Differential Privacy with Fine-Grained Provenance: Opportunities and Challenges

Xi He
University of Waterloo
xi.he@uwaterloo.ca

Shufan Zhang
University of Waterloo
shufan.zhang@uwaterloo.ca

Abstract

Differential privacy (DP) offers a robust framework for protecting individual privacy when analyzing data. However, the elegant abstractions used in DP theory do not always translate seamlessly to real-world systems. For example, the basic DP system maintains a global privacy budget and updates it when it answers a query. However, tracking this level of privacy information cannot character the privacy loss for heterogeneous data types, complex DP mechanisms, or the privacy loss towards different data analysts. Several DP systems have shown success by tracking more fine-grained information about privacy usage. Inspired by this, we propose a novel perspective: leveraging fine-grained privacy provenance to build practical DP systems. First, we propose a taxonomy of privacy provenance for DP, including why-, how-, and where-DP-provenance that characterizes different aspects of DP. Next, we review several systems that feature different techniques in each category of privacy provenance. Through this unique lens, we summarize the open challenges for DP systems and future directions for deeper integration between DP and data provenance techniques.

1 Introduction

Differential privacy (DP) [40] emerged in 2006 as a groundbreaking concept for protecting individual privacy in data analysis. DP offers a powerful privacy-preserving approach by mathematically ensuring that data releases reveal minimal information about any single person. This has led to the development of numerous DP mechanisms and systems like PINQ [88], FLEX [65], PrivateSQL [75], GoogleDP [5], and Chorus [66]. However, despite its theoretical elegance and strong privacy guarantees, DP's practical deployments lag behind its potential. While a few pioneering cases exist, such as the 2020 US Census disclosure [1, 57], widespread adoption remains limited. Companies like Amazon, Snowflake, Google, LinkedIn, Uber, and Apple, and startups like Tumlut Labs and Transcend are exploring DP in data management products or statistical learning scenarios [4, 111, 53, 128, 60, 107, 27, 96, 118, 117], but these integrations are often experimental and face challenges in production environments [114, 47, 134], suggesting a divergence between theory and practice.

In theory, DP relies on a privacy budget, represented by the parameters (ϵ, δ), which controls the overall privacy guarantee. By carefully injecting controllable noise, it can be proved that a mechanism can only reveal bounded information about any individual in a *static* dataset, and thus, this mechanism satisfies the notion of DP.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

However, translating this theory into practical systems presents several challenges. First, even with a simple use case that only focuses on the central DP setting[1], the system has to interact with private data and data analysts and has to maintain the system budget at least correctly and faithfully. Some systems resort to large or frequently reset budgets [107, 4], which may jeopardize long-term privacy. Second, DP systems often assume static datasets. However, real-world data can be dynamic, integrating information from various sources [78] and undergoing regular updates [22]. The individuals in the dataset may have different privacy awareness [67]. These complexities require additional considerations to maintain privacy guarantees. Third, DP systems need to cater to analysts with varying levels of privacy expertise. Non-expert analysts may struggle to interpret noisy results or choose the most appropriate DP mechanism for their queries [48], especially for complex queries, like nested subqueries or batched workloads [95, 29]. As a result, implementations of the DP system can fail to deliver an optimized privacy-utility trade-off or the expected privacy guarantees.

Recent DP systems have addressed one specific challenge mentioned above by tracking fine-grained information such as the data blocks [80, 78] or the noise used for previous queries [87, 137]. Inspired by these works, we propose a broader approach to building usable end-to-end DP systems by leveraging the data provenance framework in databases [18], in which tracing and propagating *proper provenance metadata* turns out to be useful for understanding queries, integrating data, and debugging inconsistencies. Similarly, in DP systems, we envision privacy provenance — metadata that tracks the DP mechanisms and benefits the users of the systems.

