NIST Special Publication 800
NIST SP 800-226

Guidelines for Evaluating Differential Privacy Guarantees

Joseph P. Near
David Darais
Naomi Lefkovitz
Gary S. Howarth

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-226

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
U.S. DEPARTMENT OF COMMERCE

NIST Special Publication 800
NIST SP 800-226

Guidelines for Evaluating Differential Privacy Guarantees

Joseph P. Near
University of Vermont

David Darais
Galois, Inc.

Naomi Lefkovitz*
Gary S. Howarth
Applied Cybersecurity Division
Information Technology Laboratory

** Former NIST employee; all work for this publication was done while at NIST.*

This publication is available free of charge from:
https://doi.org/10.6028/NIST.SP.800-226

March 2025

U.S. Department of Commerce
Howard Lutnick, Secretary

National Institute of Standards and Technology
Craig Burkhardt, Acting NIST Director and Deputy Under Secretary of Commerce for Standards and Technology

Certain commercial equipment, instruments, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification does not imply recommendation or endorsement of any product or service by NIST, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

There may be references in this publication to other publications currently under development by NIST in accordance with its assigned statutory responsibilities. The information in this publication, including concepts and methodologies, may be used by federal agencies even before the completion of such companion publications. Thus, until each publication is completed, current requirements, guidelines, and procedures, where they exist, remain operative. For planning and transition purposes, federal agencies may wish to closely follow the development of these new publications by NIST.

Organizations are encouraged to review all draft publications during public comment periods and provide feedbac to NIST. Many NIST cybersecurity publications, other than the ones noted above, are available at https://csrc.nist.gov/publications.

NIST Technical Series Policies

NIST Technical Series Publication Identifier Syntax

Publication History
Approved by the NIST Editorial Review Board on 2024-10-30

How to cite this NIST Technical Series Publication:

Near JP, Darais D, Lefkovitz N, Howarth GS (2025) Guidelines for Evaluating Differential Privacy Guarantees. (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) NIST SP 800-226. https://doi.org/10.6028/NIST.SP.800-226

Author ORCID iDs
Joseph P. Near: 0000-0003-2314-0287
David Darais: 0000-0002-3203-3742
Gary S. Howarth: 0000-0002-3587-0546

Contact Information
Privacyeng@nist.gov

Additional Information

Additional information about this publication is available at https://csrc.nist.gov/pubs/sp/800/226/final, including related content, potential updates, and document history.

All comments are subject to release under the Freedom of Information Act (FOIA).

Abstract

This publication describes *differential privacy* — a mathematical framework that quantifies privacy loss to entities when their data appears in a dataset. The primary goal of this publication is to help practitioners of all backgrounds better understand how to think about differentially private software solutions. Multiple factors for consideration are identified in a differential privacy pyramid along with several privacy hazards, which are common pitfalls that arise as the mathematical framework of differential privacy is realized in practice.

Keywords

anonymization; data analytics; data privacy; de-identification; differential privacy; privacy; privacy-enhancing technologies.

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of other than national security-related information in federal information systems. The Special Publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

Supplemental Content

This publication comes with a companion package of Python Jupyter notebooks that illustrate some of the concepts described in the publication, including how to achieve differential privacy, situations where certain differentially private algorithms could magnify bias, and utility analysis of differentially private algorithms. Supplemental content for this publication can be found at
https://github.com/usnistgov/PrivacyEngCollabSpace/tree/master/tools/de-identification/NIST-SP-800-226-SupplementalMaterial/.

Patent Disclosure Notice

NOTICE: ITL has requested that holders of patent claims whose use may be required for compliance with the guidance or requirements of this publication disclose such patent claims to ITL. However, holders of patents are not obligated to respond to ITL calls for patents and ITL has not undertaken a patent search in order to identify which, if any, patents may apply to this publication.

As of the date of publication and following call(s) for the identification of patent claims whose use may be required for compliance with the guidance or requirements of this publication, no such patent claims have been identified to ITL.

No representation is made or implied by ITL that licenses are not required to avoid patent infringement in the use of this publication.

Table of Contents

List of Tables

List of Figures

List of Appendices

Acknowledgments

The authors thank Christine Task and Damien Desfontaines for providing feedback on early drafts of this publication, as well as Ryan McKenna, Xi He, Dan Kifer, Chike Abuah, Luís Brandão, Claire McKay Bowen, Nicolas Papernot and Abhradeep Thakurta for authoring content on the NIST Differential Privacy Blog Series,[1] which informed this publication.

[1] See https://www.nist.gov/itl/applied-cybersecurity/privacy-engineering/collaboration-space/focus-areas/de-id/dp-blog.

Executive Summary

Data analytics is an essential tool to help organizations make sense of the enormous volume of data being generated by information technologies. Many organizations — in government, industry, academia, or civil society — use data analytics to improve research, develop more effective services, combat fraud, and inform decision-making to achieve mission or business objectives. However, privacy risks can arise when the data being analyzed relates to or affects individuals, which may limit or prevent organizations from realizing the full potential of data analysis. Privacy-Enhancing Technologies (PETs) can help mitigate privacy risks while enabling more uses of data.

This publication describes *differential privacy* — a PET that quantifies privacy risk to individuals when their data appears in a dataset. Differential privacy was first defined in 2006 as a theoretical framework and is still making the transition from theory to practice. This publication is intended to help those who need to manage the risks of data analytics and data sharing — including business owners, product managers, privacy personnel, security personnel, software engineers, data scientists, and academics — understand, evaluate, and compare differential privacy guarantees. In particular, this publication highlights privacy hazards that practitioners should consider carefully.

This publication is organized into four sections. Sec. 2 defines differential privacy, Sec. 3 describes techniques for achieving differential privacy and its properties, and Sec. 4 covers important related concerns for deployments of differential privacy. A supplemental, interactive software archive is also included to increase understanding of differential privacy and techniques for achieving it.

The Differential Privacy Guarantee (Sec. 2)

Differential privacy promises that a reduction in privacy caused by a data analysis or published dataset will be bounded for all individuals about whom data are found in the dataset. In other words, any privacy reduction to an individual that results from a differentially private analysis could have happened even if the individual had not contributed their data. This section introduces differential privacy, describes its properties, explains how to reason about and compare differential privacy guarantees, describes how the differential privacy guarantee can impact real-world outcomes, and highlights potential hazards in defining and evaluating these guarantees.

Differentially Private Algorithms (Sec. 3)

Differential privacy is generally achieved by adding random noise to analysis results. More noise yields better privacy but degrades the utility of the result. This *privacy-utility tradeoff* can make it difficult to achieve both high utility and strong privacy protection. Statistical disclosure control techniques, where records or features are redacted based on their perceived identifiability, can sometimes also create or magnify systemic, human, or statistical bias in results—as is generally true for statistical disclosure control—so care must be taken to understand and mitigate these impacts.

This section describes algorithms for a wide range of data processing scenarios. Differentially private algorithms exist for analytics queries (e.g., counting, histograms, summation, and averages), regression tasks, machine learning tasks, synthetic data generation, and the analysis of unstructured data. Implementing differentially private algorithms requires significant expertise primarily due to a variety of factors which includes the use of random sampling. The randomized aspects of the algorithms can be difficult to get right and easy to get wrong, and—like implementing cryptography—it is best to use existing rigorously validated libraries when possible.

Deploying Differential Privacy (Sec. 4)

Differential privacy protects privacy of data subjects in the context of intentional differentially private data releases, but does not protect data as it is collected, stored, and analyzed in raw form. This section describes practical concerns about deploying differentially private analysis techniques, including the trust model, which describes potential malicious parties and steps they might take; implementation challenges that can cause unexpected privacy failures; and additional security concerns and data collection exposure. For example, sensitive data must be stored securely with strong access control policies and mechanisms—following industry best practices—or not stored at all. A data breach that results in the unauthorized release of sensitive raw data records will nullify any differential privacy guarantee that has been established for the leaked records; however the differential privacy guarantee will still hold for all records that were not leaked.

Toward Standardization, Certification, and Evaluation

This publication is intended to be a first step toward building standards for differential privacy guarantees to ensure that deployments of differential privacy provide robust real-world privacy protections. In particular, a standard for differential privacy guarantees should prescribe a methodology for setting parameters that addresses all of the privacy hazards described in this publication, and that also balances the strength of privacy guarantees against the anticipated benefits of publishing the data. Such a standard would allow for the construction of tools to evaluate differential privacy guarantees and the systems that provide them as well as the certification of systems that conform with the standard. The certification of differential privacy guarantees is particularly important given the challenge of communicating these guarantees to non-experts. A thorough certification process would provide non-experts with an important signal that a particular system will provide robust guarantees without requiring them to understand the details of those guarantees.

Differential Privacy and Policy

Since differential privacy is the only rigorous mathematical definition of privacy at this time, it is likely to play an important role in the release of official statistics. This document is not intended to provide guidance to U.S. federal (and other government) agencies on navigating differential privacy's interactions with law, regulation, and policy. U.S. federal agencies, especially statistical agencies, have important responsibilities to release accurate information with potentially differing definitions of accuracy.

1. Introduction

Data analytics is an essential tool to help organizations make sense of the enormous volume of data being generated by information technologies. Many entities in government, industry, academia, or civil society use data analytics to improve research, develop more effective services, combat fraud, and inform decision-making to achieve mission or business objectives. However, when the data being analyzed relates to or affects individuals, privacy risks can arise. These privacy risks can limit or prevent entities from realizing the full potential of data. Privacy Enhancing Technologies (PETs) can help mitigate privacy risks while enabling more uses of data.

This publication discusses *differential privacy*—a PET that quantifies privacy loss to entities when their data appears in a dataset. Differential privacy was first defined in 2006 as a theoretical framework. In recent years, it has been successfully deployed in production systems by large technology corporations, and the U.S. Census Bureau, for which the merits of differential privacy are well documented [1]. However, differential privacy is still in the process of making the transition from theory to practice. Although production systems exist that drive large-scale deployments, the software ecosystem for differential privacy is still in its infancy. This makes it challenging for practitioners who do not specialize in PETs to deploy it easily.

New software tools for differential privacy have emerged to make deploying differentially private systems easier. However, to use these tools effectively, practitioners must understand how to interpret properly the mathematical properties of differential privacy in use by understanding how underlying assumptions translate to real world privacy harms.

The primary goal of this publication is to help those who need to manage the risks of data analytics and data sharing—including business owners, product managers, privacy and security personnel, software engineers, data scientists, and academics—better understand how to think about differentially private software solutions.

This publication identifies common pitfalls that arise as the framework of differential privacy is realized in practice. While some technical details are discussed to give appropriate context for these hazards, dense mathematical formulas are isolated to figures. An interactive software archive is referenced to supplement understanding on how differential privacy works, its guarantees, and its trade-offs.

Differential privacy has a precise mathematical definition. However, in practice, a differential privacy guarantee relies on multiple other factors. These factors are identified in the differential privacy pyramid shown in Fig. 1. The ability for each component of the pyramid to protect privacy depends on the components below it, and each is vital to achieving a meaningful privacy guarantee for end users. Evaluating any claim to differential privacy protection requires examining every component of the pyramid. As a further aid towards evaluating claims about differential privacy protections, flowcharts are provided in each section that summarize the high level risk profiles associated with various design choices.

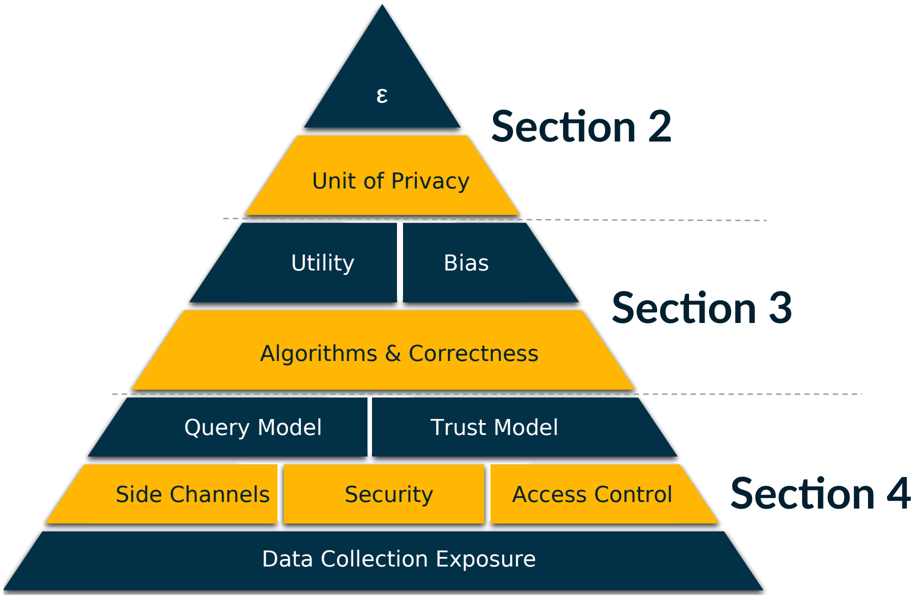


Fig. 1. Components of a differential privacy guarantee

This rest of this publication is organized into three sections:

- Sec. 2 discusses the top part of the pyramid: privacy parameters—including ε—and the unit of privacy, which together are the most direct measure of the strength of a differential privacy guarantee.
- Sec. 3 discusses the middle part of the pyramid: algorithms and correctness, how to measure utility, and the ways in which algorithms can introduce bias.
- Sec. 4 discusses the bottom part of the pyramid: access control, trust models, side channels, and data collection, each of which is important for contextualizing a differential privacy guarantee. This section also describes emerging methods for combining differential privacy with other privacy-enhancing technologies to build systems that provide comprehensive privacy protections.

This publication will help readers understand, compare, and evaluate differential privacy guarantees, and understand the ideas and tradeoffs behind some common approaches and system architectures for achieving differential privacy.

The pyramid and accompanying evaluation processes are not designed for setting the parameters of a differential privacy guarantee, though they can support it. Planning a differentially private data release or designing a differentially private system requires eliciting requirements from various stakeholders—for privacy, utility, usability, trust models, and more—then setting the parameters of the differential privacy guarantee to meet these requirements. Since differential privacy involves tradeoffs between the elements of the guarantee, it is often impossible to meet all of the requirements simultaneously, so arriving at a final design requires iterative negotiation involving all of the stakeholders.

Target Audience

This publication is primarily intended for those who need to manage the risks of data analytics and data sharing—including business owners, product managers, software engineers, data scientists, and academics. Much of the content is designed for practitioners with technical background in the design and deployment of data-processing systems, and requires a working knowledge of concepts from data science, probability, statistics, and computer science. The parts of this publication that may be helpful for less technical audiences include the privacy pyramid, flowcharts, and privacy hazards which can support decision makers in navigating the tradeoffs of deploying differential privacy solutions.

1.1. De-Identification and Re-Identification

The most common attempt to ensure that an analysis is privacy-preserving is to perform it on de-identified data. In this publication, de-identified data refers to data from which *identifying information* has been removed. Identifying information is information that could be used to identify directly a specific individual, such as a name, address, phone number, or identification number. This approach is sometimes called anonymization but is distinct from the definition of anonymization used in the European Union's General Data Protection Regulation (GDPR), Recital 26 [2].[2] NIST SP800-188 [3] provides guidance on performing effective de-identification.

Unfortunately, de-identifying data is challenging in practice because it is difficult to distinguish identifying information from non-identifying information. Every person has a unique combination of features. Any one feature (e.g., gender or age) my not be uniquely identifying on its own, but as features accumulate they inevitably become uniquely identifying in combination. As a result, de-identified data nearly always contains some information that could be identifying. For decades, it was considered prohibitively challenging to recover enough information from properly de-identified data to seriously compromise an individual's privacy [4]. However, the increasing availability of large amounts of data has led to the development of more powerful privacy attacks that disprove this assumption.

In 1997, researchers used a combination of gender, zip code, and birth date from publicly available voter registration data to re-identify individuals in a de-identified database of medical records, including Massachusetts Governor William Weld [5]. While Massachusetts stopped releasing de-identified medical records after that, researchers found that 87% of the United States population can be uniquely identified by the three elements mentioned above (gender, zip code, and birth date).[3]

The technique used by these researchers is an example of a *linking attack*: an approach for

[2] GDPR Recital 26 defines anonymous information as "information which does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable."

[3] See https://aboutmyinfo.org/identity.

exposing information specific to individuals in a de-identified dataset by matching records with a second dataset (often called the auxiliary data). Since the feasibility of a linking attack relies on the availability of good auxiliary data, the historical lack of suitable data was one basis for the belief that de-identified datasets preserve privacy. Today, however, more data are available than ever before, and linking attacks have been used to re-identify individuals in many different settings [6]. Differentially private analyses are distinct from de-identification, and they provide protection against all potential attacks, including those that make use of auxiliary data.

Ad hoc de-identification approaches typically transform each data point by redacting information considered identifying. The previous section explained why this approach is vulnerable to linking attacks. This raises a natural question: what if the data are not simply redacted at the level of each individual data point, but instead, it is aggregated before publication?

Unfortunately, even aggregate statistics can inadvertently leak information from individual data points, and result in privacy risk. For example, *reconstruction attacks* use statistics to reconstruct original data points. There are mathematical results that show that publishing enough statistics will always result in accurate reconstruction attacks [7], as well as practical examples of such attacks being performed successfully on real statistical releases [8].

1.2. Unique Elements of Differential Privacy

Differential privacy is a mathematical framework to define what privacy means—that is, an attempt to model privacy with math. There are many different techniques for increasing privacy, called mechanisms. These mechanisms satisfy particular mathematical conditions, as will be discussed in future sections. Differential privacy's status as a definition (rather than a process or technique) represents one major difference compared to techniques like de-identification. With differential privacy, one can bound the amount of information that can be learned about any individual in the data. Non-differentially private data releases are unable to bound privacy risks.

Perhaps more importantly, differential privacy has important advantages over previous privacy techniques—including de-identification—that address many of the privacy challenges described earlier in this section. Key advantages include that differential privacy is a rigorous and precise mathematical definition of privacy, that differentially private releases are resistant to all (even not yet developed) privacy attacks, and that privacy protection via differentially privacy composes across multiple data releases. These advantages, discussed at length below, are the primary reasons why a practitioner might choose a differential privacy framework over some other data privacy technique. Since differential privacy is rather new, robust tools, standards, and best-practices are not easily accessible outside of academic research communities.

Differential privacy was designed by cryptographers; there are parallels between differential privacy and cryptography both in their mathematical definitions and in their paths towards broader application and standardization. Like differential privacy, formal cryptography began as a theoretical idea with many open questions around practical applications. Over time, these open questions were answered, and today cryptography is standardized and widely adopted. There is every reason to believe that differential privacy will follow a similar path.

The following sections define terms that use the differential privacy framework and their implications on privacy in the real world, give an overview of techniques for satisfying these definitions, and discuss deployment challenges and approaches for addressing them.

1.3. Differential Privacy and the U.S. Federal Regulatory Landscape

U.S. federal agencies are governed by various laws, regulations and policies, with each agency having its own specific considerations and obligations. For example, U.S. federal agencies are required to review the quality (including the objectivity, utility, and integrity) of information before it is disseminated to the public under Information Quality Act guidelines.[4] Trust regulation issued by OMB that instructs recognized statistical agencies and units (three of which are in DOC) may only release accurate data.[5] Implementation of differential privacy in the context of certain technologies (e.g., machine learning or other forms of artificial intelligence) may also present specific requirements.[6] We encourage readers interested in the release of private official statistics to examine NIST SP 800-188 De-Identifying Government Data Sets, and the UN Guide on Privacy-Enhancing Technologies for Official Statistics. The development of guidelines on how U.S. federal agencies should meet legal, regulatory, and policy requirements is out of scope for this document.

[4] See Notice 67 FR 8452 on Guidelines for Ensuring and Maximizing the Quality, Objectivity, Utility, and Integrity of Information Disseminated by Federal Agencies, and Memorandum M-19-15 on Improving Implementation of the Information Quality Act.

[5] See 5 CFR 1321.6, Credibility and accuracy. Responsibilities of each Recognized Statistical Agency or Unit.

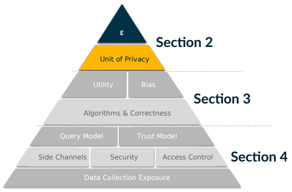
[6] See Executive Order 13960 on Promoting the Use of Trustworthy Artificial Intelligence in the Federal Government, and Memorandum M-24-10 on Advancing Governance, Innovation, and Risk Management for Agency Use of Artificial Intelligence.

2. The Differential Privacy Guarantee

This section introduces differential privacy, describes its properties, and explains how to reason about and compare differential privacy guarantees. It focuses on how the specifics of the differential privacy guarantee can impact real-world outcomes and highlights potential hazards in defining and evaluating these guarantees. Specifically:

- Sec. 2.1 defines differential privacy and describes how to interpret its formal definition in real-world terms.
- Sec. 2.2 introduces privacy parameters, which are one key factor in controlling the strength of the privacy guarantee.
- Sec. 2.3 describes several commonly used variants of the differential privacy definition.
- Sec. 2.4 describes the unit of privacy, which is the other key factor in controlling the strength of the privacy guarantee.
- Sec. 2.5 describes how to compare different privacy guarantees to each other, including the hazards of these comparisons.
- Sec. 2.6 examines the impact of mixing differential privacy with other kinds of privacy protection.

2.1. The Promise of Differential Privacy



Differential privacy frameworks provide mathematical definitions of what it means to have privacy when an individual contributes data to a particular dataset. Informally, the math of differential privacy says the chance of any outcome is about the same, whether or not the individual contributes their data. This includes every possible outcome, including those that might be considered privacy reduction to an individual. Here, the word outcome denotes the result of the analysis itself. For example, if an individual bought a pumpkin spice latte last month from their favorite coffee stand, the outcome of analyzing that coffee stand's sales data might be learning that 873 pumpkin spice lattes were sold last month. Differential privacy definitions say that the outcome of an analysis—in this case, overall pumpkin spice latte sales data—should be nearly the same with or without any single person's data. The precise notion of "nearly" is governed by the privacy budget, which is discussed in detail below.

Key Takeaway: Differential privacy promises that the chance of an outcome is about the same whether or not an individual contributes their data.

One way to view the promise of differential privacy is in terms of potential privacy harms that could be prevented, like those that can occur from re-identification attacks.

For example, imagine an insurance company wants to provide different rates to people with preexisting conditions. The company is financially incentivized to identify who has preexisting conditions through re-identification attacks on datasets. Now imagine Gary has a preexisting condition that is expensive to treat. Gary takes a survey about his medical history and the survey results are published using a differentially private mechanism. The insurance company then tries to analyze the differentially private survey results to infer information about Gary's preexisting condition. The differential privacy guarantee says that whatever the insurance company learns from the differentially private survey results will be similar with or without Gary's participation in the survey. From the perspective of the insurance company trying to learn information about Gary, this has the effect of making it appear as if Gary never contributed his data in the first place, and renders the differentially private survey results useless for the purposes of trying to learn about Gary's preexisting condition. The extent to which Gary's participation in the survey can change differentially private survey results is governed by the privacy parameters (including privacy budget) used in the differentially private mechanism, as discussed in Sec. 2. However, the insurance company could still use other data sources that are not differentially private to violate Gary's privacy and learn about his preexisting condition.

