40 | ● 4 | Medium | 46 |
| ... | ... | ... | ... |

**Figure 7.** Despite the dataset being 3-anonymous, the authors have leaked that any individual in their 50's with 3 limbs has a Low IQ score.

This type of attack is known as an "inference attack." By bringing some background information to the table, an attacker can pinpoint information about an individual, despite not knowing exactly which record it corresponds to.

To address inference attacks, additional constraints can be placed on the dataset before publication. l-diversity is one such constraint. l-diversity requires that for each sensitive attribute we publish, each cohort must have at least l instances of that attribute, where l ≥ 2. In our example, IQ would be considered sensitive. To attain l-diversity and provide deniability about the "low" IQ in the group, we could require the presence of at least 1 "medium" IQ and 1 "high" IQ in the group – making the dataset 3-diverse. To do this we can combine the two cohorts, shown in Figure 7, into one cohort of individuals ages [40, 50] and with [3, 4] limbs. The impact of this approach is shown in Figure 8.

| Age | Limbs | IQ | Mobility |
|---|---|---|---|
| ... | ... | ... | ... |
| ● [40, 50] | ● [3, 4] | Low | 5 |
| ● [40, 50] | ● [3, 4] | Low | 34 |
| ● [40, 50] | ● [3, 4] | Low | 41 |
| ● [40, 50] | ● [3, 4] | High | 23 |
| ● [40, 50] | ● [3, 4] | Medium | 46 |
| ... | ... | ... | ... |

**Figure 8.** l–diverse dataset refined from the $k$–anonymous dataset shown in Figure 7.

Caveats remain, however. If we investigate the mobility attribute, we find that nearly all (5 out of 6) of the individuals in this group have a "mobility" score of less than 45, meaning the attacker can *probabilistically infer* that the individual performed poorly on the task.

This, too, can be addressed. By restricting the attribute values within a group to be close to a certain value — say, the global mean — the researcher could effectively prevent the quasi–identifying information from being skewed towards a particular group. When combined with $k$–anonymization and *l*–diversity, this *t*–closeness constraint can provide powerful privacy protections, but it comes at a strict cost: we have eliminated or generalized many of the differences between the subgroups of the data – something that is likely of chief interest to the data scientist.

Another vulnerability of $k$–anonymization can occur when releasing two datasets anonymized under different schemes. This can be seen in the example presented in Figure 6. Records from low–density regions have been deleted from the figure on the left, while these records — initially deleted — are simply regrouped with larger regions on the right. If both datasets were to be released, an attacker could simply compare which records were disclosed by the low resolution map and not disclosed by the high resolution map, and have some confidence where the pickup occurred.

A simple example of this vulnerability is shown using the contrived dataset shown in Figure 9. The dataset contains Body Mass Index (BMI) measurements of different individuals. An analyst prepares a $k$–anonymized dataset using anonymization scheme, $k_1$, specifically averaging the lower four BMIs of the dataset and the highest three BMIs of the dataset. Sometime later a second analyst prepares a $k$–anonymized dataset but uses a different scheme, $k_2$. In this case the analyst averages the four largest and the three smallest BMIs, respectively. As a result *Doug Fulton* — one of the participants — switched cohorts. Now a savvy attacker, observing the second release, can use some simple algebra to deduce *Doug's* BMI exactly as 43.

| Name | BMI | $k_1$(BMI) | $k_2$(BMI) | Inferred BMI |
|---|---|---|---|---|
| Bob Smith | 24 | 35.75 | 33.33 | 33.33 |
| Kate Jones | 35 | 35.75 | 33.33 | 33.33 |
| Jane Armstrong | 41 | 35.75 | 33.33 | 33.33 |
| Doug Fulton | 43 | 35.75 | 46.25 | 43.00 |
| Louis Fisher | 44 | 47.33 | 46.25 | 47.33 |
| John Hopkins | 48 | 47.33 | 46.25 | 47.33 |
| Susane Lippencott | 50 | 47.33 | 46.25 | 47.33 |

**Figure 9.** Example of dataset 3-anonymized on BMI using two different schemes. Using the combination of the two schemes, an attacker can estimate the exact BMI of Doug Fulton.

This demonstrates a second vulnerability of $k$-anonymization, specifically that, under certain circumstances, changing the scheme or even adding data can compromise privacy.

## Conclusions

$k$-anonymization, and its associated constraints l-diversity and t-closeness, are holistic approaches to anonymizing datasets, addressing privacy over the totality of the datasource. The notion that data scientists should suppress high-cardinality columns and generalize quasi-identifiers to the point where few unique records exist is highly useful and can serve as an excellent test of your "privacy risk." Their benefits come with a few costs, however: there is not a common method for implementing the approach, they can be difficult to attain over high-dimensional data, the privacy protections are not robust to linkage attacked, and all can be broken with the addition of new records -- requiring that anonymity must be reassessed with each new record set. Further, when attempting to block inference attacks, the statistical differences between groups may become washed out.

There are computational challenges to finding an optimal $k$-anonymization scheme. These challenges become significantly greater as the number of quasi-identifiers expands beyond five or six attributes. In these scenarios, it is a good practice to reduce the dimensionality of the dataset by combining attributes that are strongly correlated.