In this work, we analyze different components of a DP system and explore how proper provenance metadata can offer benefits, including improved user understanding, enhanced utility optimization, and dynamic privacy management. We start with three types of privacy provenance: *why-DP-provenance*, which explains the private/noisy outputs to the data analysts; *how-DP-provenance*, which uses metadata for tighter privacy in running DP mechanisms; and *where-DP-provenance*, which tracks privacy budget consumption over dynamic data sources to satisfy different resolutions of privacy definitions. While the concept of privacy provenance is a recent development [138], the idea of leveraging additional data structures in DP algorithm design has been explored in prior work [66, 88, 46, 90]. We surveyed all the relevant DP works in our privacy provenance framework and provided discussion in this direction. Note that our characterization of privacy provenance in this work is not meant to be exhaustive. We hope this work stimulates further exploration and discussion in developing more usable and optimized systems for DP. This work also aims to complement existing visions on DP (including recent surveys [23, 92]). By introducing a system-oriented perspective through the lens of privacy provenance, we hope to pave the way for the development of more usable and optimized DP systems in the future.

Article Roadmap. The remainder of this article is organized as follows. Section 2 summarizes the preliminaries of differential privacy and provenance in databases. In Section 3, we provide a systematic view of the complexity of the DP systems and propose a taxonomy of fine-grained privacy provenance. We survey several systems that feature the usages of fine-grained privacy provenance in Section 4 and discuss challenges and future directions in Section 5. We conclude this article in Section 6.

2 Background

We introduce and summarize the related definitions of differential privacy. Next, we introduce the necessary preliminaries for differential privacy and provenance in databases.

2.1 Definition of Differential Privacy

[Differential Privacy [40]]

We say that a randomized algorithm $\mathcal{M} :\rightarrow \mathcal{O}$ satisfies (ϵ, δ)-differential privacy (DP), if for any two

[1]Assuming the existence of a trusted curator runs a (data analytics) system with DP guarantees for a curated sensitive dataset.

neighbouring databases (D, D') that differ in only 1 tuple, and $O \subseteq \mathcal{O}$, we have

$$\Pr[\mathcal{M}(D) \in O] \leq e^{\epsilon} \Pr[\mathcal{M}(D') \in O] + \delta.$$

[Global Sensitivity] For a query $q : \mathcal{D} \to \mathbb{R}^d$ the ℓ_2 global sensitivity of this query is

$$\Delta q = \max_{D,D':d(D,D')\leq 1} \|q(D) - q(D')\|_2,$$

where $d(\cdot, \cdot)$ denotes the number of tuples that D and D' differ and $\|\cdot\|_2$ denotes the ℓ_2 norm. If we replace the ℓ_2 norm with ℓ_1 norm, then we obtain the ℓ_1 sensitivity $\Delta_1 q$.

Basic DP Mechanisms. The most basic DP mechanisms are the Laplace and Gaussian mechanisms, injecting Laplace and Gaussian noises, respectively, into the query answers, which are explained below.

[Laplace Mechanism [39]] Given a numerical query $q : \mathcal{D} \to \mathbb{R}^d$, the Laplace mechanism outputs $\mathcal{M}(D) = q(D) + \eta$ where $\eta \sim Lap\,(b)^d$ where $Lap\,(b)^d$ is a vector of d i.i.d. samples from a Laplace distribution with scale b. If $b = \Delta_1 q/\epsilon$, then the Laplace mechanism preserves $(\epsilon, 0)$-DP.

[Gaussian Mechanism [39]] Let $\epsilon \in (0, 1)$. Given a numerical query $q : \mathcal{D} \to \mathbb{R}^d$, for constant $c > \sqrt{2\ln(1.25/\delta)}$, the Gaussian mechanism adds the noise vector $(\eta_1, \eta_2, \ldots, \eta_d)$ to the query answer $q(D)$, where η_i are i.i.d. random variables drawn from the Gaussian distribution $\mathcal{N}(0, \sigma^2 I)$ with $\sigma > c\Delta q/\epsilon$. The Gaussian mechanism is (ϵ, δ)-differentially private.

The standard Gaussian mechanism [39] has the limitation that it can only be used in a high privacy regime, where the privacy parameter ϵ should be within the range of $(0, 1)$. Balle and Wang [6] propose an improved mechanism, namely the analytic Gaussian mechanism, overcoming this limitation in the standard Gaussian mechanism. We give the definition of analytic Gaussian mechanism below for completeness, while skipping the details of the mechanism does not affect understanding this article.