Another useful way to consider the promise is to imagine two hypothetical worlds:

1. In the real world, X lives in a city, owns a smartphone, pays with a credit card, and uses social media.
2. In an off-grid world, X lives in an off-grid cabin and is self-sufficient. No organization collects any data directly from X.

The off-grid world is designed to encode an informal notion of "perfect privacy." Differential privacy promises that the chance of an outcome will be about the same in both worlds, meaning that privacy reductions that occur in the real world could just as easily have occurred in the off-grid world.

However, population-level information can often allow one to infer information about individuals. Differential privacy thus does not protect against inferences made about an individual as long as those inferences can be made without that individual's data. For example, differentially private statistics might allow us to learn the following fact: most people have eyebrows. From this fact, we can infer that Joe probably has eyebrows. We can infer this information whether or not Joe lives in the real world or in off-grid world. Differentially private releases prevent us from improving the accuracy of our inferences about any individual in the data while still enabling inferences about the population.

Key Takeaway: Differential privacy does not necessarily prevent somebody from making inferences about an individual.

Alignment with other definitions of privacy

It is helpful to demonstrate how the following examples of definitions do or do not align with differential privacy to illustrate the capability and the limits for differential privacy to meet specific privacy protection needs in practice.

The NIST Privacy Framework [9] characterizes privacy as a state that safeguards important values, such as human autonomy and dignity. Privacy risks arise from problematic data actions, which are actions taken on data that could cause an adverse effect for individuals.[7] Differential privacy provides a strong defense against many of these problematic data actions, including common concerns like re-identification. Methodologies like the Privacy Framework can help contextualize the protection provided by differential privacy and assess whether that protection matches real-world expectations.

Tore Dalenius, an influential survey statistician described inferential disclosure as the possibility of learning a sensitive attribute with high but not total certainty [11]. This informal notion has been used in statistical disclosure limitation (SDL) literature for decades. Differential privacy does not prevent inferences that can be made with population-level information (like the example above), even though these count as inferential disclosures under the Dalenius definition.

More recent work has shown [12–14, 14] that it is impossible to prevent Dalenius's inferential disclosures while using statistics to gain scientific knowledge. This line of work rejects Dalenius's definition, and proposes a new definition for inferential disclosure: access to privacy-preserving statistics should not enable one to learn anything about an individual that could not be learned without that individual's data. This definition of inferential disclosure aligns perfectly with the promise of differential privacy.

2.1.1. The Math of Differential Privacy

The original definition to use the differential privacy framework was pure ε-differential privacy [15]:

Definition: Pure ε-differential privacy. Let $\mathscr{M}$ be a randomized mechanism. $\mathscr{M}$ satisfies ε-differential privacy if for all *neighboring datasets* D_1 and D_2 and all possible outcomes S:

$$\frac{Pr[\mathscr{M}(D_1) \in S]}{Pr[\mathscr{M}(D_2) \in S]} \leq e^{\varepsilon}$$

D_1 and D_2 are considered neighbors if they differ in the data of one individual.

The definition says that the ratio of two probabilities should be less than or equal to e^{ε}, where ε is a number called the *privacy parameter*, the *privacy loss* or the *privacy budget*.

[7] The NIST Privacy Risk Assessment Methodology (PRAM) [10] catalogs some examples of problematic data actions.

One can think of the numerator as the chance that outcome S occurs in the real world (i.e., with X's data) due to processing the data in some way (i.e., the mechanism M), while the denominator is the chance that the same outcome S occurs in an off-grid world (i.e., without X's data). The definition is symmetric, so the two cases can be reversed. The ratio between the two probabilities should be close to 1 (i.e., $\leq e^{\varepsilon}$) and encode the requirement that the chance of each outcome should be about the same in both cases.

For example, consider a scenario in which 632 pumpkin spice lattes were sold in October. In order for this to satisfy differential privacy according to Definition 1, the probability that an analysis on dataset D_1 returns the number 632 should be about the same as the probability that an analysis on D_2 returns the same answer. This should also be true of every possible answer one could observe (i.e., every output of the analysis $\mathscr{M}$, not just 632), and for every hypothetical choice of datasets D_1 and D_2.

Definition 1 says that D_1 and D_2 must be neighboring datasets, which differ in one individual's data. Thus, the difference between the real world and an off-grid world can be encapsulated in the availability or non-availability of one person's data. Neighboring datasets can be defined using the *unit of privacy* that has major impacts on the real-world implications of the differential privacy definition. The unit of privacy is discussed in Sec. 2.4.

Key Takeaway: The differential privacy guarantee is defined by both the privacy parameters (e.g., ε) and the unit of privacy (i.e., the definition of neighboring datasets).

2.1.2. Properties of Differential Privacy

The definition of differential privacy has intuitive appeal, but it also has some important properties that address many of the shortcomings of previous approaches to privacy.

1. Differential privacy treats all information as identifying information, eliminating the challenging and sometimes impossible task of accounting for all identifying elements of the data.

2. Differential privacy is resistant to privacy attacks based on auxiliary data, so it can effectively prevent the linking attacks that are possible on de-identified data.

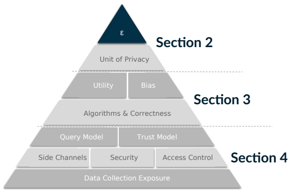
3. Differential privacy is compositional, meaning that the "total privacy reduction" of multiple data releases can be considered to ensure that it does not get too large over time.

These properties are direct mathematical implications of the definition itself—they can be proved true.

Two other useful properties of differential privacy are *post-processing invariance* and *group privacy*. The *post-processing invariance* property says that the output of a differentially private mechanism remains differentially private even if other processing is performed on it—in other words, it is not possible to un-do the differential privacy protection after it has been applied. The *group privacy* property says that if a differentially private mechanism provides privacy protection for one person (defined using the *unit of privacy*, described in Sec. 2.4), then it also provides (weaker) protection for a group of people. The strength of the guarantee depends on the size of the group.

2.2. The Privacy Parameter ε

At the top of the pyramid in Fig. 1, the privacy parameter ε controls how similar differential privacy's two hypothetical worlds need to be. If ε is very small, then the two worlds need to be nearly identical, implying a very strong privacy guarantee. When ε is large, the two worlds are allowed to be further apart, implying a weaker privacy guarantee.



This dynamic is shown in Fig. 3. The most common way to achieve differential privacy is by adding random noise. Thus, as ε gets smaller, the results show stronger privacy but less accuracy. This publication refers to this tension as the *privacy-utility tradeoff*. Sec. 3.2 discusses utility and how to measure it.

This publication will demonstrate that ε is just one of many choices for the privacy parameter (also called the privacy loss parameter), and will discuss what constitutes a good privacy parameter in this section. The flowchart shown in 2 distills the essence of what makes a privacy parameter choice low or high risk.

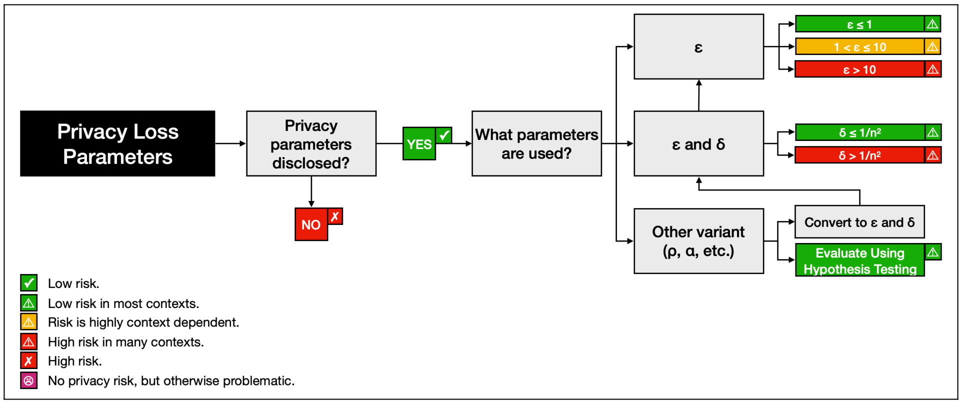

Privacy Loss Parameters
Privacy parameters disclosed?
YES ✓
NO ✗
What parameters are used?
ε
ε and δ
Other variant (ρ, α, etc.)
ε ≤ 1 ⚠
1 < ε ≤ 10 ⚠
ε > 10 ⚠
δ ≤ 1/n² ⚠
δ > 1/n² ⚠
Convert to ε and δ
Evaluate Using Hypothesis Testing ⚠
✓ Low risk.
⚠ Low risk in most contexts.
⚠ Risk is highly context dependent.
⚠ High risk in many contexts.
✗ High risk.
☹ No privacy risk, but otherwise problematic.

Fig. 2. An example flowchart for determining whether or not a privacy parameter is low or high risk. The specific values of the privacy loss parameters, ε and δ (discussed in detail in the text) will depend on a curator's goals and constraints, and the context of the analysis.

Key Takeaway: Smaller ε means stronger privacy but lower accuracy. Larger ε means weaker privacy but higher accuracy. This dynamic is called the *privacy-utility tradeoff*.

One of the properties of differential privacy is the ability to compose a privacy loss budget of multiple releases. For example, if differential privacy is released using an ε of 1, and the same analysis is re-run and published, the combined ε across both releases is 2. Some organizations are considering global privacy budgets for individuals to create an upper limit of what can be learned even with multiple releases.

Selecting privacy loss parameters, such as ε, is challenging, and we offer no specific guidelines on their selection. The choice will be depend on the sensitivity of the data, the goals and constraints of the data curator, and a consideration of the data in context. Typically, careful expert consideration is required to establish suitable privacy loss parameters.

Privacy Hazard: Large values of ε may not provide meaningful privacy.

Open Question: How to set ε is still an active area of research.

We encourage vigorous work to inform the ongoing discussion of how to wisely select privacy parameters. Publishing privacy parameters establishes trust and accountability in differentially private releases. Publishing the parameters also does not create any additional risks to data privacy.

When navigating the choice of privacy parameters, a few loose suggestions can be helpful starting places. Analysis from one study [16] suggests that ε of 0.1 generally provides strong privacy protection, and that ε values less than 1 are considered to be reasonable. The situation is less clear for larger values of ε. However, many deployments of differential privacy have used larger values (i.e., $1 < \varepsilon \leq 20$) [17]. Experiments have shown that ε values on the larger end of this scale do not always provide meaningful real-world privacy [18], but the impact of ε in the real world seems to be highly dependent on the situation, and larger values of ε may still provide meaningful privacy in some cases. Organizations choosing to release data with differential privacy are encouraged to evaluate the potential risk of the final data, and this is especially important for ε values greater than 1. NIST is actively working on empirical privacy metrology that can assist analysts in the estimating of disclosure risk. See the NIST PETs Testbed[8] as an entry point for relevant NIST initiatives.

Existing research has shown that in theory, differentially private releases with ε of 10 or greater can leave outliers vulnerable to privacy leakage. Wood et al. [16] include analysis of a simple mechanism that answers a single question. For this mechanism, even releases with $\varepsilon = 1$ can help an adversary make more confident guesses. More complex releases—which may split the privacy budget across many queries, train machine learning models, or output synthetic data—are usually less susceptible to attacks. However, research has shown that even more complicated algorithms can leave outliers vulnerable when ε is large. For example, Stadler et al. [19] show that in certain conditions, differentially private synthetic data constructed with $\varepsilon = 10$ may leave outliers vulnerable to linkage attacks, and Nasr et al. [20] show that maliciously crafted training data sets can result in significant leakage from differentially private neural networks when $\varepsilon = 10$.

[8] https://www.nist.gov/itl/applied-cybersecurity/privacy-engineering/collaboration-space/testbed

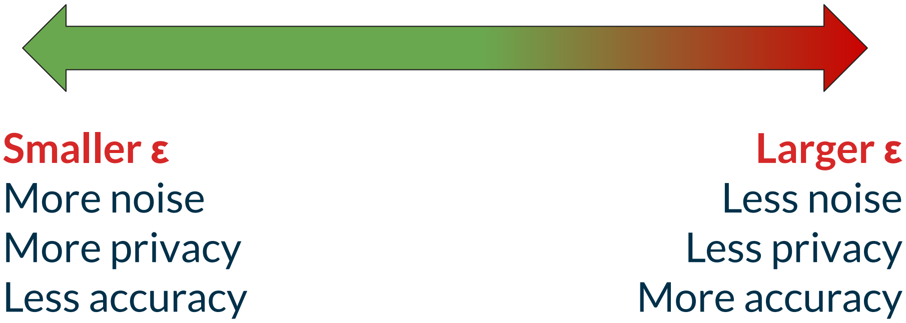


Fig. 3. Impact of the privacy parameter ε: the privacy-utility trade-off.

Over time, we hope it will be possible to benchmark enough use cases to establish guidelines for privacy parameters within specific contexts. One effort in that direction is the NIST Collaborative Research Cycle (CRC), an effort to benchmark de-identification algorithms generally, including differentially private methods. The CRC uses real demographic data sourced from the American Communities Survey from the U.S. Census Bureau, accepts de-identified instances of the data from the community, and evaluates the de-identified data using a host of fidelity, utility, and privacy metrics. With this and other similar efforts, we hope the community will work toward developing a more sophisticated understanding of the interplay of privacy and utility that will lead toward best practices. We think it likely that domain and task-specific best practices will start to emerge with time.

It is common for the same data to be analyzed many times. In this context, it is common to view the ε parameter as a *privacy budget*—an upper bound on the total allowable privacy loss for all analyses of the data. The composition property of differential privacy allows us to add up the individual ε parameters for many analyses of the same data to compute an upper bound on the cumulative privacy loss of these analyses. For example, an organization may perform 10 individual differentially private analyses on a dataset, each with a privacy parameter of $\varepsilon_i = 0.1$. In this case, the total privacy budget is $\varepsilon = 10 \times \varepsilon_i = 1$.

Key Takeaway: If one sensitive dataset is analyzed many times using differential privacy, the individual ε parameters can be added up for the analyses to compute an upper bound on the cumulative privacy loss of these analyses—a "total ε" often called the *privacy budget*.

The privacy parameter ε is an upper bound on privacy loss, rather than an approximation or measurement of it—the actual privacy loss experienced by an individual will never be larger than ε, but may be much smaller. Moreover, ε is just one possible upper bound on privacy loss; other variants of differential privacy may provide more accurate modeling of privacy loss, and often leverage parameters other than ε. These variants are discussed in Sec. 2.3.

2.3. Variants of Differential Privacy

The original definition of differential privacy is also called ε-differential privacy or pure differential privacy. Since the original development of this definition, several variants have been designed that model privacy loss more accurately in some cases. Here we consider some of the common variants and their trades offs. For a more detailed discussion of variants, their parameters, important characteristics, and how variants relate to each other see [21].

Benefits of privacy variants

Table 1 summarizes the commonly used variants of differential privacy. The primary benefit of most variants is improved utility over pure ε-differential privacy. There are two main reasons for the improvement:

1. All four variants enable the use of Gaussian noise (described in Sec. 3.1), which can significantly improve utility in some cases.
2. All four variants enable tighter bounds on composition, resulting in lower privacy budgets for iterative algorithms.

To obtain these benefits, each of the variants weakens the privacy guarantee slightly compared to pure ε-differential privacy.

Selecting a variant.

When many statistics are being released or an iterative algorithm is used, then using one of these variants can significantly improve accuracy. When selecting a variant, Rényi differential privacy, zero-concentrated differential privacy, or Gaussian differential privacy are preferred because they offer the best utility and the smallest weakening of the guarantee.

(ε, δ)-differential privacy and catastrophic failure.

The final variant—(ε, δ)-differential privacy (also called approximate differential privacy)—includes a parameter δ (pronounced "delta") that allows mechanisms to provide no privacy guarantee at all for rare events (see Appendix Sec. B.1 for the formal definition). For example, a mechanism that picks one person from a dataset of n people and releases their data with no noise at all can still satisfy (ε, δ)-differential privacy as long as $\delta > \frac{1}{n}$.

Privacy Hazard: Due to the possibility of catastrophic failure, when using (ε, δ)-differential privacy, set $\delta \leq \frac{1}{n^2}$.

This guarantee can allow for a complete, catastrophic failure of privacy. To obtain meaningful real-world privacy protection with (ε, δ)-differential privacy, δ is typically set very small compared to n so that mechanisms like the example above are not

Table 1. Variants of differential privacy

Differential Privacy Variant	Parameters	Benefit over ε-DP
ε-DP (Pure DP)	ε	—
(ε,δ)-DP (Approximate DP)	ε,δ	Gaussian mech.; mechanisms with catastrophic failure
Rényi DP (RDP)	α,ε	Gaussian mech.; precision; no catastrophic failure
Zero-Concentrated DP (zCDP)	ρ	Gaussian mech.; precision; no catastrophic failure
Gaussian DP (GDP)	μ	Gaussian mech.; precision; no catastrophic failure

possible. In other words, catastrophic failure is so unlikely that it is never expected to occur [22]. A common recommendation is to set $\delta \leq \frac{1}{n^2}$. Another sensible approach is $\delta \leq \frac{1}{n \log n}$. Neither of these approaches is appropriate for small values of n. Typically, values of δ exceeding 10^{-5} are suspicious and should be justified carefully if used.

For many mechanisms, the variants in Table 1 provide the same (or better) utility as (ε,δ)-differential privacy without the possibility of catastrophic failure, and these variants should be used when possible. However, some useful mechanisms do have catastrophic failure modes, and thus require the use of (ε,δ)-differential privacy. These mechanisms can offer unique utility benefits. One example is determining the set of bar chart bins from the data—see Sec. 3.4.1 for details. When such mechanisms are needed to support a desirable use case, then (ε,δ)-differential privacy must be used, and δ should be set so that $\delta \leq \frac{1}{n^2}$ to avoid catastrophic privacy failures.

Interpreting guarantees.

Each of the variants in Table 1 has a different set of privacy parameters, and measures privacy loss in a different way. Even when the parameters overlap, parameters with the same name can have different meanings. For example, the ε in Rényi differential privacy is only similar to the ε in pure ε-differential privacy when α is very large. Given these differences, how can one interpret and compare guarantees in different variants?

The most precise approaches for interpreting and comparing differential privacy guarantees are based on hypothesis testing [23, 24] or interpretation via Bayesian or frequentist semantics [14]. These approaches allow direct, precise interpretation of the privacy guarantee. For example, the formal definition of pure ε-differential privacy (Definition 1) can be viewed as bounding the error rates in a hypothesis test over the output of a differentially private mechanism:

- H_0: The mechanism's input dataset was D_1
- H_1: The mechanism's input dataset was D_2

In this framework, the privacy guarantee can be described in terms of hypothesis testing error rates. The false positive rate (or Type I error rate) is the probability that the adversary guesses that H_0 is true, but in fact H_0 is false and H_1 is true. The false negative rate (or Type II error rate) is the probability that the adversary guesses that H_0 is false (thus H_1 is true), but in fact H_0 is true and H_1 is false.

Definition 1 implies that the error rates achieved by any adversary who observes the output of an ε-differentially private mechanism must obey:

- $\Pr[\textit{false positive}] + e^{\varepsilon} \Pr[\textit{false negative}] \geq 1$ **and**
- $e^{\varepsilon} \Pr[\textit{false positive}] + \Pr[\textit{false negative}] \geq 1$

This interpretation allows the adversary to trade off between false positives and false negatives, but when ε is small, the adversary cannot achieve low error rates for both error types simultaneously.

The other variants in Table 1 can also be viewed through the hypothesis testing lens, allowing direct and precise comparison between variants and precise interpretation of the privacy guarantee in common terms.

The main challenge of interpreting privacy guarantees this way is complexity. The hypothesis testing interpretation yields many possible tradeoffs between false positives and false negatives, and interpreting the guarantee requires considering all of them. Comparing two guarantees requires comparing this tradeoff across the whole range of error rates.

Wasserman and Zhou [25] describe a framework for comparing mechanisms that provides precise comparisons by leveraging the semantics of the privacy guarantee directly. When comparing two mechanisms, this approach can provide a precise comparison whose result is easy to interpret.

Both of these approaches currently require significant technical expertise to apply and interpret.

Key Takeaway: Precise and direct interpretation and comparison of differential privacy guarantees can be performed via hypothesis testing interpretations and other direct interpretations of the privacy guarantee's semantics. Correct application of these approaches currently requires significant technical expertise.

Interpreting guarantees via conversion.

Guarantees given in two different variants can also be interpreted and compared by converting them to a common format. All of the variants in Table 1 can be converted to (ε, δ)-differential privacy for comparison, as shown in Fig. 4. This approach is simpler than the direct interpretations mentioned earlier, but can be significantly less precise.

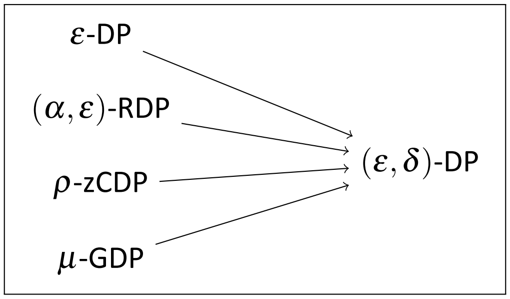


Fig. 4. All of the differential privacy variants shown in Table 1 can be converted to (ε,δ)-differential privacy.

Key Takeaway: Rényi differential privacy, zero-concentrated differential privacy, and Gaussian differential privacy guarantees can be converted to (ε,δ)-differential privacy guarantees to enable a simplified interpretation and comparison between them. The conversion is both loose and lossy, and must be done with care.

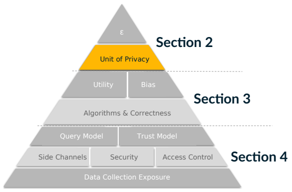
There are two important limitations of interpreting guarantees by converting to (ε,δ)-differential privacy. First, the conversion is loose—the original privacy parameter(s) provide a more accurate upper bound on privacy loss than the converted ε and δ, and the difference is sometimes significant. This effect can cause the conversion result to communicate a more pessimistic view of privacy loss than the original parameter.

Second, the conversion is lossy—when performing the conversion, the analyst chooses a value for δ and calculates ε; each guarantee in these variants corresponds to many possible (ε,δ) pairs. This effect means that choosing a large δ can result in a misleading optimistic value for ε. For example, a zero-concentrated differential privacy guarantee with $\rho = 0.1$ corresponds to infinitely many (ε,δ)-differential privacy guarantees, including both $\varepsilon = 1.45, \delta = 10^{-2}$ and $\varepsilon = 4.39, \delta = 10^{-20}$.

Due to these limitations, conversion to (ε,δ)-differential privacy is not the most precise method for interpreting guarantees and should not be used as a replacement for reporting the original privacy parameters. When performing conversions, analysts should set $\delta \leq \frac{1}{n^2}$, as described earlier.

Key Takeaway: When converting a guarantee to (ε,δ)-differential privacy, set $\delta \leq \frac{1}{n^2}$. When reporting guarantees, report all of the original privacy parameters to allow third parties to perform more precise interpretation of the guarantee.

2.4. The Unit of Privacy



The second layer of the differential privacy pyramid (Fig. 1) is the *unit of privacy* for a differential privacy guarantee. Definition 1 defines differential privacy in terms of *neighboring datasets* and says that two datasets D_1 and D_2 are neighbors if they differ in one person's data. This is an informal description, and how it is formalized significantly impacts the actual meaning of a differential privacy guarantee. The formal definition of neighboring datasets in a differential privacy guarantee implies a real-world unit of privacy that specifies exactly what is protected by the guarantee. In many ways, it is just as important to real-world privacy as the setting of the privacy parameters.