There are some applications, however, where $k$-anonymization may be a very useful approach. Consider the situation where a data broker wants to create a high-level, searchable dataset to display what records are available to a potential customer. In this case, the generalization process may reduce the traceability of an uncommon attribute, and the requirement that there be at least $k$ instances of a record to be included in the database will not be an issue. After purchase, the broker could provide access to the high-resolution dataset. The anonymized dataset, then, serves as a low-resolution intermediary for publicly-revealing information about what is contained in the actual dataset.

# Differential Privacy

All anonymization techniques alter data and therefore diminish utility in some way. We give up utility in exchange for privacy. But how much privacy is gained in exchange for the loss of utility?

Unfortunately, most methods come without formal mathematical bounds. For instance, de-identification requires that certain attributes be obfuscated or redacted, but there are no assurances that these modifications are sufficient to prevent an outsider from being able to infer the participation of a certain individual from examining other record attributes or derived statistics. Similar issues exist for $k$-anonymization, which attempts to mitigate singling out individual records, but is not robust when the attacker has access to external information.

Differential Privacy (DP), on the other hand, begins from a formalization of privacy. Specifically, DP can guarantee that an attacker is limited in their ability to make inferences about individual records in a data set, even with access to ample external data. This is done by injecting a tunable amount of noise into the analysis, sabotaging the attacker's ability to make probabilistic inferences with confidence.

## Motivation

Consider a game in which an outside attacker wishes to learn some private facts regarding an individual. We want to ensure that the utility of the published analysis remains marginal when combined with outside information. To do this, we will take an extreme position. We will assume that the outsider has access to two versions of the database: one containing the record for the individual in question and one without it. Now imagine a scenario where a hypothetical analyst picks one of the two databases uniformly at random, performs differentially-private analysis, and gives the result to the attacker along with both versions of the database. The analyst is careful not to specify which version of the database was used as input to the analysis. The attacker is now asked to guess which version of the database was used to produce the analysis results. The attacker wins the round if they are able to guess which database was used as input to the analysis. Otherwise, they lose.

At first glance this may seem silly. Why give the attacker a version of the database that contains the exact record contents that we are trying to prevent them from knowing in the first place? The answer is that, in practice, no one would ever do such a thing. However, this is beside the point: Imagine that even when given this information the attacker is still unable to confidently guess whether or not the analysis was performed over the version of the database that contained the individual's data. It must follow that the analysis results carry very little information regarding the contents of the individual's record.

This kind of difficulty is guaranteed under differential privacy. The results of a differentially-private analysis are insufficient to confidently infer the participation of an individual in a database. Better still, no amount of external information could possibly help with this task — an attacker who already knows enough external information to be able to infer every other record in the database has only a negligible advantage!

## Defining Differential Privacy

A privacy mechanism for an analysis is a randomized algorithm that takes as input a database — which is generally understood to contain private information regarding some individuals — and outputs some result that is expected to reasonably approximate the analysis. Differential privacy is a property of privacy mechanisms which demands that the mechanism give the same answers with similar probabilities over any pair of databases that differ by a single row. This way, an individual may claim that the output of the mechanism, came from a database that did not include their data.

Formally, a privacy mechanism, $A$, is called $(\varepsilon, \delta)$-*differentially-private*, if for any pair of databases $D_1$, $D_2$ which differ from each other by the insertion (or deletion) of a single record, and any $S \subseteq \mathrm{Range}(A)$, then it holds that:

$$\Pr[A(D_1) \in S] \leq e^{\varepsilon} \Pr[A(D_2) \in S] + \delta.$$

In this expression $\varepsilon$ and $\delta$ are parameters chosen to limit the attacker's confidence. Roughly, when $\varepsilon$ is small, and $\delta = 0$, this condition ensures that there exists no set of inputs to $A$ that provide a significant advantage in helping an attacker determine whether the privacy mechanism was evaluated over $D_1$ or $D_2$. As $\varepsilon$, the overlap between the distributions of $\Pr[A(D_1) \in S]$ and $\Pr[A(D_2) \in S]$ is reduced, it's easier to distinguish between the two databases.

We can quantify the advantage conferred to an adversary hoping to discriminate $D_1$ from $D_2$ as follows: Let $S \subseteq \mathrm{Range}(A)$. We think of $S$ as an event which is observed by the attacker whenever $A$ returns an element of $S$. The *privacy-loss* of observing the event $S$ under the privacy mechanism $A$ is:

$$L_A(S, D_1, D_2) := \ln\left(\frac{\Pr[A(D_1) \in S]}{\Pr[A(D_2) \in S]}\right).$$

Thus, when $A$ is $(\varepsilon, \delta)$-*differentially-private*, it holds that for any pair of databases differing by a single record, the privacy loss is no larger than with probability at least $1 - \delta$.

## Sensitivity

The goal of differential privacy is to protect how much an adversary can learn about the underlying data from analysis results. This leads to the following concern: if an analysis can be heavily influenced by the presence of certain items, it may be possible to draw conclusions about the participation of those by observing the output.