[Analytic Gaussian Mechanism [6]] Given a query $q : \mathcal{D} \to \mathbb{R}^d$, the analytic Gaussian mechanism $\mathcal{M}(D) = q(D) + \eta$ where $\eta \sim \mathcal{N}\left(0, \sigma^2 I\right)$ is (ϵ, δ)-DP if and only if

$$\Phi_{\mathcal{N}}\left(\frac{\Delta q}{2\sigma} - \frac{\epsilon\sigma}{\Delta q}\right) - e^{\epsilon}\Phi_{\mathcal{N}}\left(-\frac{\Delta q}{2\sigma} - \frac{\epsilon\sigma}{\Delta q}\right) \leq \delta,$$

where $\Phi_{\mathcal{N}}$ denotes the cumulative density function (CDF) of Gaussian distribution. In this mechanism, the Gaussian variance is determined by $\sigma = \alpha\Delta q/\sqrt{2\epsilon}$ where α is a parameter determined by ϵ and δ [6].

Privacy Composition Theory. Differential privacy enjoys the nice property of being compositional. Running a DP mechanism multiple times is also DP, but with a higher privacy cost. One can, therefore, build complex mechanisms by composing basic DP mechanisms using the following composition theorems.

[Sequential Composition [39]] Given two mechanisms $\mathcal{M}_1 :\to \mathcal{O}_1$ and $\mathcal{M}_2 :\to \mathcal{O}_2$, such that $\mathcal{M}_1$ satisfies (ϵ_1, δ_1)-DP and $\mathcal{M}_2$ satisfies (ϵ_2, δ_2)-DP. The combination of the two mechanisms $\mathcal{M}_{1,2} :\to \mathcal{O}_1 \times \mathcal{O}_2$, which is a mapping $\mathcal{M}_{1,2}(D) = (\mathcal{M}_1(D), \mathcal{M}_2(D))$, is $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$-DP.

[Parallel Composition [88]] Let a mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{O}$ be (ϵ, δ)-DP. If $D_1, \ldots, D_n \in \mathcal{D}$ are n arbitrary disjoint sets of databases and $X = D_1 \cup \cdots \cup D_n$, the release of mechanism output sequence $\mathcal{M}(D_1), \ldots, \mathcal{M}(D_n)$ satisfies (ϵ, δ)-DP.

[Post Processing [39]] For an (ϵ, δ)-DP mechanism $\mathcal{M}$, applying any arbitrary function f over the output of $\mathcal{M}$, that is, the composed mechanism $f \circ \mathcal{M}$, satisfies (ϵ, δ)-DP.

2.2 Provenance in Databases

Provenance (sometimes also called lineage or pedigree) in databases [18] studies the origin or history of the data through a high-level structured computation (e.g., data transformation or query execution) in its lifecycle. By recording more metadata through annotation [15] or additional data structure (e.g., semiring) [55, 54], data

management systems can explain the relationships on how data items in the output (e.g., a tuple produced) depend on (various) source/input (e.g., multiple relational tables). The most common provenance models are "why-", "how-", and "where-" provenances, aim to explain, respectively, why a particular tuple appears in the output [14], how an output tuple is processed and derived [55], and where, precisely, an output tuple is originated in the sources according to the computation [14]. Provenance information is useful for understanding the query behavior, data processing steps, and scientific workflows, and thus critical in auditing the integrity, reproducibility, and reliability of data in many scenarios [18].

The relationship between provenance and data privacy has received increasing attention [97, 25, 24, 26, 98, 99]. This line of research focuses on the potential privacy risks associated with tracking provenance information, particularly for sensitive data. Researchers explore how revealing provenance details could leak information and investigate techniques to minimize such risks. This might involve suppressing specific parts of provenance queries or hiding certain intermediate data while still achieving desired privacy guarantees. Other recent work [130] highlights how provenance can be used for maintaining and versioning machine learning models to mitigate privacy attacks towards the models.

3 Privacy Provenance Overview

This section explores how provenance information can be leveraged to enhance the effectiveness of DP systems. We introduce the concept of DP provenance, a systematic approach to capture and utilize provenance to support DP systems. We propose a new taxonomy that categorizes existing work based on the type and granularity of the provenance considered. We then delve into how different types of provenance can be used to improve various aspects of DP systems.

3.1 DP System Model: A Provenance Perspective

Every data system that integrates DP has to record and track some additional metadata, compared to that without DP. We start with a simple, hypothetical system with DP to show the most basic functionalities and the coarsest privacy provenance tracking provided by it. We then lay out the different entities/components of a DP system that motivate the need for finer-grained privacy provenance or additional provenance features.