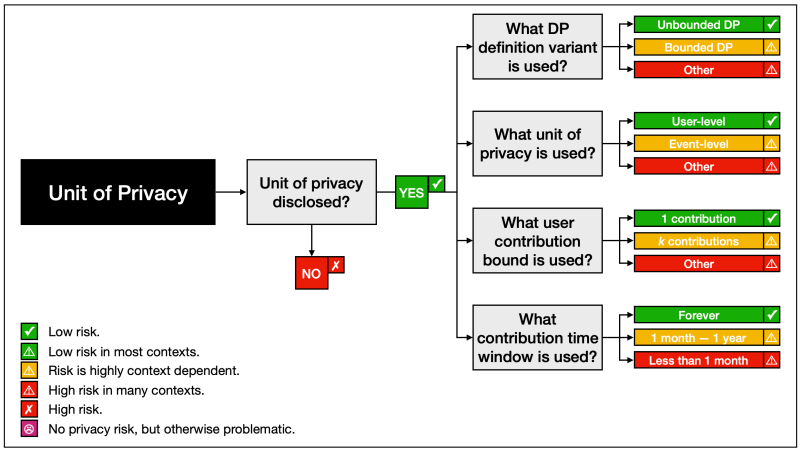

Unit of Privacy
Unit of privacy disclosed?
YES ✓
NO ✗
What DP definition variant is used?
Unbounded DP ✓
Bounded DP ⚠
Other ⚠
What unit of privacy is used?
User-level ✓
Event-level ⚠
Other ⚠
What user contribution bound is used?
1 contribution ✓
k contributions ⚠
Other ⚠
What contribution time window is used?
Forever ✓
1 month — 1 year ⚠
Less than 1 month ⚠
✓ Low risk.
⚠ Low risk in most contexts.
⚠ Risk is highly context dependent.
⚠ High risk in many contexts.
✗ High risk.
☹ No privacy risk, but otherwise problematic.

Fig. 5. An example flowchart for determining whether or not a unit of privacy is low or high risk. Actual values will depend on the context.

Fig. 5 shows a potential flowchart for determining the unit of privacy. The unit of privacy has two components that together determine the formal definition of neighboring datasets: (1) what it means for two datasets to "differ," and (2) a definition of "one person's data." This publication focuses on the second component of the unit of privacy, since some settings for this component can significantly weaken the privacy guarantee. *User-level privacy* provides a strong guarantee, and is the best default setting for this component.

2.4.1. Bounded and Unbounded Differential Privacy

What does it mean for two datasets to "differ"? The two most common definitions are called *unbounded differential privacy* and *bounded differential privacy* [26]. In unbounded differential privacy, two datasets D_1 and D_2 differ in one person's data if it is possible to construct D_2 from D_1 by **adding or removing** one person's data. In unbounded differential privacy, D_1 and D_2 have different sizes, because of the addition or removal of one person's data. In bounded differential privacy, two datasets D_1 and D_2 differ in one person's data if it is possible to construct D_2 from D_1 by **changing** one person's data. In bounded differential privacy, D_1 and D_2 are the same size.

Both unbounded differential privacy and bounded differential privacy can provide robust privacy in many cases, but there are subtle differences that are important in some contexts. Most importantly, for a dataset D of size n, all of D's neighbors under bounded differential privacy also have size n; this property means that a mechanism which releases the dataset size n without any noise at all can still satisfy differential privacy (see Sec. 3.1 for details). Thus if the size of the dataset is considered sensitive—or if criteria for inclusion into the data are sensitive—then bounded differential privacy is not a good choice.

Mechanisms that satisfy unbounded differential privacy also satisfy bounded differential privacy (via a conversion that increases the privacy parameter by a scaling factor), and also protect the size of the dataset n. Unbounded differential privacy is therefore a safer choice, and should be used when possible. However, some mechanisms cannot be proven differentially private under unbounded differential privacy, and thus require the use of bounded differential privacy. When these mechanisms are used, it is important to consider that the deployed system may reveal the total size of the dataset. Another constraint to be aware of is that the size of the dataset n is always considered sensitive when using unbounded differential privacy, and therefore should not be used to set public parameters, e.g., to set the δ parameter using the formula $\delta = \frac{1}{n}$. When using bounded differential privacy, the size of the dataset n is never considered sensitive.

2.4.2. Defining One Person's Data

How does one define "one person's data" formally? The answer depends on the underlying assumptions for a given scenario's trust model, and it defines exactly what is protected by the differential privacy guarantee. This section describes several common choices for this component of the unit of privacy and their implications for real-world privacy harms.

Unit of Privacy: Event Level

Privacy Hazard: Event-level privacy protects events, rather than people, and can result in surprisingly weak privacy guarantees.

To see why the unit of privacy is so important, consider how one would determine whether D_1 and D_2 are neighboring datasets in the earlier example scenario of the number of pumpkin spice lattes sold in October. One could say that D_1 and D_2 are neighbors if they differ in one event (e.g., a single transaction). This is an easily formalized definition and is sometimes called *event-level privacy*. It is also sometimes called row-level differential privacy because single events often translate directly to single rows in a database.

To think about how this unit of privacy impacts the real-world privacy of individuals, imagine a scenario in which a particularly thirsty customer (Customer X) buys 610 of the 632 pumpkin spice lattes sold in October. Imagine that an adversary knows the identities and purchase history of all of the pumpkin spice latte customers except for Customer X and wants to find out whether Customer X purchased a small number of pumpkin spice lattes (e.g., fewer than 30) or a large number (e.g., more than 200). The adversary might be able to figure out which of these two hypothetical situations is the real one, even if differential privacy is used because the full strength of the differential privacy guarantee applies only to neighboring datasets. Under the event-level unit of privacy, the datasets associated with the adversary's hypotheses are not neighbors. The event-level unit of privacy says that neighboring datasets differ by one event (i.e., by a single pumpkin spice latte transaction), and the adversary's hypotheses differ by much more than this. The event-level unit of privacy does protect against an adversary who wants to know whether Customer X bought 632 or 633 pumpkin spice lattes because the associated datasets are neighbors under this unit of privacy.

Event-level privacy can result in surprisingly weak privacy guarantees, since it protects individual events rather than people. There are cases when event-level privacy makes sense, but use of event-level privacy requires careful consideration.

Unit of Privacy: User Level

For a stronger real-world guarantee, one can use a different unit of privacy: D_1 and D_2 are neighbors if they differ in one user's data. This definition of neighboring datasets is called *user-level privacy*. Under this unit of privacy, the adversary's hypotheses about Customer X are represented by neighboring datasets. In fact, any dataset where Customer X purchases n pumpkin spice lattes is a neighbor of a dataset where Customer X purchases m lattes for any values of n and m (for bounded differential privacy), or of a dataset where Customer X is not present at all (for unbounded differential privacy). Thus, differential privacy does translate to a meaningful real-world privacy guarantee against the adversary discussed above if the unit of privacy is set correctly.

Other Units of Privacy

More complex units of privacy are sometimes used in practical deployments. For example:

- Attribute-level privacy protects a specific attribute of each individual in the data: D_1 and D_2 are neighbors if they differ in a single user's attribute. For example, gender-level privacy would protect against an adversary whose only goal is to find out someone's gender (assuming all other attributes that correlate with gender have been accounted for).
- User-day-level (or week, month, etc.) privacy protects activities of each individual in a specific time period: D_1 and D_2 are neighbors if they differ in a single user's data taking place on the same day. This is often used when the input dataset grows

over time, and the data are regularly shared or published. In this model, differential privacy guarantees hold under the assumption that data in each time window is fully independent, which may be an unrealistic assumption.

In some contexts (e.g., social network data, establishment statistics, or location data), the unit of privacy must be adapted to the context in a domain-specific way [27]. In use cases that involve sharing or publishing multiple statistics along several distinct dimensions of the data, it is often useful to measure privacy loss with multiple units of privacy, to provide a more complete picture of the overall guarantees. For example, a mechanism could provide both gender-level privacy with $\varepsilon = 0.2$, age-level privacy with $\varepsilon = 0.3$, and user-level privacy with $\varepsilon = 1$. Finally, in some use cases, the privacy loss associated with a specific unit of privacy can vary depending on the value of each record [28, Sec. 5]. This is useful in cases that involve heavily-skewed data, for example when computing statistics about companies [27].

Complex units of privacy like these are difficult to interpret and can weaken the privacy guarantee in surprising ways. For example, a user-day privacy guarantee with $\varepsilon = 1$ may appear to be a strong guarantee (due to the small value of ε), but the total privacy loss may be $\varepsilon = 365$ over the course of an entire year. A guarantee that protects gender with $\varepsilon = 0.2$ may fail to provide any protection for proxies of the gender attribute. Complex units of privacy thus require careful scrutiny due to their potential for unintentionally revealing sensitive information.

Key Takeaway: *user-level privacy* provides stronger guarantees than attribute-level, user-day-level, or event-level privacy, and a privacy unit at least as large as an individual user should be used when possible.

Transforming the Unit of Privacy: Bounding Contributions

A common way to achieve user-level privacy when each user submits multiple events is to enforce an upper bound on the number of events contributed by each user by transforming the data (e.g., keeping the first k events they submit and throwing away any further events or by keeping a random size-k subset of their events). Approaches like this are used to bound the contributions made by each user.

Bounding contributions transforms the unit of privacy from the event level to the user level, but it also scales up the sensitivity (described in Sec. 3.1) of operations on the data by the upper bound k. As a result, user-level guarantees achieved by bounding contributions require more noise for the same value of ε, and k should be set carefully to maximize accuracy.

Bounding contributions can also be used to achieve other kinds of privacy units. For example, it is possible to enforce an upper bound of k events per user per day (or other unit of time) or per location (or other unit of geography). These guarantees tend to be stronger than event-level privacy but weaker than user-level privacy, and their strength can be difficult to interpret (see Sec. 2.5).

Evaluating the Unit of Privacy

To determine whether a unit of privacy is sufficient, start with the user-level unit of privacy. Then consider possible real-world privacy loss risk, and evaluate whether the unit of privacy makes guarantees in the associated scenarios.

Privacy loss risk can be defined in terms of pairs of hypothetical situations that an adversary would like to distinguish (i.e., they would like to know which hypothesis is true). The example above described a potential privacy loss in terms of two hypotheses:

1. Customer X purchased fewer than 30 pumpkin spice lattes.
2. Customer X purchased more than 200 pumpkin spice lattes.

Now, consider the datasets D_1 and D_2 associated with the two hypothetical situations. D_1 will contain fewer than 30 transactions from Customer X, while D_2 will contain more than 200 transactions.

Privacy Hazard: If the difference between two hypothetical situations is not captured by the unit of privacy, then differential privacy may not prevent an adversary from distinguishing the two situations.

If these two datasets are neighbors based on the chosen unit of privacy, then the differential privacy guarantee applies to the underlying privacy loss. If they are not, then differential privacy makes no direct guarantee about the privacy loss. In the previous example, the event-level unit of privacy means that D_1 and D_2 are not neighbors, so differential privacy makes no direct guarantees about this situation. Under the user-level unit of privacy, the two are neighbors. In some cases, privacy guarantees can be converted from one unit of privacy to another. While conversion can be used to establish guarantees for an appropriate unit of privacy from an inappropriate one, it typically loses considerable precision or leads to a guarantee that is too weak to be meaningful. In the case of group privacy, conversion from user-level privacy does not lose precision, and so it is often calculated from a user-level privacy guarantee as needed.

Choosing a Unit of Privacy

Privacy Hazard: *user-level privacy* is a strong setting for the unit of privacy. Other settings may provide significantly weaker protection in practice.

The user-level unit of privacy is an excellent default and generally provides robust real-world privacy. Relaxing the unit of privacy can improve accuracy and reduce ε and δ simultaneously, but it can also lead to surprising real-world privacy failures. In particular, it may be possible to learn a significant amount about an individual's habits when event-level privacy is used.

Example scenarios that highlight the impact of event-level privacy include:

- Event-level privacy for website logs protects a single visit to a URL but not repeat visits.
- Event-level privacy for taxi trip data protects a single trip but not an individual's common destinations (e.g., home or work).
- Event-level privacy for smart meters protects a single meter reading but not trends in electricity use (e.g., the use of power-hungry Bitcoin mining equipment).

Bounds on user contributions can strengthen the privacy guarantee significantly, but the bounds must be selected carefully. A total contribution limit is strongest and equivalent to user-level privacy. Bounds that reset periodically can be much weaker.

Example scenarios that highlight the impact of bounding contributions include:

- A total contribution limit is equivalent to user-level privacy and generally provides robust real-world privacy.
- A per-day contribution limit protects activities in a single day but not activities that repeat across multiple days.
- A per-month contribution limit protects activities in a single month but not activities that occur every month.

The safest default for any differential privacy guarantee is user-level privacy or a total contribution bound that transforms the guarantee into user-level privacy. Weaker units of privacy can improve accuracy or reduce ε, but they can also weaken the privacy guarantee significantly. When a weaker unit of privacy is used, it is important to assess whether the differential privacy guarantee still offers the desired protection against real-world privacy risks. For example, group privacy is a unit of privacy that naturally maps directly to real-world privacy risks in some settings.

2.5. Comparing Differential Privacy Guarantees

This section demonstrates the implications of different kinds of differential privacy guarantees by comparing different guarantees to each other.

Privacy Parameter ε

The setting of the privacy parameter ε has the most visible impact on real-world privacy, and comparing ε values is the first step in comparing two guarantees. For example, a pure ε-differential privacy guarantee (i.e. $\delta = 0$) with $\varepsilon = 0.1$ is strictly stronger than a guarantee with $\varepsilon = 10$.

ε	2.5
δ	$1 \cdot 10^{-25}$
Privacy Unit	User level

(a)

ε	2.5
δ	$1 \cdot 10^{-5}$
Privacy Unit	User Level

(b)

Privacy Hazard:
Guarantees with different values of δ are not directly comparable.

Fig. 6. An example of two differential privacy guarantees that have the same ε value. The two guarantees are not directly comparable because they have different δ values.

ε	2.5
δ	$1 \cdot 10^{-5}$
Privacy Unit	User Level

(a)

ε	2.5
δ	$1 \cdot 10^{-5}$
Privacy Unit	Event Level

(b)

Privacy Hazard:
Guarantees with different units of privacy are not directly comparable.

Fig. 7. An example of two differential privacy guarantees that have the same ε and δ values. The two guarantees are not directly comparable because they have different units of privacy.

Privacy Parameter δ

Because of the direct relationship between ε and δ described earlier, it is usually not possible to directly compare two (ε, δ)-differential privacy guarantees when their δ values differ. For example, consider the two guarantees in Fig. 6. Their ε values are the same, but their δ values are different, so they are not directly comparable. If the ε and δ values result from conversion from another variant of differential privacy, the original privacy parameters should be used to make the comparison.

In practice, it is common to use the ε value alone to get a rough sense of the strength of the privacy guarantee or to roughly compare two guarantees (after confirming that $\delta \leq \frac{1}{n^2}$). As described earlier, this approach can be imprecise.

Unit of Privacy

An improper setting for the unit of privacy can unintentionally reveal information about individuals. For example, consider the two guarantees in Fig. 7. Guarantee **(a)** is strictly stronger because its unit of privacy is strictly larger even though the other parameters are the same for both guarantees. Guarantee **(b)** may not provide meaningful privacy when one person's data contributes to many events, and as a result an attacker may be able to determine sensitive information about the individual, in spite of the differential privacy guarantee.

ρ	0.1
ε	1.45
δ	$1 \cdot 10^{-2}$
Privacy Unit	User Level

(a)

ρ	0.1
ε	4.39
δ	$1 \cdot 10^{-20}$
Privacy Unit	User Level

(b)

Fig. 8. An example of two differential privacy guarantees that have different ε and δ values. The two guarantees are directly comparable because one is convertible to the other using a conversion formula.

Privacy Hazard: When converting a guarantee to (ε,δ)-differential privacy, choosing a large value for δ results in a misleading value for ε.

Conversion Between Variants

Converting to (ε,δ)-differential privacy from another variant of the differential privacy definition requires picking a value for δ. In this situation, the δ parameter is important for interpreting the resulting ε and comparing it with other guarantees. For example, consider the two guarantees in Fig. 8. Guarantees **(a)** and **(b)** are equivalent even though the reported ε values are very different. The difference comes from the trade-off between ε and δ in the conversion process from zero-concentrated differential privacy—a larger δ allows for a smaller ε, and a smaller δ requires a larger ε.

When a variant is converted to (ε,δ)-differential privacy, the original privacy parameters should also be given (e.g., for zero-concentrated differential privacy, the value of ρ). This information allows third parties to perform their own conversion with other values for δ, enabling direct comparison with other guarantees.

2.6. Mixing Differential Privacy With Other Data Releases

In some contexts, it may be necessary to release both differentially private statistics and non-differentially private statistics calculated from the same underlying data. For example, an organization may wish to make two releases based on the same underlying data:

Privacy Hazard: The use of differential privacy does not mitigate privacy risks associated with other (non-differentially private) releases based on the same underlying data.

1. Exact summary statistics without differential privacy (under the assumption that the associated privacy risk is low, even without differential privacy)
2. Detailed statistics with differential privacy

When evaluating privacy risk, it is important to consider the total impact of all releases. In particular, the use of differential privacy in the second release does not improve privacy for the first release. In situations like this, it is important to consider the total privacy risks of all releases (e.g., using the NIST Privacy Risk Assessment Methodology [10]).

In this setting, it is possible to ensure consistency between the two releases by "post-processing" the differentially private release. This involves modifying the differentially private release to make it consistent with the non-differentially private release. It is important to note that this kind of post-processing does not fall under differential privacy's *post-processing invariance* guarantee—even though the same terminology is often used—because it leverages the original sensitive data to perform the post-processing. Post-processing of this kind thus does not satisfy differential privacy.

2.7. Auditing and Empirical Measures of Privacy

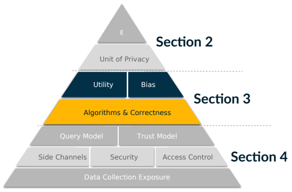
A number of approaches for *privacy auditing* have been developed that test the level of privacy provided by an implementation of differential privacy experimentally. These approaches can be used for query-answering and data release systems [29–34], and for machine learning systems [35–37]. They typically work by running the algorithm being tested many times to determine if the distribution of results satisfy the differential privacy definition. Approaches for auditing are an example of empirical methods for measuring privacy risk.

Empirical approaches, including auditing, cannot prove that a system correctly provides a desired differential privacy guarantee. However, auditing approaches can be helpful in finding implementation bugs: if the auditing procedure finds a counterexample, then the system under test definitely does not provide the desired privacy guarantee.

The results of auditing procedures can be difficult to interpret. For example, some approaches report average-case results, which can significantly underestimate the privacy risk to outliers in the dataset and result in false confidence in the system's privacy protection. Aerni et al. [38] describe the pitfalls associated with empirical measurement of privacy in machine learning systems; many of their conclusions also apply in other contexts. When used carefully, auditing approaches can be effective tools to help find bugs and supplement (rather than replace) privacy proofs.

3. Differentially Private Algorithms

This section describes specific algorithms for differentially private analysis. It focuses on high-level descriptions of established approaches with a particular emphasis on algorithms that are practical and easy to deploy. The first three sections describe important general considerations of differentially private algorithms, including utility and bias:



- Sec. 3.1 gives an overview of several building blocks used in differentially private algorithms.
- Sec. 3.2 describes utility, and accuracy, and some methods for measuring them.
- Sec. 3.3 explores the impacts that some differentially private algorithms have on different forms of bias in data releases.

Thereafter, the sections are organized by analysis type:

- Sec. 3.4 describes techniques for analytics queries on a single data table (e.g., counting, summation, and average queries).
- Sec. 3.5 describes techniques for machine learning, including deep learning.
- Sec. 3.6 describes techniques for generating differentially private synthetic data.
- Sec. 3.7 discusses unstructured data (e.g., text, photos, and video).

This section describes some specific differentially private techniques to give practitioners a basic idea of how differential privacy is implemented and to highlight the impact of implementation choices on utility, bias, and other factors. NIST strongly recommends that practitioners use well-tested implementations provided by libraries rather than implementing these mechanisms and algorithms themselves. As discussed in Sec. 4, implementing differentially private algorithms can be tricky, and custom implementations increase the risk of privacy vulnerabilities.

Privacy Hazard: Avoid custom implementations of differentially private algorithms, and use well-tested libraries instead.

3.1. Basic Mechanisms and Common Elements

Randomized functions (often called mechanisms) are used to achieve differential privacy. If Definition 1 is proven for a mechanism, it is called a differentially private mechanism.

This section describes two basic differentially private mechanisms that are often used to build larger mechanisms and systems: the Laplace mechanism and the Gaussian mechanism. Both work by adding noise to the output of a query, and both mechanisms scale the noise according to the *sensitivity* of the underlying query. Sensitivity is defined to measure how much the output of a query could change when its input (i.e., the data being queried) changes. Two commonly used sensitivity measures are L_1 and L_2. The L_1 sensitivity is measured using L_1 distance (i.e., Manhattan distance), while the L_2 sensitivity is measured using L_2 distance (i.e., Euclidean distance). See Appendix Sec. B.2 for the formal definitions.

Key Takeaway: The *sensitivity* of a query is designed to measure how much the query output could change as a function of how much the input could change.

Mechanism: Laplace Mechanism [22]. The *Laplace mechanism* adds random noise drawn from the Laplace distribution to the output of a query. It uses L_1 sensitivity and guarantees $(\varepsilon, 0)$-differential privacy.

Mechanism: The *Gaussian mechanism* adds random noise drawn from the Gaussian (or normal) distribution to the output of a query. It uses L_2 sensitivity and guarantees privacy in a number of different variants of differential privacy, including the ε, δ variant.

Choosing a Mechanism

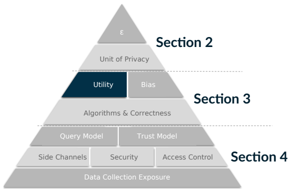
While both the Laplace and the Gaussian mechanisms add noise to a query's output to satisfy differential privacy, they differ in two major ways: the guarantee they provide and the measure of sensitivity they require.

The Laplace mechanism satisfies all of the differential privacy variants in Table 1, including pure ε-differential privacy. The Gaussian mechanism satisfies all of the variants in Table 1 **except** pure epsilon differential privacy. If the stronger pure ε-differential privacy guarantee is required, then the Gaussian mechanism is not an option.

If either guarantee is sufficient, then the choice can be made based on which mechanism provides better accuracy. For queries with *low-dimensional* outputs (i.e., for a query $f : D \to \mathbb{R}^k$ for small k, including $k = 1$), the Laplace mechanism often provides better accuracy. For queries with *high-dimensional* outputs (i.e., large k), the Gaussian mechanism often provides better accuracy because it allows the use of L_2 sensitivity. For high-dimensional outputs, L_2 sensitivity is typically much smaller than L_1 sensitivity, which significantly improves accuracy. These general properties do not always hold, and the most accurate mechanism depends on the privacy parameters, sensitivity, and chosen accuracy metrics; see one example in the next section. Exact calculations can be made based on these values to select the most accurate mechanism for a specific situation.