We now present a pair of rather extreme examples that nicely capture the intuition.
Consider the following table:

| name | title | salary |
|---|---|---|
| Shelly McGuire | CEO | $689,000.00 |
| Cedric Rogers | Data Scientist I | $104,450.00 |
| Joel Glover | Data Scientist I | $107,759.00 |
| Jaime Holland | Data Scientist II | $125,630.00 |
| Melanie Day | Data Scientist II | $121,480.00 |
| Chris Wheeler | Data Scientist II | $129,000.00 |

**Figure 10.** Example salary table.

Suppose an analyst wishes to publish statistics by role and runs the following query:

```
SELECT MAX(salary) FROM salaries.
```

The above query is thought of as *sensitive* since, assuming no repeated entries, it is entirely determined by a single row in the table. On the other hand, a query such as

```
SELECT 3 FROM salaries GROUP BY true,
```

always returns the same result,[16] $3$. Since the latter query is entirely independent of the data, it can be thought of as *insensitive*, and the analyst need not worry about revealing it as it leaks no information about the contents of the database.

How can we quantify sensitivity? To help formalize, first view a quantitative analysis as a mathematical function, $f$, returning $k$ numerical values. That is, a vector-valued function over databases $D_1$ and $D_2$. One common measure is the $\ell_1$-*sensitivity*. This is formulated as the sum of deviations in each component maximized over all pairs of adjacent[17] databases:

$$\Delta(f) = \max_{D_1, D_2} \sum_{i=1}^{k} |f(D_1)_i - f(D_2)_i|.$$

As an example, let's assume that $f$ simply counts the number of records in the database. In this case, since any pair of adjacent databases $D_1$, $D_2$, differ by the presence of exactly $1$ record, $|f(D_1) - f(D_2)| = 1$, and thus it immediately follows that the $\ell_1$ -sensitivity of the count function is $1$.

[16] Assuming that the table must be non-empty.

[17] Adjacent databases are pairs of databases which differ by the insertion (or deletion) of a single record. The maximization is carried out over the space of all possible databases over a fixed schema.

# Mechanisms

There are numerous algorithmic mechanisms that attain differential privacy. This section quickly summarizes some of the mechanisms.

## Laplace Mechanism

The Laplace Mechanism protects function input by adding noise directly to the analysis results. At first glance it may not be clear under what circumstances adding noise to the output is sufficient to protect the input, if at all. It turns out that doing so is sufficient precisely when the $\ell_1$–sensitivity of the analysis is bounded.

The $0$–centered Laplace distribution with scale parameter $b$, denoted $\mathrm{Lap}(b)$, is given by a probability density function $\rho(x|b) = \dfrac{1}{2b}\exp\left(-|x|b^{-1}\right)$.

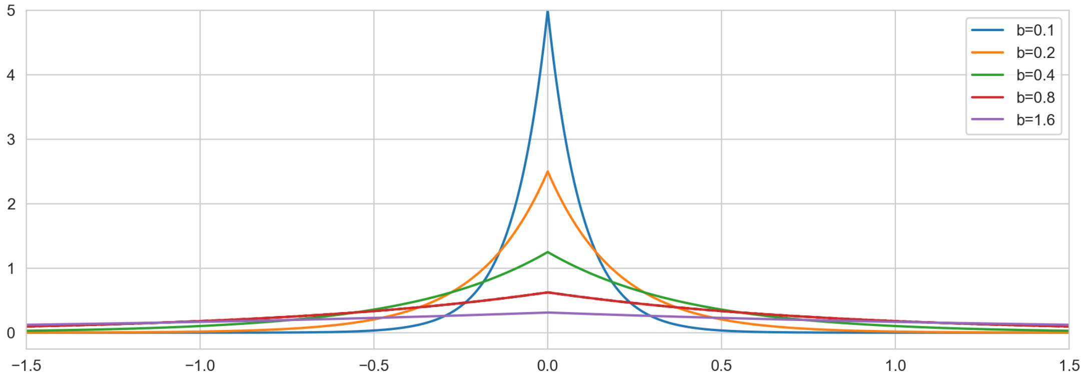


**Figure 12.** Plot of probability density function of the $0$–centered Laplace distribution, $\rho(x|b)$, for $b$ in $\{0.1, 0.2, 0.4, 0.8, 1.6\}$.

**Theorem.**[18] Let $\varepsilon > 0$, and let $f$ be a $k$–dimensional vector with real–valued entries and (finite) sensitivity $\Delta$, then the mechanism returning $f(x) + (\eta_1, \eta_2, \ldots, \eta_k)$ is $(\varepsilon, 0)$–differentially–private, provided that $\eta_1, \eta_2, \ldots, \eta_k$ are independently sampled from $\mathrm{Lap}(\Delta/\varepsilon)$.

[18] https://www.microsoft.com/en-us/research/publication/differential-privacy/

## Sample and Aggregate

Sample and Aggregate provides a method for differentially-private evaluation of sensitive functions.[19] The idea is to randomly partition the data and then evaluate $f$ over each partition. The evaluations are aggregated together into the final analysis via a differentially-private aggregation. Provided that $f$ is stable under subsampling, this strategy provides a differentially-private estimate for the evaluation $f$ over the database, even when $f$ is sensitive.