A (Hypothetically) Basic DP System. Consider running a DP system in the central setting, where a trusted data curator maintains a protected database D. The data curator sets up a finite system-wise or global privacy budget to bound the overall extent of information disclosure over this database. (A group of) untrusted data analyst(s) would like to query the private database D. Each incoming query from a data analyst specifies a per-query privacy budget that indicates the amount of budget they would like to spend on this particular query. The DP system uses *a fixed mechanism* (e.g., the Laplace mechanism) to answer this query, and subtracts the per-query privacy budget from the global budget. The system rejects a query if the remaining global budget is not sufficient for this query; it stops processing queries once the global privacy budget is fully depleted.

This basic system can only support limited queries (that could be naïvely answered through one fixed mechanism). Almost every DP system that researchers implement in literature is more complicated than it. Even though, the basic system has to track the privacy loss in terms of the per-query privacy budget and the global budget consumption[2], which is the simplest example of privacy provenance. This oversimplified setting overlooks the complexity of a real-world DP system. Next, we will provide a view of the complexity of system designs for real-world DP systems.

The Complexity of DP Systems. The design of a DP system greatly depends on the users's role, interest, and expertise levels in this system. First, the data analysts (who are the queries or the programmers) of a DP

[2]Indeed, in real-world implementations which lack careful management, DP can rapidly become excessively restrictive so that service providers have to set up a large (or even infinite) global budget, which has been shown to attacks [114].

system have no direct access to the data. They care for the accuracy of the query results and how many queries they can interact with the system. Ideally, if the data analysts have the full domain knowledge of DP and how the system works, then they can understand the DP programs supported by the systems and trace the necessary meta information for optimizing their interested queries. If not, they may not be able to specify the privacy budget correctly and understand the noisy outputs. A usable DP system in practice is expected to accommodate the needs of both types of data analysts. It should explain the noisy output to DP novices and provide modularized APIs for DP experts to program their own tasks.

Second, the data curators are responsible for setting up the data input for the programs and allocating privacy budgets among different analysts. The basic DP system makes several assumptions to simplify the privacy analysis. It assumes that the database is static (i.e., not subject to updating), and each row in the database corresponds to a single individual. In addition, the privacy analysis is rigidly enforced at the entire table level, and the protection is uniform across every row in the database. In real-world use cases, however, the underlying database is often under dynamic changing and has the hetergenous nature that not every part of the data is equally sensitive. For example, new data analysts can keep opt-in, and their data will be merged into the private database when the system is running. The data contributors may have different contributions to the database in terms of the number of rows. They may also have personalized opinions or different privacy awareness regarding the protection of their data. In such scenarios, keeping a single global privacy budget and simple privacy analysis at the table level is not sufficient to provide the desired level of privacy guarantees. In addition, data analysts can have different trust or privilege levels when accessing the system. Tech companies, for example, need to query their users' data for internal applications like anomaly detection. They also consider inviting external researchers with low privilege levels to access the same sensitive data for study through the shared query interface. It is unfair if a data analyst with a low privilege level asks queries with a significant portion of the global budget so that higher privileged analysts have no more budget to consume. Therefore, a real DP system involving multiple data analysts may have a range of design choices to make, from privilege allocation to query/task scheduling to assumptions on whether the analysts can collude, etc.

Third, the DP programs responsible for mapping the utility interests of data analysts and the privacy interests of data curators have a wide range of complexities. Some have a fixed query template with a deterministic sensitivity and hence noise scale (e.g., stability tracking in PINQ [88]); some require dynamic sensitivity analysis (e.g., query rewriting in Chorus [66]); and most systems deal with more than one query [88, 66, 48, 44]. The queries to be supported can be complicated, e.g., involving multiple transformations of data, and be batched into an OLAP workload. The desirable system should support multiple mechanisms (including customized ones) to answer different types of queries and additional algorithms to choose among the mechanisms to optimize the privacy-utility trade-off for the queries (especially for the workloads).

In every stage/aspect of the DP system, we have observed evidence or challenges that the basic DP system or coarse-grained DP provenance cannot address. We envision that a framework with *fine-grained privacy provenance* can help solve these challenges and ensure a more usable and optimized DP system for practical needs, which is described next.