3.2. Utility and Accuracy

Utility refers to how useful a dataset or statistic is for a specific purpose. *Accuracy* refers to the difference between a mechanism's output and the true value that it is attempting to estimate. The two are not synonymous, even though they are often used interchangeably. Utility depends on the way a statistic will be used, while accuracy is simply a measurement of the statistic's error. In particular, data can be:



- **Accurate but not useful**. For example, if a mean was provided very accurately, but a 95th percentile was required.
- **Inaccurate but still useful**. For example, an inaccurate statistic may be sufficient to demonstrate a difference between two populations if the difference is very large.

The essence of what makes a utility assessment low or high risk is distilled in a flowchart, shown in Fig. 9.

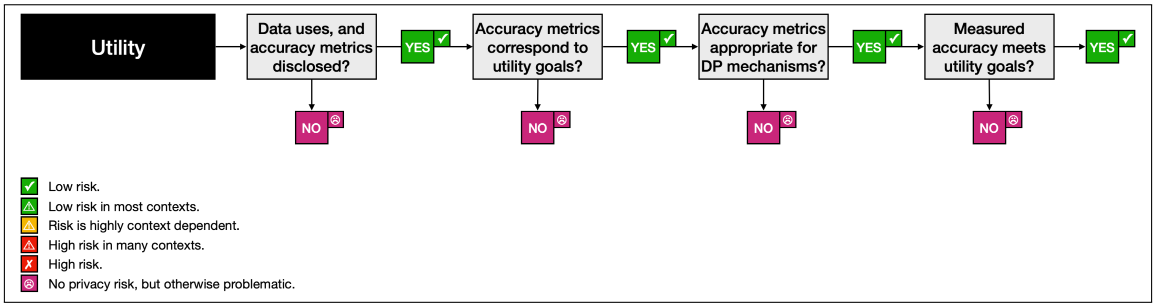

Utility
Data uses, and accuracy metrics disclosed?
YES ✓
NO ☹
Accuracy metrics correspond to utility goals?
YES ✓
NO ☹
Accuracy metrics appropriate for DP mechanisms?
YES ✓
NO ☹
Measured accuracy meets utility goals?
YES ✓
NO ☹
✓ Low risk.
⚠ Low risk in most contexts.
⚠ Risk is highly context dependent.
⚠ High risk in many contexts.
✗ High risk.
☹ No privacy risk, but otherwise problematic.

Fig. 9. A flowchart for determining whether or not a utility assessment is low or high risk.

Metrics for Utility: No General Solution

A statistic or data release can be used to answer many different questions. If the questions are known in advance, it is sometimes possible to develop *outcome-specific utility metrics* that directly measure the utility of the data for answering the specific questions of interest.

Often the specific questions of interest are not known when the data or statistics are created, so designing outcome-specific metrics based on those questions is not possible. Moreover, no single metric (or group of metrics) applies to all questions.

A number of different metrics have been developed that attempt to approximately measure utility for large classes of questions [39]. These metrics combine measures of accuracy with assessments of properties that are typically of interest to statisticians, like correlations between columns in the data. Such metrics are useful tools for evaluating the quality of differentially private statistics or data releases. However, most utility metrics do not necessarily ensure utility for all possible questions of interest. This is by design, as it is not possible in general to achieve strong utility and privacy for all possible queries of interest.

Metrics for Accuracy

Because utility is difficult to measure directly, accuracy metrics are often used as a proxy for utility. Two common accuracy metrics are absolute error and relative error. *Absolute error* is simply the absolute difference between the true query result and the noisy one. *Relative error* is the absolute error divided by the true query result.

This setting poses a challenge to measuring error: the mechanisms used for differential privacy often add random noise to query results, and that noise is, in theory, unbounded (i.e., it has no maximum or minimum). For example, it is possible to draw a Laplace noise sample in the millions or billions, but it is extremely unlikely. To get an idea about how much error is likely to be seen when running the mechanism, one can use a confidence interval. For example, a 95% confidence interval says that the absolute error of the mechanism will lie within the specified interval 95% of the time. If this interval is small, then one can be confident that the mechanism will give an accurate answer most of the time.

For example, the Laplace mechanism described earlier can be measured by bounding the absolute error of the mechanism due to the noise it adds. The absolute error for the Laplace mechanism is defined as $|f(x) - (f(x) + \mathsf{Lap}(\Delta_1/\varepsilon))|$. The noise depends on the privacy parameter ε. That is, the smaller the ε, the larger the error.

An example of a 95% confidence interval for the absolute error of the Laplace mechanism is shown in Fig. 10. In this example, the query $f(x)$ is an average, and the true result is $f(x) = 331$. The confidence interval is graphed as an error bar extending above and below the average. As ε gets smaller, the error bar becomes larger, meaning that the Laplace mechanism is more likely to return results with a larger error when ε is small.

The type of randomness used to achieve differential privacy can have important but subtle impacts on the way accuracy is measured. For example, the Gaussian distribution has lighter tails than the Laplace distribution—sampling a very large noise value is less likely when adding Gaussian noise than when adding Laplace noise. As a result, the best mechanism to use may be different depending on the desired confidence level:

- The 95% confidence interval may indicate that the Gaussian mechanism has lower error
- The 75% confidence interval may indicate that the Laplace mechanism has lower error

For precise comparisons between mechanisms, it is important to understand the interplay between the mechanism's design and the accuracy metrics chosen, since choosing a different metric may change the outcome of the comparison.

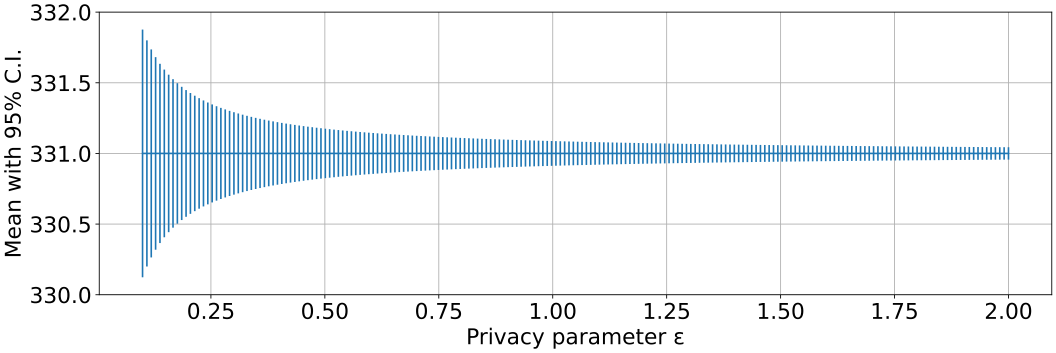


Fig. 10. The 95% confidence interval for the absolute error of the Laplace mechanism.

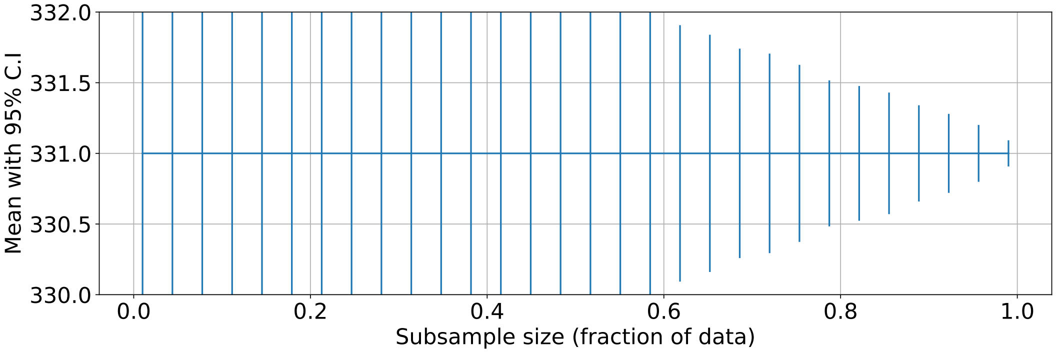


Fig. 11. A plot of subsample size vs the 95% confidence interval shown in Fig. 10.

Comparison With Subsampling

The error of the mechanism can be compared with some other approach that could be used to achieve privacy. One useful point of comparison is *subsampling*—computing the query's result using only a fraction of the original data selected at random and then measuring the error of that result against the true result. When only a small fraction of the original data are used, one can expect to obtain a less accurate result. The resulting "mechanism" does not satisfy differential privacy, but it has sometimes been used as an informal privacy mechanism.

Figure 11 plots a subsample size (measured as a fraction of the total dataset) against 95% confidence interval in the same way as Fig. 10. As the subsample size gets smaller, the confidence interval increases. This means that less accurate results can be expected with smaller subsamples. Note that the y-axis of this figure has the same scale as the earlier figure. The larger confidence intervals in the second image suggest that the Laplace mechanism can give much more accurate answers than subsampling in most settings.

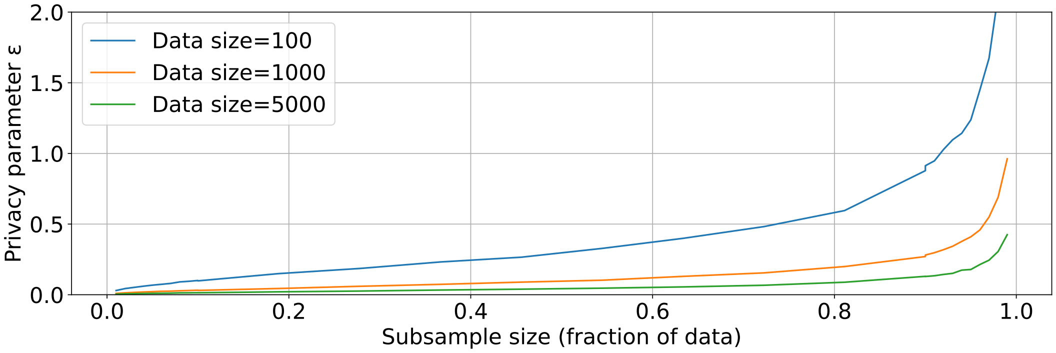


Fig. 12. A plot of subsample size vs epsilon values that give the same error confidence interval.

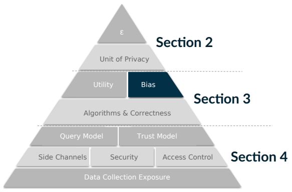
Subsampling can be directly compared with the Laplace mechanism by performing the following experiment: for a particular subsample size, consider the value of the privacy parameter ε that would have resulted in the same confidence interval as subsampling. The results are plotted in Fig. 12 with the subsample size on the x-axis and the value of ε required to achieve the equivalent confidence interval on the y-axis. These results show that even small values of ε suffice to match the accuracy of subsampling. Thus, in this case, the Laplace mechanism with commonly used privacy parameters around $\varepsilon = 1$ is likely to provide better accuracy than subsampling.

Monitoring Utility

Before publishing differentially private statistics, it is a good practice to check that the utility of the results correspond to what was expected. This step is particularly important when the data release is long-lived. For a single release, testing can be performed manually. When releasing many statistics or in automated releases over time, the testing process typically needs to be automated. In both cases, the testing process should check for software bugs and distributional shifts in the underlying data that might invalidate past assumptions made during mechanism design.

Checking utility in this way involves computing exact metrics from the private data, so it is not differentially private. It is therefore important to ensure that the testing process itself does not leak information about the private data. Testing results should not be released publicly, and should generally yield only a binary (“yes” or “no”) result. When the test fails, the root cause can be investigated manually and the issue fixed. Finer-grained approaches, like only releasing the parts of the data that have good enough accuracy, impact privacy guarantees much more negatively, and should be avoided.

3.3. Bias



Systems that process data can introduce or magnify various kinds of bias that can negatively impact the validity of conclusions drawn from the results. NIST Special Publication (SP) 1270, *Towards a Standard for Identifying and Managing Bias in Artificial Intelligence* [40], defines three important categories of bias:

- *Systemic bias* results from existing rules, processes, or norms that advantage certain social groups and disadvantages others.
- *Human bias* results from failures in the heuristics that humans use to make decisions.
- *Statistical bias* occurs when the expected value of an estimator differs from the true value of its parameter in the population.

Data-processing algorithms of all types have the potential to magnify or create all three types of bias, and differentially private algorithms have been shown to create one or more types of bias. This section describes how bias can result from the use of specific differentially private algorithms, and gives guidelines for understanding that bias and choosing alternative differentially private algorithms or mitigation measures. The essence of what makes a bias assessment low or high risk is distilled in a flowchart, shown in Fig. 13.

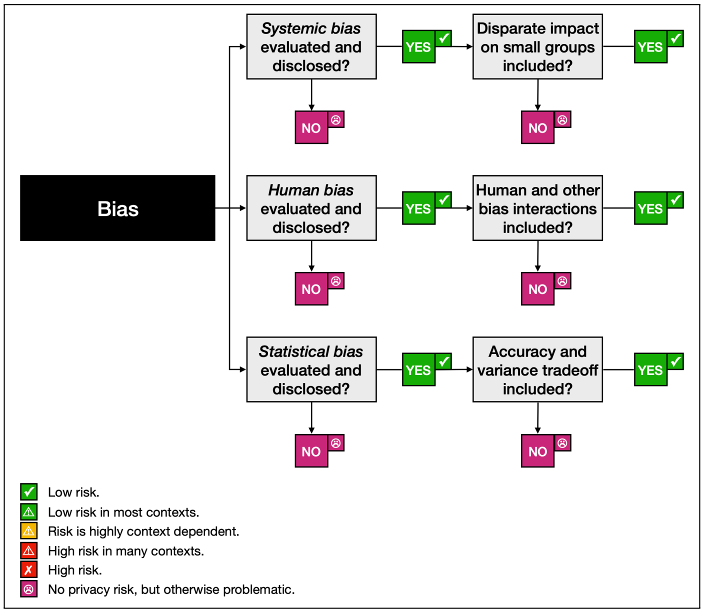

Bias
Systemic bias evaluated and disclosed?
YES ✓
NO ☹
Disparate impact on small groups included?
YES ✓
NO ☹
Human bias evaluated and disclosed?
YES ✓
NO ☹
Human and other bias interactions included?
YES ✓
NO ☹
Statistical bias evaluated and disclosed?
YES ✓
NO ☹
Accuracy and variance tradeoff included?
YES ✓
NO ☹
✓ Low risk.
⚠ Low risk in most contexts.
⚠ Risk is highly context dependent.
⚠ High risk in many contexts.
✗ High risk.
☹ No privacy risk, but otherwise problematic.

Fig. 13. A flowchart for determining whether or not a bias assessment is low or high risk.

3.3.1. Systemic Bias

Systemic bias results from rules, processes, or norms that advantage certain social groups and disadvantage others. The use of data can perpetuate and magnify systemic bias in many different contexts. This effect is perhaps most clearly visible in machine learning and other forms of artificial intelligence (AI), where numerous results have demonstrated the tendency of AI systems to "learn" and magnify systemic biases encoded in the data used to train them [40].

Recent work has also demonstrated that the use of differential privacy can make this problem worse [41]. In a relative sense, the noise introduced by differentially private algorithms impacts smaller groups more than larger ones. Since marginalized social groups are often smaller than advantaged ones (and are sometimes underrepresented in the underlying data), the noise can magnify or even create biases in the differentially private results.

Some differentially private algorithms can magnify disparate impacts on small groups. Figure 14 shows two bar charts that count population by race in a single U.S. Census Bureau district in Massachusetts [42]. Each figure includes error bars (in red) that demonstrate the 95% confidence interval for the error introduced by differential privacy noise on each bar chart bin. The only difference between the two figures is the value of the privacy parameter ε. As expected, the lower value of ε produces more error, so the error bars are proportionally larger. The y-axis is plotted on a logarithmic scale to accommodate the variation in bin sizes. Note that for the lowest population race (i.e., American Indian), the error bar is larger than the population when $\varepsilon = 1$. For the higher population races, the error bars are proportionally smaller than the populations for both values of ε. All of the error bars in each figure have the same absolute size (they only have different visual sizes because of the logarithmic scale). However, the same absolute error may disproportionately impact small groups. In this example, when $\varepsilon = 1$, there is a chance that the noise required by differential privacy will reduce the American Indian population to zero. For larger populations, this kind of extreme impact is virtually impossible.

Privacy Hazard: Differential privacy can magnify or create systemic bias.

Open Question: Finding and mitigating systemic bias is an open area of research. Users of this publication may find [40, 43–46] helpful for understanding the considerations.

This type of bias is an unavoidable property of any robust anonymization mechanism, and is not unique to differential privacy. In the example, the true population count for American Indian in the U.S. Census district is exactly 1: it refers to the data of a single person in the dataset. It is fundamentally impossible to both accurately convey this information and adequately protect the data of all individuals in the input dataset.

Differential privacy can also magnify disparate impacts in machine learning. Figure 15 shows the accuracy of a machine learning classifier trained on the same census data as the previous

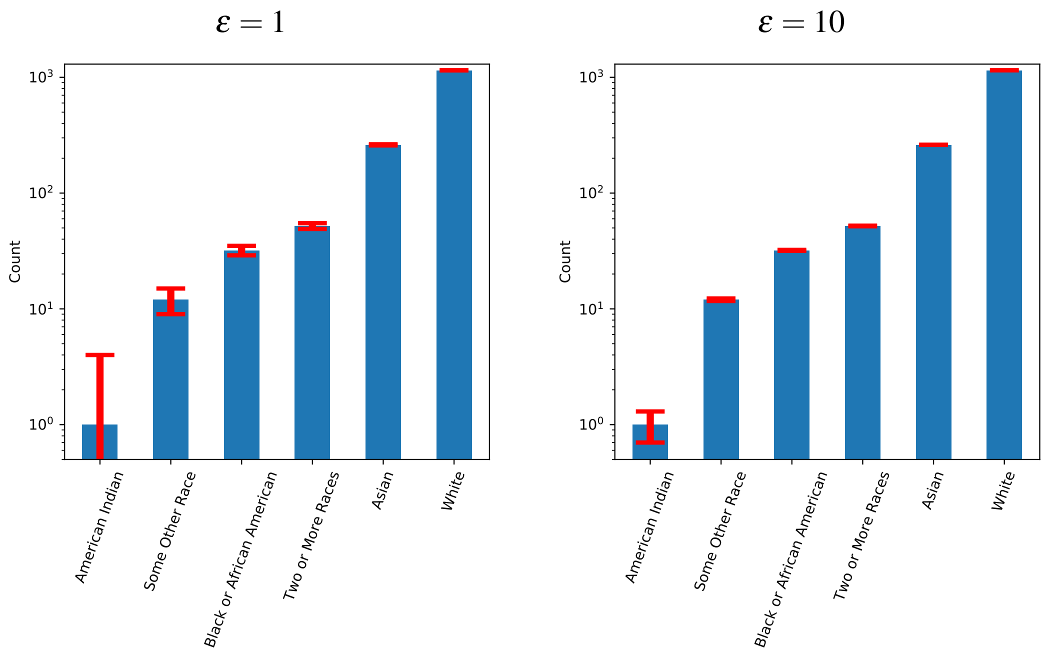


Fig. 14. Two bar charts of population count by race in a single U.S. Census Bureau district in Massachusetts computed with differential privacy for $\varepsilon = 1$ (left) and $\varepsilon = 10$ (right). Confidence intervals are displayed in red overlaying each bar.

example [42]. The classifier is trained to predict an individual's housing type (i.e., single family versus multi-family housing) from other attributes of that individual. Many classifiers with different values of ε were trained, and the accuracy of the trained classifiers was separately plotted for (1) the majority race in the data and (2) records with the race "Native Hawaiian and Other Pacific Islander." The results show that the classifier is actually *more* accurate for the minority group than for the majority group at very large values of ε, but the noise required for differential privacy affects the minority group much more than it does the majority group. In practice, this means that models trained with differential privacy may produce lower-quality outputs for minority groups. Perfect approaches for mitigating this effect are not known, but it is helpful to test carefully for output quality across all subgroups and to ensure balance between classes during training.

3.3.2. Human Bias

Human bias results from the heuristics that humans use to make decisions based on data. Common examples include confirmation bias (i.e., believing data that supports one's beliefs) and anchoring bias (i.e., believing the first piece of data received).

Human bias has the potential to negatively impact belief in the validity of differentially private results. In particular, individuals may believe that differentially private results are

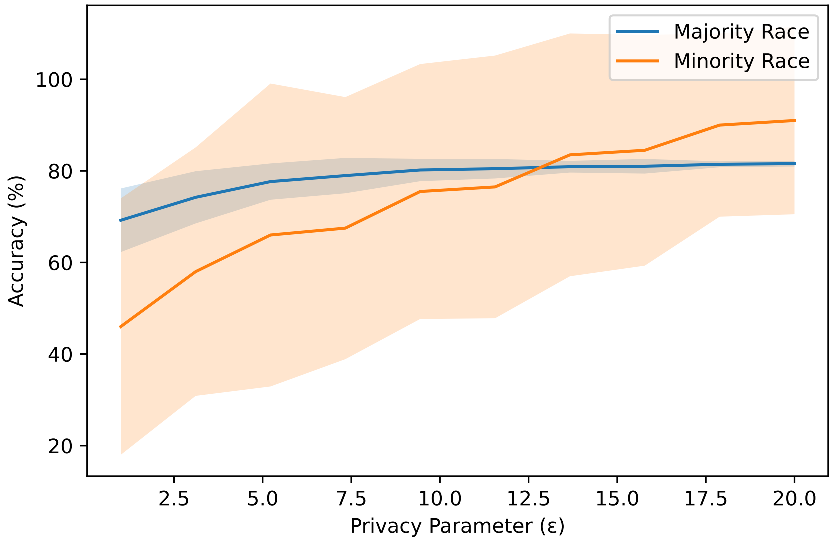


Fig. 15. Classifier accuracy for a machine learning classifier trained on U.S. Census data with differential privacy for various values of ε.

invalid because they know that noise has been added to the results or the results do not conform to typical expectations of what "good data" looks like (e.g., differentially private bar charts may contain fractional or negative counts).

Interventions that attempt to address potential human bias resulting from the use of differential privacy may actually introduce other kinds of bias. For example, differentially private counts are often rounded to the nearest integer and forced to be non-negative on the assumption that data recipients might be concerned by fractional or negative counts that do not "look like" non-differentially-private results. However, these changes can actually harm the results by introducing statistical bias.

Privacy Hazard: Before deploying interventions to address sources of human bias, carefully consider the other impacts of those interventions.

3.3.3. Statistical Bias

The statistical bias of a mechanism refers to a difference between the true query result $f(x)$ and the expected value (i.e., the average over many samples) of the mechanism's output. For example, the statistical bias of the Laplace mechanism is $\mathbb{E}[f(x) - \text{Lap}(\Delta_1/\varepsilon)] - f(x)$. The equation can be rearranged to $\mathbb{E}[\text{Lap}(\Delta_1/\varepsilon)]$, and the Laplace distribution centered at zero has an expected value of zero.

Privacy Hazard: Differential privacy mechanisms can introduce statistical bias. It is important to understand, quantify, and evaluate the statistical bias present in any differentially private data release.