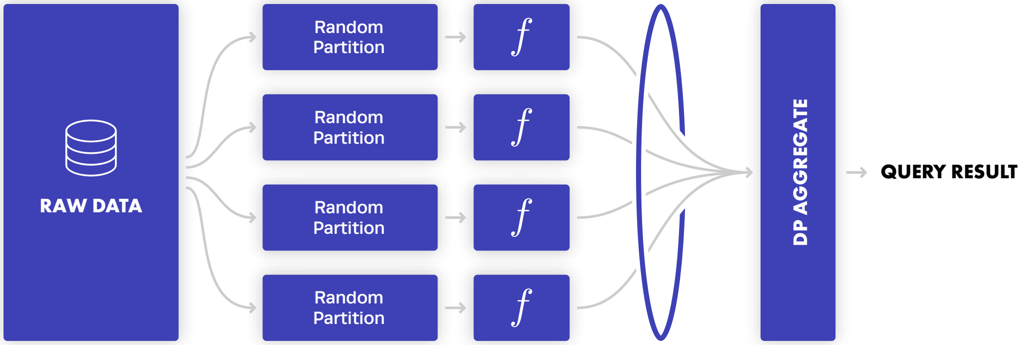


**Figure 13.** Schematic diagram of Sample and Aggregate

## Composition of Mechanisms

It is natural to wonder what level of privacy protection is afforded to individual rows in the database when multiple independent evaluations of differentially-private privacy mechanisms are available to an outsider. Effectively, neither the net privacy losses (epsilons) nor the combined tolerances of failure (deltas) can exceed their respective individual sums across all releases.

As a special case, when the database is partitioned into disjoint subsets, application of privacy mechanisms over separate partitions compose in a parallel manner. In other words, since the partitions do not overlap, neither privacy-losses nor failure tolerances accumulate, and the net privacy loss and failure tolerance are those given by the worst case values over any partition element.

# Weaknesses

**Differential privacy has several barriers to being implemented in practice:**

1. Differential privacy is a definition, not a specific process or algorithm. Though differentially-private algorithms exist for all kinds of applications — everything from machine learning models to minimum spanning trees — there is no universally defined process to achieve differential privacy. Even when algorithms are known, it can be difficult to find a readily-available implementation.
2. Differential privacy is only practical for aggregate analysis. This follows essentially immediately from the definition, which requires that any differentially-private process be insensitive to the presence of a single row.

[19] http://people.csail.mit.edu/asmith/PS/stoc321-nissim.pdf

3. Privacy erodes over time. Sequential private releases referencing the same source data can often be combined in such a way to increase the attacker's confidence. For example, in the case of the Laplace mechanism, simple averaging of privatized releases has noise-cancelling effects.

4. While differentially-private techniques enjoy formal bounds on the relative likelihood that a mechanism produces the same value over a pair of databases, it can be difficult to translate these bounds into statements that have practical value. For instance: given a differentially-private release with certain values of $\varepsilon$ and $\delta$, how likely is it that a motivated attacker will successfully learn, with at least some specified confidence, the participation of any single individual in the database? This question seems difficult — or even impossible — to address.

While differential privacy is difficult to implement, there are some software packages available. Immuta[20] offers automatic application of differential privacy to aggregate SQL queries. TensorFlow Privacy provides TensorFlow optimizers for learning differentially-private models.[21] Ektelo[22] implements an operator-based execution engine for differentially-private evaluation.

## Conclusions

Differential privacy can be employed to guarantee that data subjects achieve plausible deniability. Some mechanisms do this by adding a significant amount of noise. In many cases, you may also be required to constrain your process a bit — imagine trying to adapt a modern machine-learning framework, which is designed to be trained on example records and ask aggregate-only questions. Differential privacy may be worth the effort when your data is extremely sensitive.

[20] https://www.immuta.com/
[21] https://github.com/tensorflow/privacy
[22] https://github.com/ektelo/ektelo

# Local Differential Privacy

Local differential privacy (LDP) can be thought of as a formalization of a technique known as *randomized response*. Randomized response is a technique enabling survey participants to confidentially answer sensitive or embarrassing questions.[23]

Randomized response works by randomly instructing participants when not to answer truthfully, without revealing this fact to the interviewer. For example, one implementation places cards labeled "Answer yes," "Answer no," and "Answer truthfully" in each of three envelopes. The participant is instructed to select one at random. Since false "yes" and "no" answers occur at the same rate, they are expected to cancel. Note that two-thirds of the participants contribute nothing to the survey and yet one may estimate the true proportion from the disparity between yes and no answers.

Local differential privacy places formal constraints on the randomized substitutions. In particular, it requires that any chosen substitution be nearly (but not necessarily exactly) as likely to arise from any given input. This is a very useful property, as it ensures that all potential inputs look plausible to an attacker wishing to undo the randomized substitution.

## Motivation

Like differential privacy, local differential privacy employs randomization to enhance privacy. However, unlike differential privacy, randomization is applied prior to submission. Since the anonymization technique is applied prior to the data leaving the device, data subjects are assured protection from the moment of submission. This protection remains privatized — even in the case of subsequent breach. One key difference from differential privacy is that data release is viewed to happen at the time of collection. Thus, under this view, data subjects are not afforded plausible deniability about their participation in the data but, rather, plausible deniability about the contents of their submission.