3.2 The Taxonomy of Fine-Grained Privacy Provenance

We draw an analogy to provenance in databases and, similarly, characterize the privacy provenance into three categories: "why-DP-provenance" (or output-provenance), "how-DP-provenance" (or process-provenance), and "what-DP-provenance" (or input-provenance), based on the information/metadata they track and the aspects they influence a DP system.

Why-DP-Provenance, also known as output-provenance, refers to the additional information that helps data analysts understand the reasoning behind a DP system's output. It focuses on the interaction between the system's results and the analyst's needs. Why-DP-provenance aims to answer questions like, why is a specific privacy budget chosen for the analyst's query, how does the level of noise in the answer achieve the desired accuracy,

Table 3: Comparison between different types of provenance in databases [18] and privacy provenance.

	Database Provenances [18]	**Privacy Provenances**
Why-(DP-)Provenance	Identifying sub-instances of the input that "witness" a part of the output	Explaining why a noisy output satisfies accuracy requirements and/or why certain queries are rejected
How-(DP-)Provenance	Providing additional information on how the output tuple is derived, e.g., transformations applied during processing	Tracing query metadata for tighter privacy analysis during executing DP program/mechanisms (that are complex or involving multiple data analysts)
Where-(DP-)Provenance	Pinpointing where an attribute value in the output tuple is exactly copied from	Managing budget consumption for heterogeneous private input data sources (e.g., user-level DP, personalized DP)

why is a noisy answer still considered useful, and what factors lead to a query rejection? Benefits of enforcing fine-grained why-DP-provenance include:

- Explainability and Interpretability: Why-DP-provenance details query answers' usefulness and confidence intervals, which allows analysts to understand the trade-offs between privacy guarantees and accuracy.
- Tighter Privacy Composition: Why-DP-provenance facilitates a more accurate understanding of how different queries impact the overall privacy budget.
- Optimal Privacy-Accuracy Trade-Offs: Why-DP-provenance empowers analysts to make informed decisions about the balance between privacy and the usefulness of results.

Techniques for why-DP-provenance device approaches for accuracy-first query specification, error specification, fine-grained budget specification, and others. Unlike the why-provenance in databases that focuses on "why" something happened or exists, the why-DP-provenance answers "why" a specific level of parameters in a DP algorithm is chosen and its impact on results within a DP system.

How-DP-Provenance, also known as process-provenance, focuses on capturing details about the specific privacy mechanisms employed within a DP system. This information becomes crucial for answering complex queries privately, building complex DP mechanisms, or selecting the optimal/best DP mechanisms for a query workload. How-DP-provenance solves the following questions: How is the noise for a private query calculated based on a series of transformations of data? What structural properties of a query can be used for a tighter privacy bound? Can we reuse the noise calibrated to historical queries for answering new queries to save the privacy budget? Fine-grained how-DP-provenance tracking enables:

- Efficient Privacy Analysis: How-DP-provenance tracks the series of modularized transformations or the query structural properties for complex queries to make sensitivity analysis more efficient.
- Better Utility for Query Workloads: How-DP-provenance reuses noise from historical query answers or injects correlated noise to a batch of queries to increase the overall utility for the workload.
- Effective Budget Allocation: How-DP-provenance facilitates the budget allocation across multiple data analysts so that the privacy loss is tightened when analysts collude.

Techniques that support fine-grained how-DP-provenance include transformation tracking and noise tracking. Different from the how-provenance in databases that generates a derivability relationship between output and

input tuples, the how-DP-provenance answers "how" a better DP mechanism could be designed or modularized for answering queries.

Where-DP-Provenance, also known as input-provenance, accounts for which (part of) private input data is used for a DP mechanism. The complex data type can involve multiple database tables, and queries over different tables can result in different DP guarantees — tracking which tables are used for what queries becomes essential for multi-relational DP systems. Furthermore, even with a single table, different queries or mechanisms may be interested in different parts of the table — accounting privacy loss at the table-level can waste more privacy budgets. Third, data may be merged at different timestamps or belong to different users. Thereby, where-DP-provenance tracks privacy loss for data by answering questions of what part of the private data is used to answer a particular query, what (group of) users are associated with the data. With where-DP-provenance, the system gains:

- Flexibility with Multi-Resolution Privacy: Where-DP-provenance allows data curators to specify policies with different resolutions of privacy guarantees.
- Continuously Running System: Where-DP-provenance enables the replacement of retired data with new data so that the system can continuously execute.