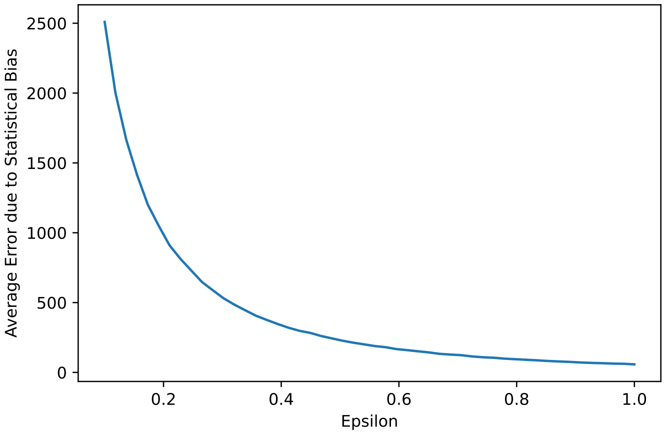


Fig. 16. A plot of average error due to statistical bias of changing negative counts to zero vs choice of ε.

However, not all differential privacy mechanisms are unbiased. Some mechanisms can introduce statistical bias (an example appears in Sec. 3.4.2). In addition, post-processing approaches designed to improve data quality or reduce human bias can also result in statistical bias. Statistical bias must be considered as part of a utility analysis of a mechanism.

Some post-processing approaches used with privacy-preserving mechanisms can result in statistical bias. Figure 16 shows the total absolute error due to statistical bias of changing negative counts to 0 in the bar chart example from Sec. 3.3.1. The results show that this bias increases as the privacy parameter ε decreases. This type of post-processing does not impact privacy, but can introduce a tradeoff between bias and variance that impacts utility.

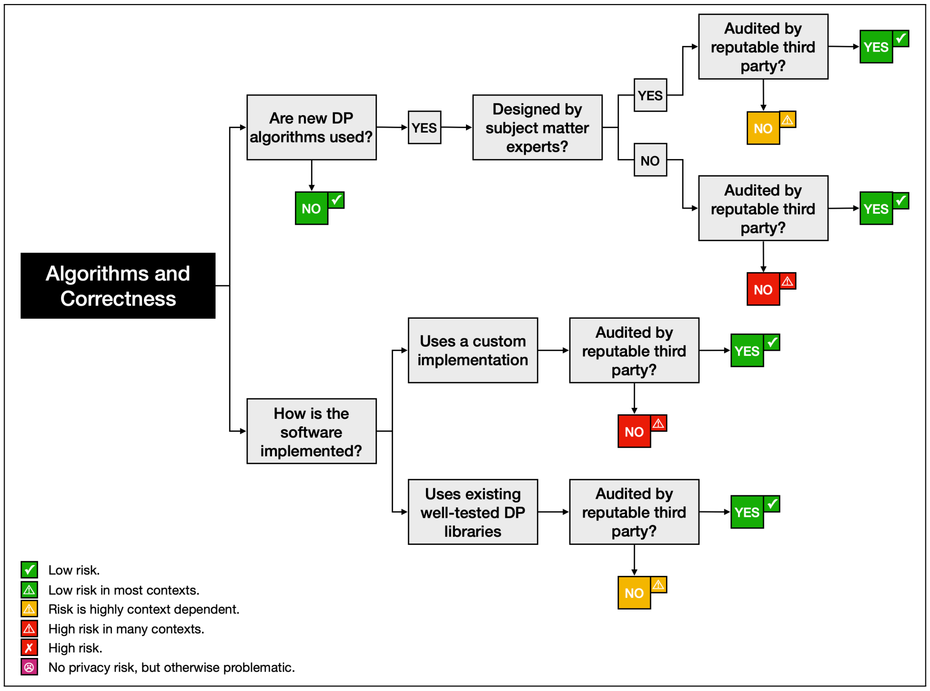
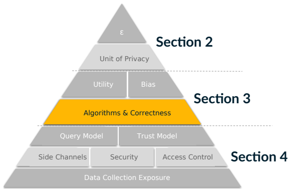

Algorithms and Correctness
Are new DP algorithms used?
NO ✓
YES
Designed by subject matter experts?
YES
Audited by reputable third party?
YES ✓
NO ⚠
NO
Audited by reputable third party?
YES ✓
NO ⚠
How is the software implemented?
Uses a custom implementation
Audited by reputable third party?
YES ✓
NO ⚠
Uses existing well-tested DP libraries
Audited by reputable third party?
YES ✓
NO ⚠
✓ Low risk.
⚠ Low risk in most contexts.
⚠ Risk is highly context dependent.
⚠ High risk in many contexts.
✗ High risk.
☹ No privacy risk, but otherwise problematic.

Fig. 17. A flowchart for determining whether or not an algorithm design and implementation process is low or high risk.

3.4. Analytics Queries

This section describes various algorithms for achieving differential privacy to build intuition for how differential privacy works, and for the degrees of freedom within the space of algorithm design. These descriptions alone are not suitable to use as the primary basis for implementing production-grade differentially private solutions. Where possible, existing libraries that are reputable and well-tested should be preferred to inventing one's own implementation, subject matter experts should be utilized when designing and implementing algorithms, and reputable third party auditors should be consulted to ensure designs and implementations are free of design errors, implementation errors and side channels. The essence of what makes an algorithm design and implementation process low or high risk is distilled in the flowchart shown in Fig. 17.

3.4.1. Counting Queries



This section describes how to answer counting queries with differential privacy. A *counting query* counts the number of rows in a dataset with a particular property. While they seem simple or trivial, counting queries are used extremely often and can express many useful business metrics, such as the number of transactions that took place in a given week or which market has produced the most sales.

Counting queries are often the basis for more complicated analyses as well. For example, the U.S. Census Bureau releases data that is essentially constructed by issuing many counting queries over sensitive raw data collected from residents.[9] Each of these queries belongs in the class of counting queries discussed in the following sections and computes the number of people living in the U.S. with a particular set of properties (e.g., living in a certain geographic area, having a particular income, belonging to a particular demographic).

Defining Counting Queries

Consider two examples of counting queries. The result of the first is a single number, and the second is a specific form of counting query called a bar chart that reports multiple counts derived from disjointed parts of the dataset.

Example: Counting Query. How many pumpkin spice lattes were purchased in October?

Example: Bar Chart. For each month, how many pumpkin spice lattes were purchased in that month?

Achieving Differential Privacy

To achieve differential privacy with counting queries, noise is added to the raw count that is proportional to the sensitivity of the query. Many counting queries have low sensitivity, and for these queries it is often possible to achieve high utility over a single table.

When bounding user contributions, more noise is required to compensate for the fact that each individual may contribute multiple records. Even in this case, it is often possible to achieve good utility for counting queries. See Appendix Sec. B.3 for technical details.

Privacy Hazard: When bounding user contributions, additional noise must be added to ensure user-level privacy.

Binned Data: Histograms & Time Series

For a bar chart, noise can be added to each "bin" of the result individually since each individual in the data will appear in exactly one "bin" of the result. However, there is a subtle but important difference: the result of a bar chart query reveals the identities of the bins in addition to the count for each one, and the presence or absence of a bin can

[9] While most of the primary Decennial Census releases (PL94-171 Redistricting, Demographic and Housing Characteristics products) consist of counts, this is not true of all Decennial Census releases, and even less so across all U.S. Census Bureau releases, which include non-counting queries such as medians, averages, and outputs of various models.

reveal information about an individual. Database systems commonly infer the set of bins from the data. For example, if no pumpkin spice lattes were purchased in June, then the resulting bar chart would not even contain a bin for June, thus implicitly revealing a "count" of zero pumpkin spice lattes with no noise at all.

To address this additional information leakage, the analyst must specify the set of bins in advance, and the bar chart must report a count for every bin in the set, even if the count is zero. Then, noise can be added to each count (including the zeros) and correctly satisfy differential privacy.

Specifying bins is an additional burden on the analyst that is not typical in traditional database query languages. Sometimes, specifying the bins is easy (e.g., if the bins are the months of the year). However, when the bins themselves are complex, the burden of specifying them manually can be significant. Techniques do exist for automatically determining the set of bar chart bins from the data without violating differential privacy [47], which can help to eliminate this additional burden. However, the accuracy of these techniques is data-dependent, which complicates understanding of utility.

Privacy Hazard: In differentially private bar charts, revealing the bar chart bins may violate privacy. To avoid this, the analyst must either determine the bins ahead of time before processing the data, or use specific algorithms that determine and reveal the bins without violating privacy.

Utility

For a single count, the Laplace mechanism generally yields better accuracy than the Gaussian mechanism (as described earlier, this is not always the case, and more precise calculations can be used to make sure). The Gaussian mechanism works best when L_1 sensitivity grows much faster than L_2 sensitivity, which often happens when adding noise to many statistics at once (e.g., when answering a workload of hundreds or thousands of prespecified queries), and when a single unit of privacy can contribute to many of these statistics.

When using the basic Laplace or Gaussian mechanism, the noise is determined by the query's sensitivity, which is independent of the size of the group being counted. The same amount of noise is added whether the count is 20 or 20 million. This means that the absolute error one can expect is constant. However, the relative error is smallest when the size of the group being counted (i.e., the signal) is large. As group size gets smaller, the strength of the signal goes down while the noise remains the same, resulting in higher relative error.

In a bar chart, the group size associated with each "bin" (i.e., the signal) tends to go down as the number of groups goes up. Thus, finer-grained differentially private bar charts that break down results across more categories tend to result in higher relative error than coarser-grained bar charts.

Key Takeaway: To minimize relative error in differentially private statistical analyses, minimize the number of data groups.

3.4.2. Summation Queries

A *summation query* calculates the sum of specific values. For example, a summation-query could return the sum of the transaction amounts for all pumpkin spice latte purchases in a year.

Example: Summation query. What is the total amount spent on pumpkin spice lattes since 2010?

For a summation query, the amount of noise needed to achieve differential privacy depends on the maximum value of the things being summed up. As a result, the mechanism must enforce that each summand is bounded by fixed upper and lower bounds—a process called *clipping* or *clamping*. These bounds—called the *clipping parameter*—must either be specified by the analyst before processing the data, or automatically determined using some of the privacy budget [48]. For large datasets, it is often possible to achieve good utility with differentially private summation queries. See Appendix Sec. B.4 for technical details.

Key Takeaway: Differentially private summation queries require fixed upper and lower bounds on data elements, which must be given without looking at the data, or automatically determined from the data using a differentially private mechanism. The bounds should generally be as small as possible to reduce noise while ensuring that only extreme outliers fall outside of the bounds.

Utility

Utility for summation queries is typically measured using the same metrics as counting queries. In addition, the *clipping parameter* can introduce bias in the results by reducing large values while preserving small ones. Utility analysis of summation queries should measure and consider this bias.

The clipping parameters (i.e., the upper and lower limits) are extremely important for accuracy. If the upper limit is too high, it will add unnecessary noise. If it is too low, then information that was present in the data will be lost by modifying too many of the data points (i.e., introducing bias). Some methods for determining clipping parameters require inspecting the data. Such methods could potentially leak sensitive information, and so should either be avoided or performed with careful analysis of the privacy risks involved.

3.4.3. Average Queries

An *average query* determines the mean of a set of values.

Example: Average Query. What is the average amount spent on pumpkin spice lattes since 2010?

An average query can be decomposed into a summation query and a counting query, and it can be answered with differential privacy via such a decomposition (see Appendix Sec. B.5 for technical details). Differentially private averages can yield high utility for large datasets.

Utility

The same metrics are used to evaluate average queries as summation queries. Because this process incorporates a summation query, it has the potential to introduce bias into the results. Like summation and counting queries, the best relative error will be achieved when group sizes are large and the *clipping parameter* is set appropriately.

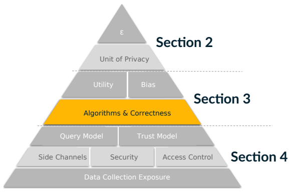
3.4.4. Min/Max Queries

Two other aggregation functions commonly available in database engines and used in statistical analysis are the minimum (min) and maximum (max). These are not commonly used in differentially private analyses because they have unbounded sensitivity. These aggregation functions do not really aggregate multiple values from the data. Rather, they return a single data element that represents the max or min, potentially degrading the privacy of the individual corresponding to that value.

When an estimate of dataset scale (i.e., the size and shape of the data) is needed, differentially private quantile estimation [49, 50] can be used instead of the min and max functions.

3.5. Machine Learning

Machine learning techniques are often used to understand data, and deep learning techniques have become especially popular because of their capabilities in complex domains like vision and language.



Common machine learning techniques, including the neural networks used in deep learning, start with a model that has trainable parameters. The model can be used to perform a task (e.g., recognizing pictures of pumpkin spice lattes), and the parameters control how the model operates. The training process is designed to set the model parameters so as to maximize the model's ability to perform its task on the training data. For example, a training dataset might contain some pictures of pumpkin spice lattes and some pictures of other objects. The goal in training would be to set the parameters so that the model correctly identifies all of the pictures of pumpkin spice lattes, and does not identify the other objects as pumpkin spice lattes.

Privacy Risks in Machine Learning

In the past few years, strong privacy attacks against trained models have sometimes allowed an attacker to learn information about the training data used to train the model. This can raise serious concerns for models trained on sensitive data (e.g., medical diagnosis models trained on x-ray data or language models trained on private emails).

Privacy Hazard: Machine learning techniques do not automatically protect privacy. Neural networks are particularly susceptible to memorizing training data.

Deep neural networks are particularly susceptible to these kinds of attacks. Recent work has shown that deep neural networks often memorize their training data [51], and techniques like membership inference attacks [52] can leverage this kind of memorization to detect whether or not a particular data element was used to train the model. Other kinds of attacks have been used to directly extract training data from image recognition models [53], image generation models [54], and large language models [55, 56].

Achieving Differential Privacy

To defend against privacy attacks in machine learning, a significant amount of research has explored how to train differentially private models [57–60]. The most commonly used technique is called differentially-private stochastic gradient descent (DP-SGD) [58] (see Appendix Sec. B.6 for technical details).

Differentially private implementations are also available for boosted decision trees, k-means clustering, graph analysis, item and set extraction, model alignment, few-shot learning, and similar common machine learning and data processing methods [61–70]. In general, a differentially private implementation of a common data processing algorithm will yield favorable privacy-utility tradeoffs compared to generating a differentially private synthetic dataset first, followed by applying the standard (non-private) version of the algorithm.

Utility

Adding differential privacy to the training process using current techniques typically lowers accuracy, sometimes significantly [71].

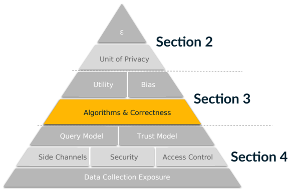
In general, two major factors influence the accuracy of differentially private machine learning. First, simple models are much easier to train with privacy from scratch than are complex models. Complex models, like deep neural networks, can have millions or billions of trainable parameters and are more likely to be affected by the noise added for differential privacy. Large models that have been pre-trained on public data can show strong privacy-utility tradeoffs when fine-tuned or trained in a continued pretraining regime using DP-SGD, and this privacy-utility tradeoff appears to improve with model size [72–74]. However, not all publicly available information is necessarily non-sensitive, and one should take care to ensure that any publicly available information used to pre-train large models is indeed free of sensitive information. Simpler models, like linear models, can be much easier to train with differential privacy. Second, larger training datasets generally lead to more accurate models. As in the analytics queries discussed earlier, aggregating over larger groups generally leads to better accuracy, and aggregating over smaller groups implies worse accuracy. With enough training data, differentially private approaches to machine learning can approximately match the accuracy of non-private training [59], but a large amount of data are often required.

One effective approach for training differentially private neural networks is to perform *pre-training* on publicly-available data using standard non-privacy-preserving training algorithms, followed by *fine-tuning* using a differentially private training algorithm. For example, a large language model might be pre-trained on public-domain text, then fine-tuned on sensitive electronic health records. The pre-training step initializes the model with general information about the English language, and the fine-tuning step adds information about the target domain. The pre-training step reduces the amount of information the model needs to learn from the sensitive data, and thus reduces the negative impact on accuracy of the noise used to achieve differential privacy. This approach has been shown to be effective for image models [59] and language models [75].

Key Takeaway: Current techniques for differentially private machine learning work best for simple models and very large training datasets. When public data is available, *pre-training* using public data can improve accuracy.

3.6. Synthetic Data

A *differentially private synthetic dataset* is a synthetic dataset built with differential privacy. A *synthetic dataset* looks like the original dataset in that it has the same schema and attempts to maintain the properties of the original dataset (e.g., correlations between attributes). However, it consists of algorithmically generated data associated with “fake” individuals. Because it looks like the original data, synthetic data are particularly easy to use. It can be analyzed using existing tools and workflows without modification. This section summarizes privacy considerations for synthetic data, and describes some approaches for constructing it.



For the purposes of this document, we focus on synthetic data that was generated from potentially identifiable data.

Privacy Considerations for Synthetic Data

Many techniques have been proposed for constructing synthetic data. Some are differentially private, while others are not. Nearly all of these techniques claim to provide some privacy benefits.

Synthetic data techniques that do not satisfy differential privacy generally provide only informal privacy guarantees. They may appear to protect the privacy of individuals, but like the de-identification techniques discussed earlier, they do not provide robust protection against all privacy attacks. Recall that differential privacy is resistant to all privacy attacks, even attacks not yet invented. Non-differentially private protections do not have the same resistance to future attacks. There are many reports of privacy attacks against non-differentially private synthetic data that have successfully revealed the original data [19].

Differentially private synthetic data can be used to prevent these attacks. This section summarizes some techniques for generating synthetic data while satisfying differential privacy. Techniques that do not specifically satisfy differential privacy may not necessarily provide robust privacy protection.

Privacy Hazard: Synthetic data generated without differential privacy may be susceptible to privacy attacks.

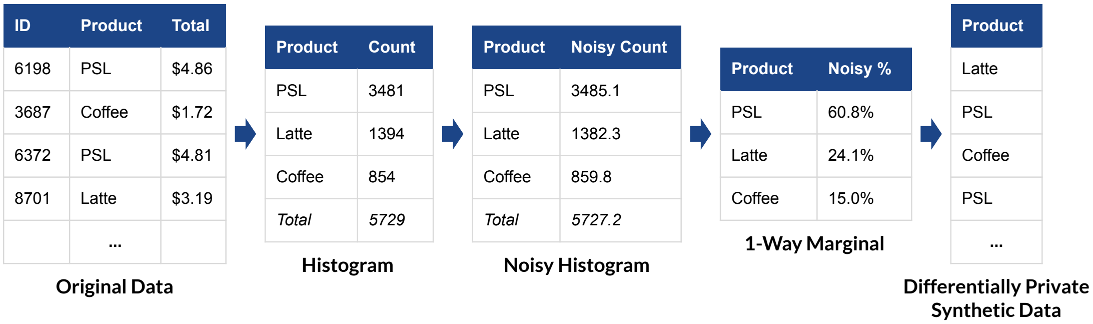
Key Takeaway: To provide robust privacy protection, including against novel developments in privacy attacks, synthetic data should be generated using differentially private algorithms.

Utility Considerations of Synthetic Data

While synthetic data are convenient for downstream data users, it can also introduce utility challenges that are difficult to mitigate. In some cases, synthetic data can reduce accuracy for sub-populations, leading to systemic bias [76]. Synthetic data can also complicate understanding of the accuracy of statistics in the data. The synthetic data generation process adds additional sources of uncertainty to the statistical uncertainty in the original data. This error has the potential to propagate to downstream data uses. Similarly, bias introduced by the generative algorithms can also propagate error to downstream users. These utility challenges apply to all synthetic data, whether differentially private or not.

Generating Synthetic Data

Conceptually, all techniques for generating synthetic data—privacy-preserving or not—start by building a probabilistic model of the underlying population from which the original data was sampled. This model is then used to generate new data. If the model is an accurate representation of the population, then the newly generated data will retain all of the properties of that population, but each generated data point will represent a “fake” indi-



ID	Product	Total
6198	PSL	$4.86
3687	Coffee	$1.72
6372	PSL	$4.81
8701	Latte	$3.19
	...	

Original Data

Product	Count
PSL	3481
Latte	1394
Coffee	854
Total	*5729*

Histogram

Product	Noisy Count
PSL	3485.1
Latte	1382.3
Coffee	859.8
Total	*5727.2*

Noisy Histogram

Product	Noisy %
PSL	60.8%
Latte	24.1%
Coffee	15.0%

1-Way Marginal

Product
Latte
PSL
Coffee
PSL
...

Differentially Private
Synthetic Data

Fig. 18. Generating a differentially private synthetic data using a marginal distribution. (PSL = Pumpkin Spice Latte)

vidual who does not actually exist. Building the model is the most challenging part of this process. Many techniques have been developed for this purpose, from simple approaches based on counting to complex ones based on deep learning.

Differentially Private Synthetic Data via Private Marginals

Imagine that one would like to generate synthetic sales data for a pumpkin spice latte company. One way to accomplish this would be to use a differentially private marginal distribution, as in Fig. 18. A bar chart could be constructed from the original tabular data by counting the number of each drink sold. Next, noise would be added to the bar chart to satisfy differential privacy. Finally, each noisy count would be divided by the total to determine what percentage of all drinks were of a specific type. This final step would produce a one-way marginal distribution since it would consider only one attribute of the original data and ignore correlations between attributes. The one-way marginal distribution could then be used to generate a "fake purchase" using weighted randomness. A drink type would be randomly chosen with the randomness weighted according to the one-way marginal distribution that has been generated. In the example in Fig. 18, 60.8% of the generated purchases should be pumpkin spice lattes, 24.1% should be lattes, and 15.0% should be regular coffees.

Marginal distributions form the basis for many differentially private synthetic data algorithms. The major challenge of this approach is preserving correlations between data attributes. For example, sales data might include the customer's age in addition to their preferred drink type, and age might be highly correlated with drink type (e.g., younger customers may be more likely to purchase pumpkin spice lattes than other drink types). The process used above can be repeated on both data attributes separately, but that approach does not capture the correlation that was present between the two.

This correlation can be preserved by calculating a two-way marginal—a distribution over both data attributes simultaneously. However, this marginal has many more possible options (all of the possible combinations of age and drink type), and it will result in a weaker "signal" relative to the noise for each option. Preserving correlations like these requires a careful balance between the marginals being measured and the strength of the signal being preserved.

Differentially Private Synthetic Data via Deep Learning

Another way to build a model of the underlying population from the original data is with machine learning techniques. In the past several years, deep learning-based methods for generating synthetic data have become more capable in some domains [60]. Approaches like generative adversarial networks (GANs)—a particular type of neural network—are particularly good at generating convincing photos of imaginary people. The same approach can be used to generate synthetic data in other domains (e.g., latte sales data) by training the neural network on original data from the right domain.

Generative models have been used extensively to produce non-private synthetic data. As described earlier, these techniques do not necessarily provide robust privacy protection for individuals in the original dataset, and the resulting synthetic data may be susceptible to privacy attacks. If robust privacy protection is desired, a differentially private training algorithm like DP-SGD must be used to train the generative model.

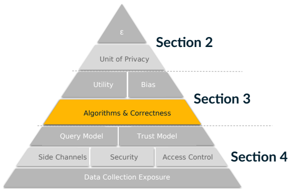
To achieve differential privacy, the neural network can be trained using a differentially private algorithm, like the DP-SGD algorithm described earlier. If the neural network modeling the underlying population is trained with differential privacy, then by the *post-processing invariance* property, the synthetic data it generates also satisfies differential privacy.