Viewed from the point of view of an attacker looking to make inferences about participants, local differential privacy is strictly weaker than that of differential privacy. To see that this is the case, consider the following example: A medical study that is assembled to investigate the efficacy of some drug for patients showing certain anomalous blood-enzyme readings. During the study, members of the anonymous population are given devices that continuously monitor their blood-enzyme levels. The study ends with inconclusive results, and the LDP device data is made publicly available.[24] Later,

[23] https://www.jstor.org/stable/2283137

[24] This example assumes that the data is not anonymous, something which is not typically the case. As our goal is to demonstrate a certain failing of local differential privacy, we ignore legal issues and constrain ourselves only by what is permissible within the LDP

it is learned that the only possible thing that could ever cause these kinds of blood-enzyme readings is illegal drug use. As a result, the world now knows with certainty that the study participants have abused illicit substances. Such a scenario is impossible under differential privacy, since participation remains deniable.

Still, when participation can be hidden — or is of a nature such that each individual's participation would not reveal anything private about that individual[25] — then the guarantees are very similar to those provided by differential privacy. Namely, each individual user may credibly deny the contents of their record, as a significant fraction of the participants will have mis-reported.

We now formally describe these notions.

**Definition (Substitution Mechanism).** A substitution mechanism $r$ from $X$ to $Y$, (denoted $r : X \to Y$), is a randomized algorithm that takes some input value $x \in X$ and (randomly) outputs a substitute value $y \in Y$. We use the notation $r(x)$ to denote the evaluation of the substitution mechanism on an $x \in X$. Further, for any $x \in X$, and $y \in Y$, we denote the probability that $y = r(x)$, as $r(y|x)$.

**Definition (Local Differential Privacy).** Let $\varepsilon$ be a non-negative real number. A substitution mechanism, $r : X \to Y$, is *$\varepsilon$-locally differentially-private* (*$\varepsilon$-LDP*) if, for any possible output $y \in Y$, and any pair of possible inputs $x_1, x_2 \in X$, it holds that $r(y|x_1) \leq e^{\varepsilon} \cdot r(y|x_2)$.

Note that when $\varepsilon = 0$, the probability that the substitution mechanism outputs any given value must be independent of the input. To see this, fix a $y \in Y$, along with an arbitrary pair of inputs $x_1, x_2 \in X$ and note that $r(y|x_1) \leq e^0 \cdot r(y|x_2) = r(y|x_2)$. Likewise, $r(y|x_2) \leq r(y|x_1)$ since the definition still must hold if the inputs are interchanged. Combining these results show that $r(y|x_1) \leq r(y|x_2) \leq r(y|x_1)$. Thus, $r(y|x_1) = r(y|x_2)$. Since the chosen pair of inputs are arbitrary we conclude that they all must result in $y$ with the same probability, $p$. Last, it follows that $p = 1/|X|$ since,

$$1 = \sum_{x \in X} r(y|x) = \sum_{x \in X} p = |X| \cdot p.$$

model.

[25] For example: Assume that all mobile phones reported usage statistics with LDP, and that all users in the world carried exactly one mobile phone, then an attacker equipped with a crystal ball capable of providing "Yes"/"No" answers to questions of the form: "Did use X participate in the protocol" would find it useless — they are able to make all of the same inferences without it.

# Mechanisms

For our purposes, we may think of an individual as submitting a structured record consisting of $0$ or more attributes. In this picture, an individual may choose to apply a substitution mechanism to certain attributes within their record. We now outline two common methods achieving *$\varepsilon$–LDP*.

## Numerical Randomized Response

One method of randomizing numerical data is similar to the Laplace mechanism outlined in the section on Differential Privacy.

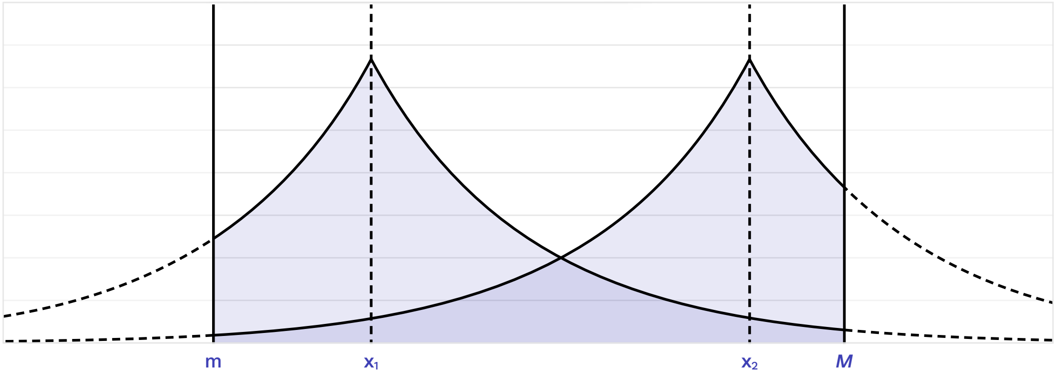


**Figure 14.** Schematic diagram of numerical randomized response. Numeric values $x_1, x_2$ come from the interval $[m, M]$. Plots of $r(x|x_1)$, $r(x|x_2)$ are shown. Note that it is possible to receive substitution values  outside of the interval $[m, M]$. If this is not desired, it is easily fixed by replacing outlying values with the nearer of $m$, or $M$.