Privacy Hazard: Current deep learning-based approaches for differentially private synthetic data produce significantly lower quality data than approaches based on marginals.

Unfortunately, deep learning-based approaches for differentially private synthetic data are currently much less useful than the marginal-based approaches for low-dimensional tabular data (e.g., the data in the latte example). In fact, deep learning-based approaches often fail to preserve even basic statistical properties of the original data. This difference is likely due to the model complexity challenges described earlier since generative models tend to be especially complex.

3.7. Unstructured Data

Unstructured data often refers to text, pictures, audio, and video—formats that often lack structure that relates data to individuals. This lack of structure sometimes makes it difficult to think about privacy.



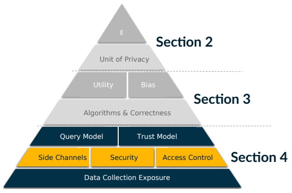
This lack of structure makes it difficult to define a meaningful unit of privacy. Relying on ownership or authorship ignores the additional people who may appear in unstructured data (e.g., a video that contains several people). Adding or removing all of a single owner's data may not remove all presence of that person in the dataset, and may remove part of the presence of several other people.

Due to these challenges, research in differential privacy has not focused on unstructured data. Existing techniques generally require specifying a unit of privacy (e.g., one minute/hour of video) that may represent a compromise in privacy.

If a suitable unit of privacy can be determined, then it is often possible to compute differentially private statistics and train machine learning models on unstructured data. In machine learning, there has been significant work on image recognition [58, 59, 77], natural language processing [75, 78], and obfuscating the author of a text [79]. Differential privacy has also been applied to video [80] and to mask patterns of communication (including metadata) in anonymous communication systems [81].

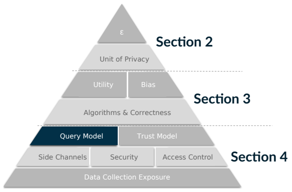
Privacy Hazard: For unstructured data, defining the unit of privacy can be difficult or impossible because it is often unclear who (in addition to the owner) appears in a piece of data. As a result, defining meaningful differential privacy guarantees for unstructured data is challenging.

4. Deploying Differential Privacy



This section describes practical concerns in deploying differentially private analysis techniques. Chief among these is the trust model (Sec. 4.2), which describes who can be considered trustworthy and who should be considered malicious. This section also discusses several implementation challenges for differentially private mechanisms that can cause unexpected privacy failures (Sec. 4.3). The final subsections describe security concerns (Sec. 4.4) and data collection exposure (Sec. 4.5).

4.1. Query Models



The deployment of differential privacy is separated into two common models: the *data release* model and the *interactive query answering* model. The data release model is simpler and more trustworthy but is limited. The interactive query answering model is more flexible but more complex to deploy and, thus, more vulnerable to security bugs in its implementation. Furthermore, risk from lack of side channel protections may be compounded due to the choice of query model. The essence of what makes a query model and side channel mitigation approach low or high risk is distilled in a flowchart, shown in Fig. 19.

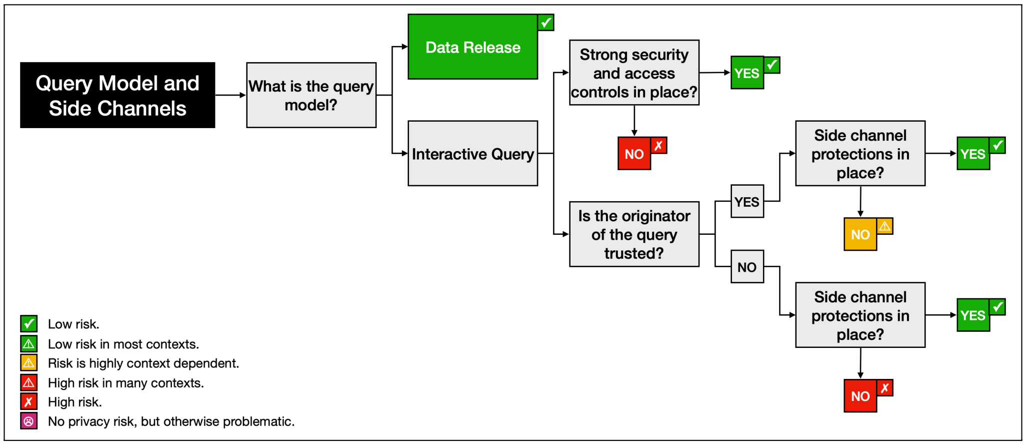

Query Model and Side Channels
What is the query model?
Data Release ✓
Interactive Query
Strong security and access controls in place?
YES ✓
NO ✗
Is the originator of the query trusted?
YES
NO
Side channel protections in place?
YES ✓
NO ⚠
Side channel protections in place?
YES ✓
NO ✗
✓ Low risk.
⚠ Low risk in most contexts.
⚠ Risk is highly context dependent.
⚠ High risk in many contexts.
✗ High risk.
☹ No privacy risk, but otherwise problematic.

Fig. 19. A flowchart for determining whether or not a query model and side channel mitigation approach is low or high risk in combination.

In the *data release* model, the queries are known in advance and are often specified by the same organization collecting the data. The organization can collect the data, use differentially private mechanisms to answer the queries, and release the results all in one step. In the data release model, the predetermined queries generally attempt to describe the population from which the data was collected. For example, they may generate bar charts (§3.4) or synthetic data (§3.6). The U.S. decennial census is one example of the data release model: the queries are prespecified by the U.S. Census Bureau and designed to describe the U.S. population. The data release model is simpler than the alternatives, but it requires all queries to be specified in advance and does not allow new queries to be asked after the release.

In the *interactive query answering* model, the queries are not known in advance, and analysts interact with a system designed to answer queries on an ongoing basis. Queries may be specified in large batches (i.e., a *workload*) or individually, and analysts may or may not be members of the same organization that collected the data. The query answering model empowers analysts to specify their own custom queries at any time, which is a significant advantage over the data release model for some applications. However, compared to the data release model, the query answering model raises significant additional challenges in the areas of privacy budgeting and security.

Privacy Hazard: Compared to the data release model, the interactive query answering model raises significant additional challenges related to privacy budgeting and security.

Privacy Budgeting

In the data release model, the entire privacy budget can be allocated among the predetermined queries, and the result is intended to adequately describe the important properties of the original population. By the *post-processing invariance* property of differential privacy, the results can be used by anyone as many times as desired without incurring additional privacy loss.

In the interactive query answering model, each unique query answered by the system incurs additional privacy loss and must count against the total *privacy budget*. In this context, budgeting requires forecasting how many queries the system will need to answer. If the budget runs out, then the system must either refuse to answer new queries—an outcome that may be extremely problematic — or allow for an increased total privacy budget, which will incur additional privacy risk.

System Security and Malicious Analysts

In the data release model, the original data can be discarded or archived in a high-security environment after the differentially private results are calculated and released. This approach provides strong protection against the accidental release of the original sensitive data (e.g., due to data breaches). The differentially private results can then be computed by a trusted party within the same organization that collects the data. In this context, it is reasonable to assume that the party computing the results will make an honest attempt to correctly implement differential privacy and will not intentionally issue queries that target individuals.

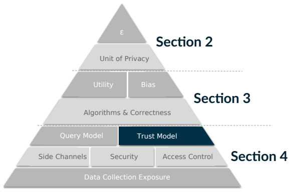
In the interactive query answering model, the original sensitive data must be kept available for querying on an ongoing basis. The system that accesses the data must therefore be highly secure in order to avoid data breaches that expose this data. Ensuring this kind of security adds significant complexity to a query answering deployment compared to a data release. Analysts may not be trustworthy and may intentionally try to violate the privacy guarantee, especially if the query answering system is exposed to the public or to analysts outside of an organization. Query answering systems are complex, and implementing them correctly is challenging and costly. Even carefully designed systems are likely to have bugs that cause security vulnerabilities (see Sec. 4.3 for details). Malicious insiders may attempt to find and exploit these bugs to break the privacy guarantee and reveal the original sensitive data.

Utility and Data Trustworthiness

In the data release model, utility can be evaluated in a confidential manner by the trusted party before the results are released. This trusted party can make sure that the results are fit-for-use, and provide data analysts with guidance on what kind of analyses are going to produce accurate outcomes using differentially private data.

In the interactive query answering model, accuracy information cannot be entirely conveyed and evaluated to untrusted data analysts: some details, like noise scales used for underlying mechanisms, can be communicated, but others, like the impact of clipping thresholds on the utility of returned results, cannot easily be returned in a way that enforces the desired differential privacy guarantee. This makes it difficult for data analysts to trust that the returned data are fit for use, and to quantify and account for the various kinds of inaccuracies that may have been introduced by the underlying differential privacy mechanisms.

4.2. Trust Models



A *trust model* describes assumptions about how trustworthy the components of a system are expected to be. In the setting of differential privacy, there is typically an assumption that final results will be released to the public. Since some members of the public may not be trustworthy, such results should be protected with a guarantee like differential privacy. However, the final results might not be revealed to the public and instead revealed only to a smaller group of people. This section describes several different trust models that are commonly used for deployments of differential privacy in terms of which participants in the system are trusted and which are untrusted. The essence of what makes a trust model low or high risk is distilled in a flowchart, shown in Fig. 20.

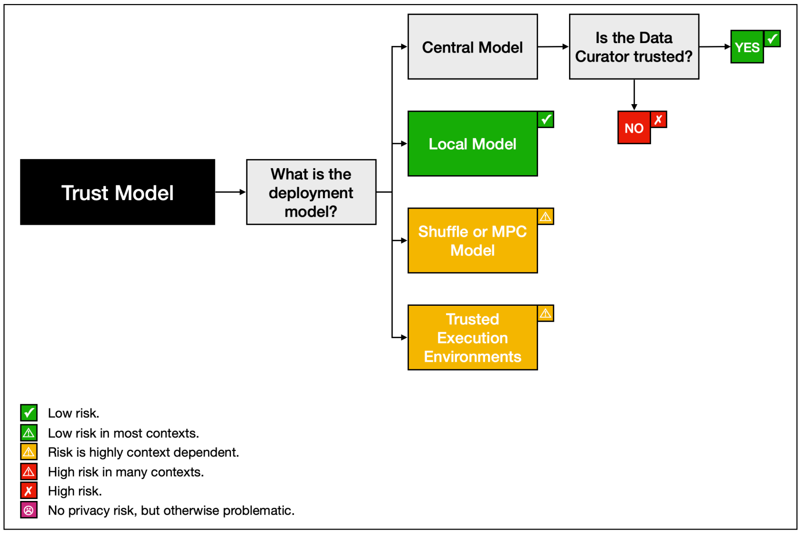

Trust Model
What is the deployment model?
Central Model
Is the Data Curator trusted?
YES ✓
NO ✗
Local Model ✓
Shuffle or MPC Model ⚠
Trusted Execution Environments ⚠
✓ Low risk.
⚠ Low risk in most contexts.
⚠ Risk is highly context dependent.
⚠ High risk in many contexts.
✗ High risk.
☹ No privacy risk, but otherwise problematic.

Fig. 20. An example flowchart for evaluating the appropriateness of a trust model.

Definition: Trust Assumption. A *trust assumption* about a party describes how that party is expected to behave when they are given access to sensitive data.

- A *trusted party* will keep sensitive data safe and will not reveal it to others. It is assumed that no privacy harms will result from sharing sensitive data with trusted parties.
- An *untrusted party* may not keep sensitive data safe and may reveal it to others. Privacy harms may result from sharing sensitive data with untrusted parties.

Most trust models for differential privacy are described in terms of the trust assumptions made about the following three parties:

1. The *data subjects*: who the data are about
2. The *data curator*: who aggregates the data
3. The *data consumer(s)*: who receive differentially private results

In many cases, the set of data consumers is very large. For example, when differentially private results are released to the public, everyone is a member of the set of data consumers. In other cases, differentially private results are only released to certain people.

Table 2 summarizes the trust assumptions made in some commonly used trust models for differential privacy. All of the models assume that the data subjects are trusted because differentially private systems are designed to protect the data subjects from the other parties, and there is no incentive for data subjects to cause privacy harms to themselves. The models differ in the trust assumptions for the other parties.

In general, trust models that require fewer trusted parties are stronger, but stronger trust

Table 2. Common deployment models for differential privacy and their trust assumptions.

Model	Data Subjects	Data Curator	Data Consumer	Details
Central Model	Trusted	Trusted	Untrusted	§ 4.2.1
Local Model	Trusted	Untrusted	Untrusted	§ 4.2.2
Shuffle Model	Trusted	Untrusted*	Untrusted	§ 4.2.3
Secure Computation	Trusted	Untrusted*	Untrusted	§ 4.2.3

* indicates additional system-dependent security assumptions.

models often trade other desirable features in exchange for lower trust requirements. The rest of this section describes these trade-offs in detail.

When evaluating a differential privacy guarantee, the most important consideration is whether the trust model's trust assumptions match reality. For example, in the central model of differential privacy (described in Sec. 4.2.1), the curator must be trusted. If the central model is used with an untrustworthy curator, then the differential privacy guarantee breaks down because the curator may simply release the sensitive data to another party. The choice of trust model is therefore directly constrained by realistic assumptions about the trustworthiness of the parties involved.

Privacy Hazard: The trust assumptions made by a differential privacy guarantee's trust model must hold in the real world. A failure of any of the trust assumptions makes the corresponding differential privacy guarantee meaningless.

Trust in the real world is complicated, and it can be difficult or impossible to relate real-world ideas about the trustworthiness of a party to a precise trust assumption in a trust model. For example, a differential privacy guarantee that requires an assumption of trust in the curator (e.g., central differential privacy) may be better than no guarantee at all, even when the data subject may not completely trust the curator in all respects.

4.2.1. Central Model

The most commonly used trust model in differential privacy research is called the central model of differential privacy (or simply, "central differential privacy"). This trust model is summarized in Fig. 21.

The key component of the central model is a trusted data curator. Each individual submits their sensitive data to the data curator, who stores all of the data in a central location (i.e., on a single server). The data curator is trusted in that users assume that they will not look at the sensitive data directly, will not share it with anyone, and cannot be compromised by any other adversary. In other words, with this model, there is an assumption that the server holding the sensitive data cannot be hacked.

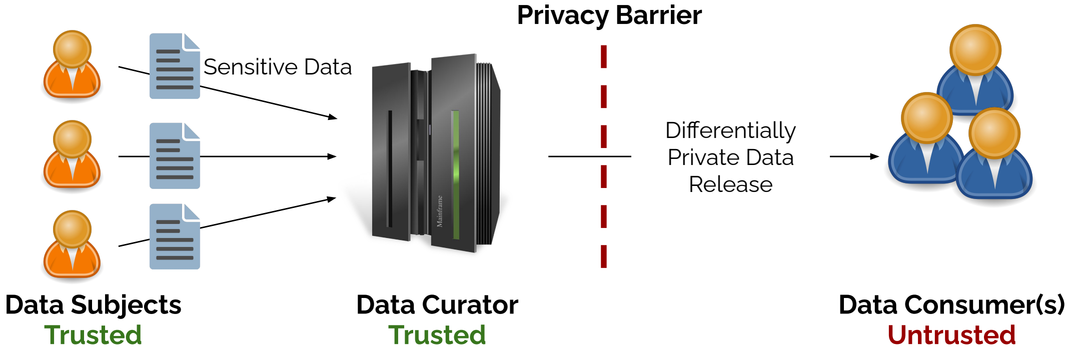


Fig. 21. Central model of differential privacy

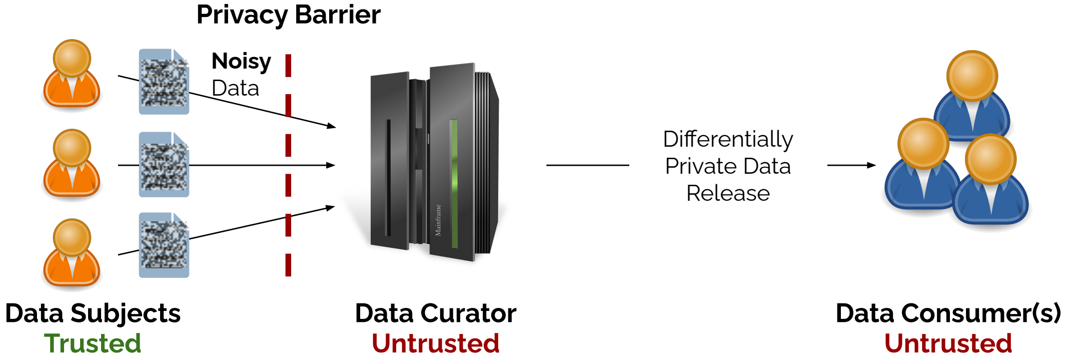


Fig. 22. Local model of differential privacy

In the central model, noise is typically added to results, as in the analyses described in Sec. 3. The advantage of this model is that it allows algorithms to add the smallest possible amount of noise and therefore produce results with the maximum accuracy allowed under differential privacy. The figure below demonstrates this process. The privacy barrier is placed between the trusted data curator and the data consumer. To the right of the privacy barrier, only differentially private results can be viewed, so the data consumer does not need to be trusted.

The disadvantage of the central model is that it requires a trusted data curator, and many data curators are not considered trustworthy. In fact, a lack of trust in the data collector is often a primary motivation for the use of differential privacy.

4.2.2. Local Model

The local model of differential privacy addresses the security issue in the central model by eliminating the trusted data curator. Each individual adds noise to their own data before sending it to the data curator. This means that the data curator never sees the sensitive data and does not need to be trusted. Fig. 22 demonstrates the local model, where the privacy barrier stands between the data subjects and the (untrusted) data curator. Even if the data curator's server is hacked, the hackers only see noisy data that already satisfy differential privacy. This is why the local model was adopted for Google's RAPPOR system [82] and Apple's data collection system [83].

However, the local model produces less accurate answers than the central model. In the local model, each individual adds enough noise to satisfy differential privacy. Thus, the total noise for all participants is much larger than the single noise sample used in the central model. As a result, the local model is only useful for queries with a very strong "signal." Apple's system, for example, uses the local model to estimate the popularity of emojis, but the results are only useful for the most popular emojis (i.e., where the "signal" is strongest). The local model is typically not used for more complex applications like machine learning.

4.2.3. Future Directions: Shuffle and Secure Computation Models

The central and local models of differential privacy offer a stark trade-off between trust assumptions and accuracy. A significant amount of recent research has investigated new ways to achieve the higher accuracy of the central model under the stronger trust assumptions of the local model. This section summarizes two approaches that are still in the early stages of development and have not yet been used in large-scale deployments.

One approach is the shuffling model, first implemented in Prochlo [84]. The shuffling model includes an untrusted data curator, individual data contributors, and a set of partially trusted shufflers. In this model, each individual adds a small amount of noise to their own data and submits that data to the shuffler, which adds additional noise before forwarding batches of data to the data curator. It is assumed that shufflers are unlikely to collude with the data curator or each other, so the small amount of noise added by individuals is sufficient to guarantee privacy. Each shuffler operates on a batch of inputs (same as the central model), so a small amount of additional noise guarantees privacy for the whole batch. The shuffling model is a compromise between the local and central models—it adds less noise than the local model but requires more noise than the central model.

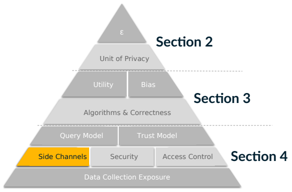
Another approach is to combine differential privacy with techniques from cryptography, such as secure multi-party computation (MPC) or fully homomorphic encryption (FHE). FHE allows for computing on encrypted data without decrypting it first, and MPC allows a group of parties to securely compute functions over distributed inputs without revealing the inputs. Computing a differentially private function using secure computation is a promising way to achieve the accuracy of the central model with the security benefits of the local model. In this approach, the use of secure computation eliminates the need for a trusted data curator. Recent work [85–87] demonstrates the promise of combining MPC and differential privacy to achieve most of the benefits of both the central and local models. In most cases, secure computation is several orders of magnitude slower than native execution, which is often impractical for large datasets or complex queries. However, secure computation is an active area of research, and its performance is improving quickly.

Secure hardware enclaves (also known as trusted execution environments) are special security-enabled CPUs that can provide security for data during computation by decrypting data only within the CPU itself, such as Intel's Software Guard Extensions (SGX), AMD's Secure Encrypted Virtualization (SEV), and ARM's TrustZone. Such platforms promise similar capabilities to the cryptographic techniques described above but with significantly enhanced performance. However, these platforms are still under development, and several existing hardware enclaves have been vulnerable to attacks that can extract sensitive data.

Secure multi-party computation, homomorphic encryption, and trusted execution environments are often cited as examples of privacy-enhancing technologies, since they are capable of hiding data during execution. Alone, none of these techniques can replace differential privacy; however, combining one or more of these techniques with differential privacy can enable new applications of differential privacy.

4.3. Mechanism Implementation Challenges

The private mechanisms introduced in the preceding sections were described using analytical equations, but in order to use them, they have to be implemented on computers. This section gives an overview of the subtle differences between the math and the implementation that can cause unexpected failures in privacy. Because of these challenges, it is best to use existing well-tested libraries whenever possible. In many cases, the developers of these libraries have worked to understand the potential implementation-based sources of privacy failure and have addressed the ones identified. Not every library provides solutions for all of these challenges; Appendix B.7 provides guidance for evaluating specific software libraries. Furthermore, risk from lack of side channel protections may be compounded due to the choice of query model. The essence of what makes a query model and side channel mitigation approach low or high risk is distilled in a flowchart, shown in Fig. 19.



Floating-Point Arithmetic

Previous sections have described the Laplace and Gaussian mechanisms in terms of infinite-precision real numbers. On computers, floating-point numbers are typically used instead. Since the floating-point representation has finite precision, there are some real numbers that simply cannot be represented using floating-point numbers. When noise is added to a sensitive floating-point number, the set of representable noisy values depends on the sensitive value. This means an adversary may be able to infer something about the sensitive value from the noisy value, by leveraging knowledge about what floating-point outputs are possible [88].

Privacy Hazard: Implementing differential privacy mechanisms is tricky and requires considering side-channel vulnerabilities.

The impact of floating-point imprecision on differential privacy implementations has been known for more than a decade [88], and techniques for addressing the associated challenges have been developed and implemented in most (but not all) libraries designed for practical use. The basic mechanisms in these libraries will generally be safer to use than custom-built implementations that do not take floating-point imprecision into account.

Timing Channels

In some cases, the time it takes to run a query may reveal something about the underlying data. This risk is especially pronounced if untrusted analysts are allowed to write their own queries and measure how long it takes to receive the answer. For example, it might be possible to write a program whose running time reveals whether or not Joe is a part of the data with 100% certainty:

Example: Timing Channel Attack.

```
if Joe in Data:
  return slowQuery()
else:
  return fastQuery()
```

In many settings, timing is not an issue because analysts are not allowed to design and submit their own queries, or they are not able to observe how long those queries take to run. If analysts can submit their own queries and measure running time, careful implementations must be used to hide the information revealed by the running time.

Backend Issues

In actual deployments where datasets may contain millions or billions of rows, it makes sense to reuse existing infrastructures to store and query data. Therefore, many systems for differentially private analysis leverage existing databases or distributed data processing solutions that were not originally designed for differentially private analysis.

This distinction can lead to the unexpected loss of privacy. For example, some database engines throw an error if a query attempts to divide by zero, so a malicious analyst might craft a query that divides by zero exactly when their target individual is part of the dataset. In this case, observing whether or not an error is thrown is a direct violation of privacy.

As in the case of timing channels, these concerns are less serious when analysts are not allowed to interact with the system directly. When analysts are allowed to craft their own queries and observe the results, it is important to ensure that the underlying systems that make up the differentially private query infrastructure do not contain additional channels that might leak private information, as in the example above.

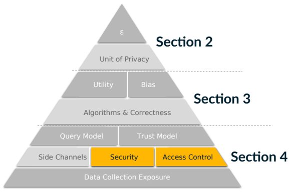
Tuning Hyperparameters

Many differential privacy mechanisms have tunable *hyperparameters* that must be set carefully to ensure utility, and the settings are often data-dependent. The *clipping parameter* used in summation queries and described in Sec. 3.4.2 is a good example—if the parameter is set too high, too much noise may be added, but if the parameter is set too low, clipping may introduce substantial bias. How should such hyperparameters be chosen?

A natural approach is to try different mechanisms and hyperparameters, measure the accuracy of the results obtained for each combination, and select the best-performing one. However, if this accuracy measurement is done using the private data, this step is not differentially private, and the choice of hyperparameters and mechanisms itself can leak information about the data [89].

One solution is to use non-sensitive data to perform the tuning. For example, tuning can be done with a historical dataset that was released previously, or with a synthetic dataset whose scale and distributional properties is expected to be similar to those of the actual sensitive dataset that will be used for deployment. Another solution is to use differentially private algorithms to perform the tuning [89].

4.4. Data Security and Access Control



The security of data plays an important role in the overall privacy guarantee, even though technologies for security perform a different function than differential privacy. These technologies control who can access the data, rather than what can be learned from the data. Many of the techniques described earlier require direct access to the original noise-free data. In the event of a data breach, the release of the original data makes the differential privacy guarantee meaningless. For this reason, data should be protected with strong security measures, both at rest (i.e., when it is being stored for later use) and during computation. Measures for protecting data at rest include encryption (combined with careful key management), access control, and strong system security. For more guidance on security and privacy controls relevant to reducing privacy risk in information systems, see NIST SP 800-53 Rev. 5 [90] and NIST SP 800-161 Rev. 1 [91].

Protecting data during computation is more challenging because computing on data typically requires decrypting it. This challenge has grown in recent years with the rise of cloud computing. As mentioned in Sec. 4.2, cryptographic techniques, hardware enclaves, and novel system architectures can help address this challenge, but all of these are active areas of research and have not been commonly deployed.

Privacy Hazard: Failures in data security can result in data breaches that make differential privacy guarantees meaningless.

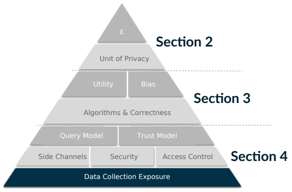
Access control policies describe who is allowed to access the data. For example, if the data are encrypted, an access control policy might say who has the keys.

For many security mechanisms, including encryption, data only remains secure if the individuals who have access to it are trustworthy. Some of the techniques discussed in Sec. 4.2 can help shift the trust requirements for a differentially private system.

Privacy Hazard: Failures in access control policy can result in data breaches that make differential privacy guarantees meaningless.

4.5. Data Collection Exposure

The majority of this publication has explored the technical features of a differential privacy guarantee with the assumption that users will know ahead of time what they want to learn and what sensitive data is needed in order to learn it. This is a strong assumption that is often untrue in practice.



The strongest possible approach to privacy is to not collect the data to begin with. Before evaluating if differential privacy is an appropriate framework to use in a data release, it is important to consider whether the data being analyzed needs to be collected at all. In some cases, it may be possible to collect less data and still achieve the desired final results.

By offering strong privacy protection for individuals, differential privacy might appear to eliminate the risks associated with collecting too much data. However, the use of differential privacy can reduce but not eliminate these risks, as demonstrated by the privacy hazards described throughout this document. The application of differential privacy is not an excuse to collect more data than necessary.

Privacy Hazard: Differential privacy does not eliminate the risks associated with collecting sensitive data. Organizations should minimize data collection, even when using differential privacy.

Surprisingly, collecting more information can sometimes enable stronger differential privacy guarantees. For example, an organization may avoid collecting user identifiers in order to reduce the risk associated with collecting this information, but a lack of user identifiers can make it impossible to bound user contributions to provide a user-level unit of privacy (as described in Sec. 2.4).

4.6. Conclusion

Differentially private algorithms are currently the best known method for providing robust privacy protection against known and future attacks, even in the face of multiple data releases. This publication has summarized just a few of the many kinds of data analyses that can be accomplished with differential privacy, and current research is expanding these capabilities every year. In addition, an increasing number of open-source libraries and systems are starting to bring these techniques into practice.

This publication has described important considerations for implementing differential privacy and key hazards in evaluating differential privacy guarantees. The privacy parameter ε and the unit of privacy are particularly important since differential privacy provides very little protection when these parameters are not set appropriately. The whole system implementing a differential privacy guarantee should also be carefully considered, including security measures used to protect sensitive data while it is being processed. Weak differential privacy guarantees risk becoming instances of privacy theater—measures that claim to protect privacy but actually fail to do so. This publication is intended to help practitioners tell the difference between stronger and weaker differential privacy guarantees and deploy differential privacy in ways that actually provide robust privacy protection.

This publication is also intended to be a first step toward building differential privacy guarantee standards that provide parameter settings and solutions for all of the privacy hazards described in this publication (e.g., the value of ε, the unit of privacy, etc.). For some hazards, a standard should describe specific measures that practitioners should take to ensure that their deployments are free of problems known to undermine the privacy guarantee or lead to other issues (e.g., mechanism implementations are bug-free, results do not magnify bias, data collection is minimized, and sensitive data are properly secured). Such a standard would allow for the construction of tools to evaluate differential privacy guarantees and the systems that provide them as well as certification that systems conform with the standard. The certification of differential privacy guarantees is particularly important given the challenge of communicating these guarantees to non-experts [92]. A thorough certification process would provide non-experts with an important signal that a particular system will provide robust guarantees without requiring them to understand the details of those guarantees.

We hope that the path to standardization will parallel the successful development of cryptography from theoretical ideas to practical implementations and then to robust standards. However, the path to standardization in differential privacy may be even more challenging than it was in cryptography. There are still parameters that are not yet fully understood (e.g., the impact of ε on real-world privacy), and differential privacy imposes an inherent trade-off between privacy and utility that can be hard to navigate. Moreover, managing this trade-off requires considering the often conflicting interests of multiple stakeholders. For example, data analysts may prioritize utility, while data subjects may prioritize privacy. These challenges have resulted in a complicated policy-making process for existing deployments of differential privacy [93]. Finally, emerging combinations of differential privacy with other privacy-enhancing technologies (as described in Sec. 4.2.3) will significantly expand the application space for differential privacy, and may introduce additional complexity. Sharing the lessons learned from an increasing number of use cases and deployments of differential privacy will provide greater insights on how to address these challenges as well as the others described in this publication, and pave the way towards standardization.

References

[1] Gong R, Groshen EL, Vadhan S (2022) Harnessing the Known Unknowns: Differential Privacy and the 2020 Census. *Harvard Data Science Review* (Special Issue 2). Https://hdsr.mitpress.mit.edu/pub/fgyf5cne.

[2] European Parliament, Council of the European Union Regulation (EU) 2016/679 of the European Parliament and of the Council. Available at https://data.europa.eu/eli/reg/2016/679/oj.

[3] Garfnkel S, Garfinkel S, Near J, Dajani A, Singer P, Guttman B (2023) *De-Identifying Government Datasets: Techniques and Governance* (US Department of Commerce, National Institute of Standards and Technology).

[4] Ohm P (2009) Broken promises of privacy: Responding to the surprising failure of anonymization. *UCLA Law Review, Vol 57, p 1701, 2010* .

[5] Sweeney L (1997) Weaving technology and policy together to maintain confidentiality. *The Journal of Law, Medicine & Ethics* 25(2-3):98–110.

[6] Garfinkel S, Near J, Dajani A, Singer P, Guttman B (2023) Security and privacy controls for federal information systems and organizations (De-Identifying Government Datasets: Techniques and Governance, Gaithersburg, MD), NIST Special Publication (SP) 800-188. https://doi.org/https://doi.org/10.6028/NIST.SP.800-188

[7] Dinur I, Nissim K (2003) Revealing information while preserving privacy. *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp 202–210.

[8] Abowd JM, Adams T, Ashmead R, Darais D, Dey S, Garfinkel SL, Goldschlag N, Kifer D, Leclerc P, Lew E, et al. (2023) The 2010 census confidentiality protections failed, here's how and why (National Bureau of Economic Research),

[9] (2020) Nist privacy framework. https://doi.org/10.6028/NIST.CSWP.01162020. Available at https://www.nist.gov/privacy-framework

[10] (2020) Nist privacy risk assessment methodology (pram). Available at https://www.nist.gov/privacy-framework/nist-pram.

[11] Dalenius T (1977) Towards a methodology for statistical disclosure control. *Statistik Tidskrift* .

[12] Dwork C, Naor M (2010) On the difficulties of disclosure prevention in statistical databases or the case for differential privacy. *Journal of Privacy and Confidentiality* 2(1). https://doi.org/10.29012/jpc.v2i1.585. Available at https://journalprivacyconfidentiality.org/index.php/jpc/article/view/585

[13] Abowd JM, Hawes MB (2022) Confidentiality protection in the 2020 us census of population and housing (arxiv), https://doi.org/10.48550/ARXIV.2209.03310

[14] Kifer D, Abowd JM, Ashmead R, Cumings-Menon R, Leclerc P, Machanavajjhala A, Sexton W, Zhuravlev P (2022) Bayesian and frequentist semantics for common variations of differential privacy: Applications to the 2020 census (arxiv), https://doi.org/10.48550/ARXIV.2209.03310

[15] Dwork C, McSherry F, Nissim K, Smith A (2006) Calibrating noise to sensitivity in pri-

vate data analysis. *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3* (Springer), pp 265–284.

[16] Wood A, Altman M, Bembenek A, Bun M, Gaboardi M, Honaker J, Nissim K, O'Brien DR, Steinke T, Vadhan S (2018) Differential privacy: A primer for a non-technical audience. *Vanderbilt Journal of Entertainment & Technology Law* 21(1):209.

[17] Desfontaines D (2021) A list of real-world uses of differential privacy, https://desfontain.es/privacy/real-world-differential-privacy.html. Ted is writing things (personal blog).

[18] Gadotti A, Houssiau F, Annamalai MSMS, de Montjoye YA (2022) Pool inference attacks on local differential privacy: Quantifying the privacy guarantees of apple's count mean sketch in practice. *31st USENIX Security Symposium (USENIX Security 22)*, pp 501–518.

[19] Stadler T, Oprisanu B, Troncoso C (2022) Synthetic data–anonymisation groundhog day. *31st USENIX Security Symposium (USENIX Security 22)*, pp 1451–1468.

[20] Nasr M, Song S, Thakurta A, Papernot N, Carlini N (2021) Adversary instantiation: Lower bounds for differentially private machine learning. *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021* (IEEE), pp 866–882. https://doi.org/10.1109/SP40001.2021.00069. Available at https://doi.org/10.1109/SP40001.2021.00069

[21] Pejó B, Desfontaines D (2022) *Guide to Differential Privacy Modifications: A Taxonomy of Variants and Extensions*. SpringerBriefs in Computer Science (Springer International Publishing). https://doi.org/10.1007/978-3-030-96398-9. Available at https://link.springer.com/10.1007/978-3-030-96398-9

[22] Dwork C, Roth A, et al. (2014) The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9(3–4):211–407.

[23] Balle B, Barthe G, Gaboardi M, Hsu J, Sato T (2020) Hypothesis testing interpretations and renyi differential privacy. *International Conference on Artificial Intelligence and Statistics* (PMLR), pp 2496–2506.

[24] Dong J, Roth A, Su WJ (2022) Gaussian differential privacy. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 84(1):3–37.

[25] Wasserman L, Zhou S (2010) A statistical framework for differential privacy. *Journal of the American Statistical Association* 105(489):375–389.

[26] Kifer D, Machanavajjhala A (2011) No free lunch in data privacy. *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pp 193–204.

[27] Seeman J, Sexton W, Pujol D, Machanavajjhala A (2023) Privately answering queries on skewed data via per record differential privacy. *arXiv preprint arXiv:231012827* .

[28] Desfontaines D, Pejó B (2019) SoK: differential privacies. *arXiv preprint arXiv:190601337* .

[29] Bichsel B, Gehr T, Drachsler-Cohen D, Tsankov P, Vechev M (2018) Dp-finder: Finding differential privacy violations by sampling and optimization. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp 508–524.

[30] Ding Z, Wang Y, Wang G, Zhang D, Kifer D (2018) Detecting violations of differential

privacy. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp 475–489.

[31] Wang Y, Ding Z, Kifer D, Zhang D (2020) Checkdp: An automated and integrated approach for proving differential privacy or finding precise counterexamples. *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp 919–938.

[32] Barthe G, Chadha R, Jagannath V, Sistla AP, Viswanathan M (2020) Deciding differential privacy for programs with finite inputs and outputs. *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, pp 141–154.

[33] Zhang H, Roth E, Haeberlen A, Pierce BC, Roth A (2020) Testing differential privacy with dual interpreters. *Proceedings of the ACM on Programming Languages* 4(OOPSLA):1–26.

[34] Kong W, Medina AM, Ribero M, Syed U (2024) Dp-auditorium: A large scale library for auditing differential privacy. *2024 IEEE Symposium on Security and Privacy (SP)* (IEEE Computer Society), pp 219–219.

[35] Nasr M, Hayes J, Steinke T, Balle B, Tramèr F, Jagielski M, Carlini N, Terzis A (2023) Tight auditing of differentially private machine learning. *32nd USENIX Security Symposium (USENIX Security 23)*, pp 1631–1648.

[36] Steinke T, Nasr M, Jagielski M (2024) Privacy auditing with one (1) training run. *Advances in Neural Information Processing Systems* 36.

[37] Jagielski M, Ullman J, Oprea A (2020) Auditing differentially private machine learning: How private is private sgd? *Advances in Neural Information Processing Systems* 33:22205–22216.

[38] Aerni M, Zhang J, Tramèr F (2024) Evaluations of machine learning privacy defenses are misleading. *arXiv preprint arXiv:240417399* .

[39] Bowen CM, Snoke J (2021) Comparative study of differentially private synthetic data algorithms from the NIST PSCR differential privacy synthetic data challenge. *Journal of Privacy and Confidentiality* 11(1). Available at https://arxiv.org/abs/1911.12704.

[40] Schwartz R, Vassilev A, Greene K, Perine L, Burt A, Hall P, et al. (2022) Towards a standard for identifying and managing bias in artificial intelligence. *NIST Special Publication* 1270:1–77. Available at https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1270.pdf.

[41] Sen A, Task C, Kapur D, Howarth G, Bhagat K (2023) Diverse community data for benchmarking data privacy algorithms. *Advances in Neural Information Processing Systems*, eds Oh A, Naumann T, Globerson A, Saenko K, Hardt M, Levine S (Curran Associates, Inc.), Vol. 36, pp 51409–51420. Available at https://proceedings.neurips.cc/paper_files/paper/2023/file/a15032f8199511ced4d7a8e2bbb487a5-Paper-Datasets_and_Benchmarks.pdf.

[42] Task C, Bhagat K, Howarth G (2023) Sdnist v2: Deidentified data report tool. *National Institute of Standards and Technology* https://doi.org/10.18434/mds2-2943

[43] Mitchell M, Wu S, Zaldivar A, Barnes P, Vasserman L, Hutchinson B, Spitzer E, Raji ID, Gebru T (2019) Model cards for model reporting. *Proceedings of the conference on*

fairness, accountability, and transparency, pp 220–229.

[44] Buolamwini J, Gebru T (2018) Gender shades: Intersectional accuracy disparities in commercial gender classification. *Conference on fairness, accountability and transparency* (PMLR), pp 77–91.

[45] Raji ID, Gebru T, Mitchell M, Buolamwini J, Lee J, Denton E (2020) Saving face: Investigating the ethical concerns of facial recognition auditing. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp 145–151.

[46] Benjamin R (2020) Race after technology: Abolitionist tools for the new jim code.

[47] Delta Calculation for Thresholding, https://github.com/google/differential-privacy/blob/main/common_docs/Delta_For_Thresholding.pdf [accessed 11/8/2022].

[48] Wilson RJ, Zhang CY, Lam W, Desfontaines D, Simmons-Marengo D, Gipson B (2020) Differentially private sql with bounded user contribution. *Proceedings on Privacy Enhancing Technologies* .

[49] Gillenwater J, Joseph M, Kulesza A (2021) Differentially private quantiles. *International Conference on Machine Learning* (PMLR), pp 3713–3722.

[50] Kaplan H, Schnapp S, Stemmer U (2022) Differentially private approximate quantiles. *International Conference on Machine Learning* (PMLR), pp 10751–10761.

[51] Carlini N, Liu C, Erlingsson Ú, Kos J, Song D (2019) The secret sharer: Evaluating and testing unintended memorization in neural networks. *28th USENIX Security Symposium (USENIX Security 19)*, pp 267–284.

[52] Shokri R, Stronati M, Song C, Shmatikov V (2017) Membership inference attacks against machine learning models. *2017 IEEE symposium on security and privacy (SP)* (IEEE), pp 3–18.

[53] Haim N, Vardi G, Yehudai G, Shamir O, Irani M (2022) Reconstructing training data from trained neural networks. *Advances in Neural Information Processing Systems* 35:22911–22924.

[54] Carlini N, Hayes J, Nasr M, Jagielski M, Sehwag V, Tramer F, Balle B, Ippolito D, Wallace E (2023) Extracting training data from diffusion models. *32nd USENIX Security Symposium (USENIX Security 23)*, pp 5253–5270.

[55] Carlini N, Tramer F, Wallace E, Jagielski M, Herbert-Voss A, Lee K, Roberts A, Brown T, Song D, Erlingsson U, et al. (2021) Extracting training data from large language models. *30th USENIX Security Symposium (USENIX Security 21)*, pp 2633–2650.

[56] Nasr M, Carlini N, Hayase J, Jagielski M, Cooper AF, Ippolito D, Choquette-Choo CA, Wallace E, Tramèr F, Lee K (2023) Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:231117035* .

[57] Chaudhuri K, Monteleoni C, Sarwate AD (2011) Differentially private empirical risk minimization. *Journal of Machine Learning Research* 12(3).

[58] Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L (2016) Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp 308–318.

[59] De S, Berrada L, Hayes J, Smith SL, Balle B (2022) Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:220413650* .

[60] Jordon J, Yoon J, Van Der Schaar M (2018) Pate-gan: Generating synthetic data with differential privacy guarantees. *International conference on learning representations*.

[61] Gopi S, Gulhane P, Kulkarni J, Shen JH, Shokouhi M, Yekhanin S (2020) Differentially private set union. *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event* (PMLR), *Proceedings of Machine Learning Research*, Vol. 119, pp 3627–3636. Available at http://proceedings.mlr.press/v119/gopi20a.html.

[62] Kim K, Gopi S, Kulkarni J, Yekhanin S (2021) Differentially private n-gram extraction. *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, eds Ranzato M, Beygelzimer A, Dauphin YN, Liang P, Vaughan JW, pp 5102–5111. Available at https://proceedings.neurips.cc/paper/2021/hash/28ce9bc954876829eeb56ff46da8e1ab-Abstract.html.

[63] Su D, Cao J, Li N, Bertino E, Jin H (2016) Differentially private k-means clustering. *Proceedings of the Sixth ACM on Conference on Data and Application Security and Privacy, CODASPY 2016, New Orleans, LA, USA, March 9-11, 2016*, eds Bertino E, Sandhu RS, Pretschner A (ACM), pp 26–37. https://doi.org/10.1145/2857705.2857708. Available at https://doi.org/10.1145/2857705.2857708

[64] Li Q, Wu Z, Wen Z, He B (2020) Privacy-preserving gradient boosting decision trees. *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020* (AAAI Press), pp 784–791. https://doi.org/10.1609/AAAI.V34I01.5422. Available at https://doi.org/10.1609/aaai.v34i01.5422

[65] Maddock S, Cormode G, Wang T, Maple C, Jha S (2022) Federated boosted decision trees with differential privacy. *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, eds Yin H, Stavrou A, Cremers C, Shi E (ACM), pp 2249–2263. https://doi.org/10.1145/3548606.3560687

[66] Tang X, Shin R, Inan HA, Manoel A, Mireshghallah F, Lin Z, Gopi S, Kulkarni J, Sim R (2024) Privacy-preserving in-context learning with differentially private few-shot generation. *The Twelfth International Conference on Learning Representations*. Available at https://openreview.net/forum?id=oZtt0pRnOl.

[67] Wu F, Inan HA, Backurs A, Chandrasekaran V, Kulkarni J, Sim R (2024) Privately aligning language models with reinforcement learning. *The Twelfth International Conference on Learning Representations*. Available at https://openreview.net/forum?id=3d0OmYTNui.

[68] Imola J, Murakami T, Chaudhuri K (2021) Locally differentially private analysis of graph statistics. *30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021*, eds Bailey MD, Greenstadt R (USENIX Association), pp 983–1000. Available at https://www.usenix.org/conference/usenixsecurity21/presentation/imola.

[69] Zeng C, Naughton JF, Cai J (2012) On differentially private frequent itemset mining.

Proc VLDB Endow 6(1):25–36. https://doi.org/10.14778/2428536.2428539. Available at https://doi.org/10.14778/2428536.2428539

[70] Durfee D, Rogers RM (2019) Practical differentially private top-k selection with pay-what-you-get composition. *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, eds Wallach HM, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox EB, Garnett R, pp 3527–3537. Available at https://proceedings.neurips.cc/paper/2019/hash/b139e104214a08ae3f2ebcce149cdf6e-Abstract.html.

[71] Tramèr F, Boneh D (2021) Differentially private learning needs better features (or much more data). *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021* (OpenReview.net). Available at https://openreview.net/forum?id=YTWGvpFOQD-.

[72] Yu D, Naik S, Backurs A, Gopi S, Inan HA, Kamath G, Kulkarni J, Lee YT, Manoel A, Wutschitz L, Yekhanin S, Zhang H (2022) Differentially private fine-tuning of language models. *International Conference on Learning Representations*. Available at https://openreview.net/forum?id=Q42f0dfjECO.

[73] Li X, Liu D, Hashimoto TB, Inan HA, Kulkarni J, Lee YT, Thakurta AG (2022) When does differentially private learning not suffer in high dimensions? *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, eds Koyejo S, Mohamed S, Agarwal A, Belgrave D, Cho K, Oh A. Available at http://papers.nips.cc/paper_files/paper/2022/hash/b75ce884441c983f7357a312ffa02a3c-Abstract-Conference.html.

[74] He J, Li X, Yu D, Zhang H, Kulkarni J, Lee YT, Backurs A, Yu N, Bian J (2023) Exploring the limits of differentially private deep learning with group-wise clipping. *The Eleventh International Conference on Learning Representations*. Available at https://openreview.net/forum?id=oze0clVGPeX.