We now formally describe the substitution mechanism $r$ which takes input in some bounded subset of the real numbers, $B$, and outputs a real number. Let $m = \inf B$, $M = \sup B$, in other words, each valid numerical value $x_i$ satisfies $m \leq x_i \leq M$. Now let $\Delta = M - m$, and consider adding to $x_i$ a value sampled from $\mathrm{Lap}(\Delta/\varepsilon)$. The probability of observing $y$ as the substitution output is

$$r(y|x_i) = \frac{\varepsilon}{2\Delta} e^{-\left(\frac{\varepsilon|y-x_i|}{\Delta}\right)},$$

Thus, for any pair of valid numerical values, $x_1, x_2$, it follows by the triangle inequality, and the fact that $|x_1 - x_2| \leq \Delta$, that,

$$r(y|x_1)/r(y|x_2) = \frac{\exp(-\varepsilon|x - x_1|/\Delta)}{\exp(-\varepsilon|x - x_2|/\Delta)} \leq \exp(-\varepsilon|x_1 - x_2|/\Delta) \leq \exp(\varepsilon).$$

That is, for any pair of valid inputs, $x_1, x_2$, it holds that $r(y|x_1) \leq \exp(\varepsilon) \cdot r(y|x_2)$, and thus this mechanism is $\varepsilon$–LDP.

## Categorical Randomized Response

Applying randomized response to categorical data requires a different approach. In this case, what is desired is a tunable substitution mechanism to blur the true value. The definition of $\varepsilon$–*LDP* constrains the behavior of the mechanism for small values of $\varepsilon$ : whatever value that it outputs must be almost equally likely under all possible inputs. Thus, if we would like the mechanism to sometimes output the true value, it must not be substantially more likely to do so then when given any other value as input. This hints at a scheme that, for any possible input, returns any valid categorical value nearly uniformly at random and that biases toward the input categorical value at large values of $\varepsilon$.

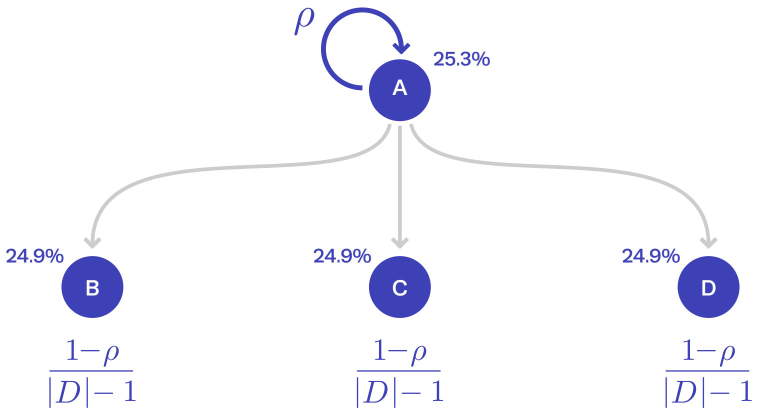


**Figure 15.** Example application of the mechanism for categorical data, for a set of possible values $\{A, B, C, D\}$, applied to an input $A$. The probability that $rr(A)$ returns $A$ is only slightly more likely than $1/|C|$, and the probability of any alternative is nearly equally likely.

**Definition (Mechanism for Categorical Data).** Let $C$ denote a set of possible labels. The substitution mechanism, $rr : C \to C$ which given any input, $c \in C$, returns $c$ with probability $e^\varepsilon/(e^\varepsilon + |C| - 1)$, and returns any other $y \in C$, $y \neq c$, with probability $1/(e^\varepsilon + |C| - 1)$, satisfies $\varepsilon$–LDP.

Notice that the definition ensures that the relative probability of releasing the true value is at most $e^\varepsilon$, which is about $1 + \varepsilon$ for small $\varepsilon$. Thus, when $\varepsilon$ is small, an attacker only has a slight advantage by guessing that the true value is the value returned by the mechanism.

We now formally show that this mechanism is $\varepsilon$–LDP. Fix a non–negative , and a substitution result $y$. Let $x_1, x_2 \in C$ denote an arbitrary pair of possible categorical values for input. First note that when $|C| = 1$, this mechanism is trivially[26] $\varepsilon$–LDP, since $x_1 = x_2 = y$, and $r(y|y)/r(y|y) = 1 \leq \exp(\varepsilon)$. When $|C| > 1$, let $x \in C$, $x \neq y$, we have:

$$r(y|x_1)/r(y|x_2) \leq \max_{x_j \in C} r(y|x_i)/ \min_{x_j \in C} r(y|x_j) \leq r(y|y)/r(y|x)$$

$$= \frac{e^\varepsilon}{e^\varepsilon + |C| - 1} \cdot (e^\varepsilon + |C| - 1) = e^\varepsilon,$$

where the second inequality follows from the fact that any value other than $y$ minimizes the denominator. That is, for any pair of valid inputs, $x_1, x_2$, it holds that $r(y|x_1) \leq \exp(\varepsilon) \cdot r(y|x_2)$, and thus this mechanism is $\varepsilon$–LDP.

[26] Intuitively, if every attribute must contain the same content then there is no risk to privacy to reveal the true value.

Note that even though the represented frequencies of the categorical values change, they do so in a predictable way. As such, it remains possible to estimate the true counts. This can be seen by observing that the expected count of any categorical value output by the randomizer is simply the sum over each input of the number of occurrences of that input that are expected to flip to that output. That is

$$C_i' = \sum_j r(i|j) C_j,$$

where, $C_i$ represents the true counts, $C_i'$ represents the transformed counts. For this mechanism we have

$$C_i' = \sum_j r(i|j) C_j = \alpha C_i + \frac{1-\alpha}{|C|-1} \sum_{j \neq i} C_j,$$

and where $\alpha = e^{\varepsilon}/(e^{\varepsilon} + |C| - 1)$ represents the probability that the input value remains unchanged by the randomizer. Given the transformed counts, $C_i'$, it is possible to solve for the input (true) counts $C_j$, provided that $\varepsilon > 0$.

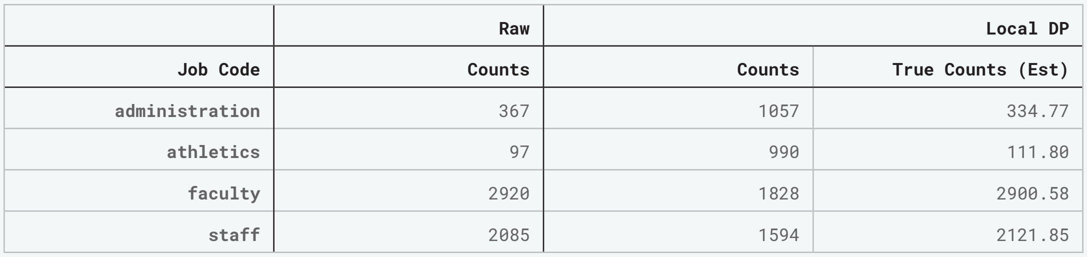
## Example

We apply these techniques to an example data set.

| | Raw | | LDP | |
|---:|---:|---:|---:|---:|
| **Hashed Name** | **Job Code** | **Salary** | **Job Code** | **Salary** |
| 033ecfec16a1a05f444f6ec5f38e30a5 | administration | $192,302.00 | administration | $193,221.49 |
| 075d0c6741c0ed1d867dfe593da83323 | faculty | $95,449.00 | faculty | $114,471.45 |
| 34ef6e5315a36264ad7ef797adeb6d89 | administration | $166,346.15 | athletics | $190,568.44 |
| 69364248bb0948fd952298bb13a80ac6 | faculty | $80,000.04 | staff | $73,701.18 |
| 6bfba29ea353a43dabcabaf1686f79ab | faculty | $31,410.28 | faculty | $202,146.13 |
| 7209c9c53029c01ff7df54f2f77d602d | athletics | $195,000.00 | athletics | $176,691.23 |
| 7fd9a94f143d2bd19e98b3844a45d3ac | athletics | $220,730.81 | athletics | $392,619.63 |
| 7fe70fef470fcc46826966e201147015 | faculty | $57,222.00 | faculty | $29,328.35 |
| 816681f7ec9c9061b5c183095aff399e | administration | $130,946.75 | administration | $217,460.27 |
| 81d6f316d169150d0e8733866c38684d | faculty | $162,282.16 | faculty | $137,565.55 |
| 8a2f0dd007880178f59f0106c4a6526a | faculty | $56,000.04 | athletics | $31,164.60 |
| 991bd2094a0c6c395a87b52707e92b60 | athletics | $189,123.04 | staff | $238,203.34 |
| b73b52b0fbcabe5ba72cf058b2106e1f | faculty | $29,324.94 | staff | $187,559.86 |
| ef783f484f403373464b8b9727913dca | staff | $29,682.95 | staff | $84,276.49 |
| ff2e211f8389cd5735ebe2cd867a7808 | administration | $392,500.04 | administration | $127,399.12 |

Figure 16. **Example application of locally differentially-private randomized response.** This table demonstrates the application of the above randomized response mechanisms for numerical and categorical data on the Salary and Job Code attributes, respectively. The effect is that infrequently occurring classes, such as Athletics and Administration are more populated. In addition, individual salaries are heavily obscured by significant noise. One pronounced example can be seen for individual 6bfba..., where their reported salary of $202,146.13 shows more than a six-fold increase from their true salary of $31,410.28.

These large swings are somewhat mitigated when the dataset is aggregated together. The table below compares the Job Code statistics measured using the raw data and the Local DP protected data.



|  | Raw | Local DP |  |
|---|---|---|---|
| Job Code | Counts | Counts | True Counts (Est) |
| administration | 367 | 1057 | 334.77 |
| athletics | 97 | 990 | 111.80 |
| faculty | 2920 | 1828 | 2900.58 |
| staff | 2085 | 1594 | 2121.85 |

Figure 17. **Example aggregation of randomized response data.** This table shows the results of aggregating the randomized response data. Despite a large change in counts, the true counts can be reasonably estimated by solving a system of equations relating the expected counts to true counts.

# Conclusion

Like differential privacy, local differential privacy can be employed to guarantee that data subjects achieve plausible deniability. Again, some mechanisms do this by adding a significant amount of noise and so error rates should be watched closely. Unlike differential privacy, deniability extends only to record contents and not to their overall participation in the dataset. So it is better to use this in cases where knowledge of participation in the dataset is unlikely to be damaging to an individual.

One major advantage of local differential privacy is that records are directly readable. This enables a straightforward approach to machine learning on private data using standard tools: iteratively apply differential privacy on the training data and train a sequence of models using decreasing values of epsilon. Among all models with acceptable performance, deploy the one with smallest epsilon.

Local differential privacy maybe worth the effort when your data is extremely sensitive or you are looking for a simple way to introduce privacy into existing machine learning pipelines.