[75] Yu D, Naik S, Backurs A, Gopi S, Inan HA, Kamath G, Kulkarni J, Lee YT, Manoel A, Wutschitz L, et al. (2021) Differentially private fine-tuning of language models. *arXiv preprint arXiv:211006500* .

[76] Abowd J, Ashmead R, Cumings-Menon R, Garfinkel S, Kifer D, Leclerc P, Sexton W, Simpson A, Task C, Zhuravlev P (2021) An uncertainty principle is a price of privacy-preserving microdata. *Advances in neural information processing systems* 34:11883–11895.

[77] Papernot N, Abadi M, Erlingsson Ú, Goodfellow IJ, Talwar K (2017) Semi-supervised knowledge transfer for deep learning from private training data. *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (OpenReview.net). Available at https://openreview.net/forum?id=HkwoSDPgg.

[78] Anil R, Ghazi B, Gupta V, Kumar R, Manurangsi P (2021) Large-scale differentially private bert. *arXiv preprint arXiv:210801624* .

[79] Fernandes N, Dras M, McIver A (2019) Generalised differential privacy for text document processing. *International Conference on Principles of Security and Trust* (Springer, Cham), pp 123–148.

[80] Wang H, Xie S, Hong Y (2020) Videodp: A flexible platform for video analytics with differential privacy. *Proc Priv Enhancing Technol* 2020(4):277–296.

[81] Van Den Hooff J, Lazar D, Zaharia M, Zeldovich N (2015) Vuvuzela: Scalable private messaging resistant to traffic analysis. *Proceedings of the 25th Symposium on Operating Systems Principles*, pp 137–152.

[82] Erlingsson Ú, Pihur V, Korolova A (2014) Rappor: Randomized aggregatable privacy-preserving ordinal response. *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp 1054–1067.

[83] (2024) Apple differential privacy technical overview. Available at https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf.

[84] Bittau A, Erlingsson Ú, Maniatis P, Mironov I, Raghunathan A, Lie D, Rudominer M, Kode U, Tinnes J, Seefeld B (2017) Prochlo: Strong privacy for analytics in the crowd. *Proceedings of the 26th symposium on operating systems principles*, pp 441–459.

[85] Mironov I, Pandey O, Reingold O, Vadhan S (2009) Computational differential privacy. *Annual International Cryptology Conference* (Springer), pp 126–142.

[86] Roy Chowdhury A, Wang C, He X, Machanavajjhala A, Jha S (2020) Cryptε: Crypto-assisted differential privacy on untrusted servers. *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp 603–619.

[87] Roth E, Zhang H, Haeberlen A, Pierce BC (2020) Orchard: Differentially private analytics at scale. *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, pp 1065–1081.

[88] Mironov I (2012) On significance of the least significant bits for differential privacy. *Proceedings of the 2012 ACM conference on Computer and communications security*, pp 650–661.

[89] Papernot N, Steinke T (2021) Hyperparameter tuning with renyi differential privacy. *arXiv preprint arXiv:211003620* .

[90] Joint Task Force Transformation Initiative Interagency Working Group (2020) Security and privacy controls for federal information systems and organizations (National Institute of Standards and Technology, Gaithersburg, MD), NIST Special Publication (SP) 800-53, Rev. 5. https://doi.org/10.6028/NIST.SP.800-53r5

[91] Boyens J, Smith A, Bartol N, Winkler K, Holbrook A, Fallon M (2024) Cybersecurity supply chain risk management practices for systems and organizations. https://doi.org/https://doi.org/10.6028/NIST.SP.800-161r1-upd1. Available at https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=958681

[92] Cummings R, Kaptchuk G, Redmiles EM (2021) " i need a better description": An investigation into user expectations for differential privacy. *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp 3037–3052.

[93] (2021) U.S. Census Bureau Press Release CB21-CN.42: Census Bureau sets key parameters to protect privacy in 2020 census results. Available at https://www.census.gov

/newsroom/press-releases/2021/2020-census-key-parameters.html.

[94] Balle B, Wang YX (2018) Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. *International Conference on Machine Learning* (PMLR), pp 394–403.

Appendix A. Glossary

trust model A collection of assumptions that characterize the trustworthiness of each component in a system.

absolute error The absolute difference between the noisy and unaltered versions of a query's output.

access control policies Policies that describe who is allowed to access the data and/or which parts of the data.

accuracy The degree to which the noisy and unaltered versions of a query's output differ.

average query A query that determines the mean of some set of values. Adapted from [22].

bounded differential privacy A unit of privacy variant that calls two datasets D_1 and D_2 neighbors if it is possible to construct D_2 from D_1 by **changing** one person's data. Under bounded differential privacy, neighboring datasets have the same size. Bounded differential privacy allows for mechanisms that release the total size of the dataset with no noise.

clamping See *clipping*.

clipping The general name for any algorithm that enforces a bound on the impact of one user's data on an aggregate statistic. A common example is enforcing lower and upper bounds on values being summed in order to bound the global sensitivity of the sum.

clipping parameter The specific choice of lower and upper bounds that are used when an algorithm performs *clipping*. The utility of a differentially private algorithm is often dependent on choosing good clipping parameters. One must be careful not to compute the clipping parameter directly from the data, as doing so may lead to a violation of privacy.

counting query A query that counts the number of rows in a dataset with a particular property. Adapted from [22].

data consumer(s) In a trust model for differential privacy, the data consumers are those who receive differentially private results.

data curator In a trust model for differential privacy, the data curator is where the data is aggregated.

data subjects In a trust model for differential privacy, the data subjects are those who the data is about.

differential privacy A mathematical framework that quantifies privacy risk to individuals as a consequence of data collection and subsequent data release. Adapted from [15].

differentially private synthetic dataset A synthetic dataset that is produced by mechanisms that satisfy differential privacy. Adapted from [22].

event-level privacy A unit of privacy that defines neighboring databases as those that differ in one event, for example, a single transaction, or a single row. Adapted from [22].

fine-tuning In machine learning, a training step that starts from a pre-trained model (sometimes called a foundation model) and adds task- or domain-specific information.

gaussian mechanism An algorithmic primitive for differential privacy that adds random noise sampled from the Gaussian distribution to the output of a query. Adapted from [22].

group privacy A property of differential privacy. It says that if a mechanism provides differential privacy for one person, then it also provides a weaker differential privacy guarantee for groups of people. The weakness of the guarantee depends on the size of the group, and the definition of one person depends on the *unit of privacy* used.

high-dimensional A statistic composed of many numbers—e.g. a histogram with 50,000 bins, or a vector with 1 million elements.

human bias A form of bias that results from failures in the heuristics humans use to make decisions. Adapted from [40].

hyperparameter In a differential privacy mechanism, a setting or parameter that controls a portion of the mechanism's behavior or execution. The best setting may be data-dependent, and a method that uses the confidential data as the basis for these parameters would not satisfy differential privacy. Examples include the clipping parameter for mechanisms that perform clipping, the number of iterations for iterative algorithms, and the learning rate or minibatch size for machine learning algorithms.

identifying information Information that could be used to identify a specific individual, such as name, address, phone number, or identification number.

laplace mechanism An algorithmic primitive for differential privacy that adds random noise sampled from the Laplace distribution to the output of a query. Adapted from [15].

linking attack An approach for exposing information specific to individuals in a de-identified dataset by matching up records with a second dataset.

low-dimensional A statistic composed of few numbers—e.g. a single count, or a histogram with 5 bins.

neighboring datasets The definition of neighboring datasets is a parameter to the differential privacy framework. In many contexts, two databases are considered neighbors if they differ in the data of one individual. Adapted from [15].

outcome-specific utility metrics A way of measuring the utility of data for answering a specific question or class of questions.

post-processing invariance A property of differential privacy. It says that the output of a differentially private mechanism remains differentially private, even if further processing is performed on it.

pre-training In machine learning, a training step that trains a general-purpose model (sometimes called a foundation model) on publicly-available data. Pre-training is often followed by *fine-tuning* to equip the model with task-specific information.

privacy budget An upper bound on allowable cumulative *privacy loss* across all analyses that process a single dataset.

privacy loss A quantitative upper bound on the statistical distance between analysis outcomes on neighboring datasets.

privacy parameter A parameter of a differential privacy definition that partly or wholly determines *privacy loss*.

privacy-utility tradeoff The fundamental tension between privacy and accuracy. Adding more noise increases privacy but reduces accuracy, and vice-versa.

reconstruction attack A privacy attack that uses published statistics to reconstruct individual data points from the original private dataset.

relative error The absolute error divided by the unaltered query output.

sensitivity A quantity that measures how much the output of a query could change as a function of a change to the input. Adapted from [15].

statistical bias A form of bias that occurs when the expected value of a released statistic does not match the true statistic.

subsampling An algorithmic strategy where the query output is computed using only a fraction of the original data, selected at random. Adapted from [22].

summation query A query that sums a derived quantity from each row in a dataset with a particular property. Adapted from [22].

synthetic dataset An alternative dataset that differs from the original, but also maintains specific properties inherent to the original, such as correlations between attributes. Adapted from [22].

systemic bias A form of bias that results from rules, processes, or norms that advantage certain social groups and disadvantages others. Adapted from [40].

trust assumption An assumption that characterizes how one expects a specific party to behave when given access to sensitive data.

trusted party A party that can be expected to keep sensitive data safe and not disclose it to others.

unbounded differential privacy A unit of privacy variant that calls two datasets D_1 and D_2 neighbors if it is possible to construct D_2 from D_1 by **adding or removing** one person's data. Under unbounded differential privacy, neighboring datasets have different sizes.

unit of privacy The choice of definition for neighboring datasets. Adapted from [22].

unstructured data Data formats that often lack explicit structure that relates data to individuals, such as text, pictures, audio, and video.

untrusted party A party that cannot be expected to keep sensitive data safe or refrain from disclosing it to others.

user-level privacy A unit of privacy that defines neighboring databases as those that differ in one user's data. Adapted from [22].

utility The degree to which a dataset or statistic is useful for a specific purpose.

Appendix B. Technical Details

Appendix B.1. Definition of (ε, δ)-Differential Privacy

Formally, (ε, δ)-differential privacy is a simple change to the original definition that adds an additive δ parameter to the original inequality. The formal definition appears in Definition 3. Setting $\delta = 0$ makes the (ε, δ) definition equivalent to the original pure ε definition (i.e., making catastrophic failure impossible).

Definition: Approximate Differential Privacy. A randomized mechanism $\mathscr{M}$ satisfies (ε, δ)-differential privacy if for all neighboring datasets D_1 and D_2 and all possible outcomes S:

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

D_1 and D_2 are considered *neighbors* if they differ in the data of one individual.

The other variants in Table 1 use slightly different ways of measuring the distance between the probability distributions $\mathscr{M}(D_1)$ and $\mathscr{M}(D_2)$. Rényi differential privacy and zero-concentrated differential privacy bound this distance using *Rényi divergence*, while Gaussian differential privacy does so using *f-divergences*.

Appendix B.2. Definitions of Sensitivity and Basic Mechanisms

The formal definition of L_1 sensitivity is:

Definition: L_1 Sensitivity. For a function $f : D \rightarrow \mathbb{R}^k$, the L_1 sensitivity Δ_1 of f is:

$$\Delta_1 = \max_{\text{neighboring } D_1, D_2} \|f(D_1) - f(D_2)\|_1$$

where D_1 and D_2 are neighboring datasets according to the unit of privacy.

This definition works for any function (or query) that outputs a vector of real numbers (including a single real number, like most aggregation functions). It defines sensitivity to be the maximum L_1 distance between the function's outputs for two inputs that differ by one unit of privacy (discussed in Sec. 2.4). The corresponding definition for L_2 distance is called L_2 *sensitivity*:

Definition: L_2 Sensitivity. For a function $f : D \rightarrow \mathbb{R}^k$, the L_2 sensitivity Δ_2 of f is:

$$\Delta_2 = \max_{\text{neighboring } D_1, D_2} \|f(D_1) - f(D_2)\|_2$$

where D_1 and D_2 are neighboring datasets according to the unit of privacy.

Both definitions measure the impact of "one unit of privacy change" on the output of the function to determine how much noise needs to be added for privacy. For the user-level unit of privacy, sensitivity corresponds to the impact of *one person's data* on the function's output, which corresponds with the intuition for differential privacy given earlier.

Mechanism: Laplace mechanism [22]. For a query with L_1 sensitivity Δ_1, the **Laplace mechanism** adds noise sampled from the Laplace distribution with center 0 and scale $\frac{\Delta_1}{\varepsilon}$.

Guarantee: $(\varepsilon, 0)$-differential privacy

Mechanism: Gaussian mechanism [94]. For a query with L_2 sensitivity Δ_2, $\varepsilon \geq 0$, and $0 \leq \delta \leq 1$, the **Gaussian mechanism** adds noise sampled from the Gaussian (Normal) distribution with center 0 and variance σ^2 . The mechanism satisfies (ε, δ)-differential privacy if:

$$\Phi\Big(\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\Big) - e^{\varepsilon}\Phi\Big(-\frac{\Delta}{2\sigma} - \frac{\varepsilon\sigma}{\Delta}\Big) \leq \delta$$

where Φ is the CDF of the Gaussian distribution.

Guarantee: (ε, δ)-differential privacy

The difference between Laplace and Gaussian noise comes from the type of sensitivity used for each mechanism: L_1 sensitivity Δ_1 for Laplace and L_2 sensitivity Δ_2 for Gaussian. For large vectors of results, $\Delta_2 \ll \Delta_1$. For a single count, $\Delta_2 = \Delta_1 = 1$. The Gaussian mechanism offers much better accuracy in the former setting, while the Laplace mechanism offers better accuracy in the latter. When many counts are requested at the same time, $\Delta_2 \ll \Delta_1$, and the Gaussian mechanism should be used.

Appendix B.3. Details: Counting Queries

The Laplace mechanism can be used to ensure differential privacy for counting queries if the L_1 sensitivity Δ_1 of the query is determined. For simple scalar-valued counting queries, the sensitivity is always 1 (assuming the unbounded neighbors model). The final count can only change by 1 when a single individual's data are added or removed. This argument holds no matter what the property is or the columns being grouped. Note that the argument only applies when no transformation in the unity of privacy is desired. When a transformation in the unit of privacy is needed (e.g., bounding user contributions), then the sensitivity of counting queries goes up.

Key Takeaway: Counting queries and histograms have a sensitivity of 1 when no transformation in the unit of privacy is desired.

The simple sensitivity analysis for counting queries makes them good targets for differential privacy. They are easy to implement and can often give highly accurate results because the sensitivity is low. To achieve differential privacy for counting queries, including the examples in this section, under unbounded differential privacy when each user contributes one row to the dataset, the Laplace mechanism with $\Delta_1 = 1$ and the desired setting for the privacy parameter ε are applied. For histograms, the Laplace mechanism with $\Delta_1 = 1$ and the same setting for ε can be applied when the bins are specified by the analyst. The noisy results satisfy $(\varepsilon, 0)$-differential privacy.

Appendix B.4. Details: Summation Queries

To achieve differential privacy for a summation query, the L_1 sensitivity Δ_1 of a summation query is needed. How much a summation query changes when a row is added to a database depends on the row. If someone spends $1 on a pumpkin spice latte, then the increase in the sum will be $1. If someone spends $10,000, the sum will increase much more.

Achieving differential privacy requires an upper limit on the *largest possible increase* there can be when a row is added or modified. For the latte query, that means an upper limit on the price of a pumpkin spice latte. This is a big challenge because no matter what limit is set, there may hypothetically be a cafe somewhere that charges more than the limit.

The solution to this problem is called *clipping*. The idea is to *enforce* an upper limit rather than assuming one. Lattes that cost more than the limit are *clipped* so that their price is equal to the limit. After clipping, all values in the database are guaranteed to fall between the lower and upper limits that were set. The guaranteed lower and upper bounds on the data can be used to determine sensitivity. If the data are clipped so that lattes cost at most $10, then the largest increase in the output of the summation query will be $10 when a single latte sale is added to the database.

The following process can be used to achieve differential privacy:

1. Clip each value v in the dataset so that $0 < v < C$.
2. Sum the clipped values.
3. Apply the Laplace mechanism with $\Delta_1 = C$ and the desired privacy parameter ε.

The first step in the process enforces bounded sensitivity, which informs how Δ_1 is set in the third step. This approach satisfies ε-differential privacy.

Appendix B.5. Details: Average Queries

Unfortunately, bounding the sensitivity of average queries is even more difficult than it is for summation queries. In addition to the upper limit on the data values themselves, how much an average changes after a row is added depends on *how many things are being averaged*. If one is averaging five numbers, then adding one more number might change the average by quite a bit. If one is averaging 5 million numbers, then adding one more probably would not change the average very much. As a general rule, however, the sensitivity of a query should not depend on the data. Otherwise, the sensitivity might itself be

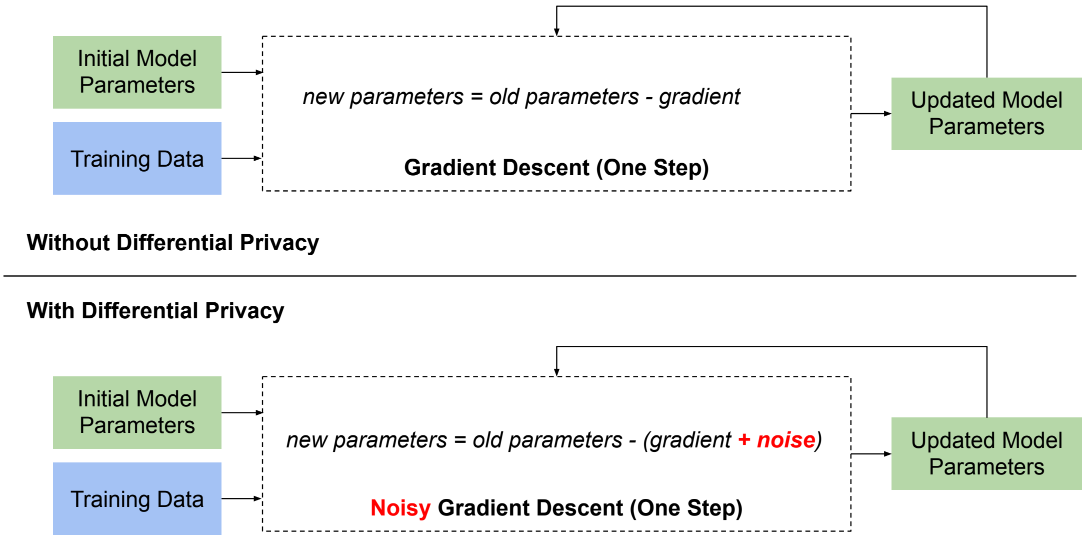

Initial Model Parameters
Training Data
new parameters = old parameters - gradient
Gradient Descent (One Step)
Updated Model Parameters
Without Differential Privacy
With Differential Privacy
Initial Model Parameters
Training Data
new parameters = old parameters - (gradient + noise)
Noisy Gradient Descent (One Step)
Updated Model Parameters

Fig. 23. Noisy gradient descent for differentially private machine learning

sensitive, meaning that it might reveal something about the data. This adds another level of complexity to bounding the sensitivity of averages.

A simple and effective solution for answering an average query using differential privacy is to split the query into two separate queries: a summation query and a counting query. To split the example query, the two following queries are computed instead:

1. What has been the total amount spent on pumpkin spice lattes since 2010?
2. How many pumpkin spice lattes have been purchased since 2010?

The first is a summation query, and the second is a counting query. The desired average can be obtained by dividing the first by the second. By the *composition* and *post-processing* properties of differential privacy, if differentially private answers to both queries are computed, their quotient also satisfies differential privacy. Therefore, the following process can be used to compute the average:

1. Compute the differentially private sum s with privacy parameter ε_1.
2. Compute the differentially private count c with privacy parameter ε_2.
3. Return the average $\frac{s}{c}$.

This process satisfies $\varepsilon_1 + \varepsilon_2$-differential privacy. For a desired privacy parameter ε, $\varepsilon_1 = \varepsilon_2 = \frac{1}{2}\varepsilon$ is typically set to equally “split” the privacy budget across the two constituent queries.

Appendix B.6. Details: Differentially Private Stochastic Gradient Descent

Figure 23 summarizes the difference between traditional non-private gradient descent and the noisy version that satisfies differential privacy. The non-private gradient descent algorithm performs many steps (or *iterations*) of the *gradient update rule*. This rule first computes the *gradient of the loss* for the current model. The *loss* quantifies how *badly* the model is performing on the training data, and the gradient's value directs how to change the model parameters to *increase* the loss. To *minimize* the loss in order to train a model that performs well, the *opposite* change is made by subtracting the gradient from the current parameters. This process is repeated many times until the model achieves the desired performance. To satisfy differential privacy, the noisy gradient descent algorithm selects a small minibatch of examples to use in the gradient calculation (which amplifies the privacy guarantee), and adds noise to the gradient before updating the model parameters [58]. Since the training data are *only* used to calculate the gradient, adding noise to the gradient is sufficient to allow the whole algorithm to satisfy differential privacy.

Noisy gradient descent adds noise to the gradient. To determine how much noise to add, the sensitivity of the gradient computation must be analyzed. In many settings, including deep neural networks, the gradient computation is complex and can have extremely high global sensitivity. For this reason, the *differentially private SGD (DP-SGD)* algorithm [58] *enforces* sensitivity rather than measures it. To enforce an upper bound on sensitivity, the algorithm clips the gradient associated with each training example, similar to the summation queries discussed earlier. Clipping the per-example gradients ensures bounded global sensitivity for the aggregated gradient used in the gradient update rule and informs how much noise is needed.

The primary alternative to DP-SGD is a technique that trains many separate models on subsets of the training data and aggregates the models themselves with a differentially private aggregation function [77]. This approach can provide more accuracy than DP-SGD for the same level of privacy, but it incurs significant computational cost because it requires training many models.

Appendix B.7. Evaluating Software Libraries for Differential Privacy

Because of the difficulty of implementing differential privacy mechanisms safely and correctly, it is good practice to use existing, actively-maintained software rather than writing custom implementations. Below are a few questions that can help prospective users of differential privacy evaluate software tools.

- Does the library adequately address known issues with differential privacy implementations? Maintainers of software libraries should be able to confidently explain what their approach is to mitigating floating-point issues, backend issues, and (in the untrusted analyst model) timing channels.
- Does the library allow performing end-to-end computation on the data? Using a

robust software library for basic mechanisms like noise addition is generally safer than implementing this from scratch, but still leaves a lot of room for error. By contrast, software frameworks that encapsulate the entire mechanism and perform automatic privacy accounting can prevent unintended privacy leakage.

- Is the library open-source? In open-source software, the privacy claims can be independently verified and audited by the differential privacy community, which is a positive sign. Conversely, in proprietary software, it is often much more difficult to evaluate code quality and the robustness of the implementation.
- Is the library well-tested and well-documented? Test coverage and documentation are indicators of software quality, and for privacy-critical software like differential privacy libraries, software quality is an essential component of robustness.
- Is the library actively used and maintained? An active user base can help to discover bugs and privacy vulnerabilities in the software, and an effective maintenance process helps to fix them quickly.
- Was the library audited by independent third-parties or proven correct using formal methods? In-depth audits of software projects take time and resources, and third-party auditors can help bring independent validation of the robustness of a differential privacy library. Formal methods can provide an additional form of oversight, by proving that the software correctly implements differential privacy.