# Preserving privacy in practice

While we have covered anonymization techniques, there are other means by which to improve the privacy of organizational data flow. These include other privacy-enhancing technologies (PETs) as well as operational practices.

Unlike the techniques already discussed, which have mostly focused on already-collected data. PETs may decrease the risk of privacy loss by decreasing the amount of sensitive data actually collected (e.g. federated learning) or by tightening how data is shared within an organization. Technological advances in PETs that may be useful in the data workflow include:

- **Federated machine learning**
  Federated learning is a process to incrementally train a centralized model across decentralized devices, alleviating the need to directly aggregate private data. This allows an organization to develop products that iteratively improve with additional users, while never directly "seeing" the potentially sensitive data.

- **Secure multi-party computation**
  A protocol enabling parties to jointly evaluate some function while keeping their own input private.

- **Zero-knowledge proofs**
  Zero-knowledge proofs are protocols that enable one party to demonstrate to another party knowledge of certain facts without actually revealing the facts.

- **Fully homomorphic encryption (FHE)**
  Fully homomorphic encryption enables an untrusted third party to carry out computation over encrypted data without having to first decrypt it. This may be useful when the data scientist wishes to outsource computation, say, to an untrusted cloud provider. Note that FHE does not constitute an anonymization technique as no information — aside from perhaps the size of the input — is released to the untrusted setting.

Enhancing the privacy of data product is more than employing a set of PETs. Creating such a data product can be a large effort involving data collection, storage, sharing, access, analysis, and publication. Private data can leak at any point of the workflow; many times the leaked data ends up being reused for purposes other than its original intent. So, in addition to transformations and technologies, there are a number of practices that decrease the likelihood of a data-loss event. A few of these are discussed below:

## Define the purpose of using data

It is vital to define how the data will be used at the outset of any project. Defining this helps contextualize the privacy-utility tradeoff and is central to gaining consent from individual data subjects. GDPR codifies this demand as *purpose limitations*, specifically requiring that organizations:

- be clear from the outset why they are collecting personal data and what you intend to do with it;
- comply with documentation obligations to specify their purposes;
- comply with transparency obligations to inform individuals about data use purposes; and
- ensure that if they plan to use or disclose personal data for any purpose that is additional to or different from the originally specified purpose, the new use is fair, lawful and transparent.[27]

## Effectively employ fine-grained access controls

Often times, in an effort to simplify management, access controls are made overly broad. For instance, analysts may be given access to an entire database or table. This practice is risky since the analyst may then have unnecessary access to sensitive data, violating the *principle of least privilege*. The end result is an increase in attack surface size for access to sensitive data, as well as an increase in the overall breach risk.

A related problem occurs when organizations instead choose to implement overly narrow access controls. This can happen in a misguided effort to secure a database, for instance, by allowing almost no one to have access. While such efforts often succeed in the stated goal (securing the database), attack surface size and breach risk nevertheless increase as demands for data are met with data dumps that may be readily duplicated, emailed, shared, left unsecured on a network device, or copied to removable media. Relatedly, it becomes much more difficult to discover when such data is being used for unapproved, and/or unsuitable purposes.

[27] https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/principles/purpose-limitation/

### Minimize the level of detail

Data collection efforts can result in massive troves of personal data. Minimization policies restrict the granularity of this information. While this may seem to hold up analytics, in most cases it may not. Frequently, data scientists and analysts will start with a subset of a dataset with which to perform exploratory analysis and initial model-building. By minimizing the amount of data that a data scientist can pull by default, a company can increase privacy protections with minimal impact to its process.

### Have a data expiration policy

Although storage is cheap, the potential cost of a privacy breach increases with size. Data expiration policies give companies a chance to reduce their data footprint in a principled way, as well as keep their focus on the most recent collection initiatives.

Preserving privacy across the entire data lifecycle is a major challenge and one that requires coordination between engineers, lines of business, IT administration, and legal teams. Although challenging, clarity about the expectations at each layer can create a culture where privacy concerns go hand-in-hand with other key objectives. One important consideration is how sharing data across multiple products or lines of business increases the exposure of data: if one team makes a dataset publicly available, then another team's effort to implement differential privacy may be in vain. It is critical for security officers to be able to grasp the centrality and reliance on certain key pieces of data *at an organizational level* in order to make decisions responsibly.

Fundamentally, protecting the privacy of your customers' data represents a trade-off between utility and privacy. While systematic approaches to obfuscating, generalizing, or randomizing data can provide set levels of protection, these decisions have to be made within a real business context, with company guidelines and regulatory requirements playing a key role in decision-making. Furthermore, teams must consider the overall governance context within which data is being leveraged.

Data scientists play a key role in leading and implementing privacy-enhancing policies; they are pioneers on a new digital frontier. As such, they are well-positioned to be leaders for what is and is not acceptable to build models off of, to implement new techniques for improving the privacy of individuals, and to put in place company practices that minimize the data "pollution" that is the natural byproduct of their work. Finding the right balance between utility and privacy will be an evolving debate, but fortunately there are many tools and techniques available to reach these goals.

Data scientists must be committed to the legal and ethical use of data to ensure that the value gleaned from this new frontier of data does not come at too high a cost.

IMMUTA

7878 Diamondback Dr, Suite C, College Park, MD 20740 | immuta.com | (800) 655-0982