While the where-provenance in databases pinpoints the exact places that an output tuple is copied from, the where-DP-provenance answers "where" incurs a privacy loss in the private input w.r.t queries.

Next, we will describe the DP techniques that fit into the framework of each why, how, and where-DP-provenance and the different granularities of these techniques.

3.2.1 Techniques for Why-DP-Provenance

Query Specification. Two types of DP query specification exist: privacy-first and accuracy-first. Most (traditional) DP building blocks or systems are privacy-first, meaning that they require the data analysts to specify a privacy budget for their queries or tasks to run. This privacy budget will be deducted from the global privacy budget if the tasks are executed and the noisy answers are returned to the analysts. While this approach is easy to analyze and has many optimal and off-the-shelf mechanisms developed over the years, it may limit the usability of a DP system: the data analysts care about the quality of the query answers, but may not have sufficient expertise to understand the DP mechanisms and the relationship between privacy and the error rate according to the chosen mechanism. Since a DP system cannot release the true query answer, a single noisy output cannot explain how well the black-box mechanism works to privacy novice analysts. In addition, the released answers (or consumed budget) can not be reversed—discarding unsatisfactory query answers simply wastes the global privacy budget.

Another approach for query specification is *accuracy-first* [48, 50, 138, 126, 132, 83, 87, 100], that allows the data analysts to superimpose an accuracy requirement in the query specification, and the system can translate the accuracy requirements into privacy budget and automatically choose the optimal mechanism to answer this query. This approach aims to close the discrepancy between privacy-oriented DP mechanisms and the needs of data analysts for understanding noisy outputs, but many open problems remain in understanding data-dependent accuracy translation. Recent work like DPella [121] and DProvDB [138] support both the privacy- and accuracy-oriented modes.

Error Specification. A line of work, including DPComp [58], Overlook [116], PSI [46], Bittner et al. [12], DPP [64], ViP [90], and DPXPlain [115], aims to provide interfaces for explaining the noisy output to the data analyst or visualizing an optimal privacy-utility trade-off due to the mechanism to help the analysts make informed choices. These tools explore a DP confidence interval of the noisy answer [21, 37, 19, 36, 90, 113] or other statistical metrics on measuring accuracy of the answer as functions of the selected privacy budget [116, 64]. ViP [90] associates the query output with a differentially private randomization interval, indicating the noise bounds of the query result with high confidence. This randomization bound could be either post-processed from a

noisy answer if the mechanism is data-independent or computed with a small portion of additional privacy budget from data-dependent mechanisms [21, 37, 19, 36]. DPXPlain [115], on the other hand, start to explore methods of providing additional explanation of aggregation query results, on which whether an unexpected answer is due to the data itself or the randomness introduced by DP.

Budget Specification. The composition of privacy loss through a series of queries can be classified into two levels, based on whether different analysts are distinguished. The coarser level of query composition is to regard all data analysts as a unified entity, and the system sequential composition to account for privacy loss during the execution of the queries. This method is easy to implement, well-suited to different types of (complex) queries, and can explain whether a query is rejected—the privacy budget or the translated privacy budget exceeds the remaining global privacy budget. Sequential composition can also be replaced by a privacy odometer [108, 79] or adaptive composition [127, 129, 68] that gives tighter privacy bound over the queries that are adaptively asked, but they are restricted to simple queries. More fine-grained query composition is to track the privacy loss as per data analysts and per queries they ask [138]. This approach can not only answer more specific questions on *why this particular analyst's query is rejected* but also achieve fairness among multiple data analysts when they are assigned different trust/privilege levels. However, fine-grained tracking requires recording more metadata proportional to the number of data analysts in the system.

Other Techniques for Why-DP-Provenance: Automated DP Proof Generation. Besides explaining noisy output to the data analysts, another line of work investigates the logical representation and execution of a differentially private mechanism/system. They check and verify the privacy properties of a DP program implementation by either generating automated proofs for DP [125, 2, 124, 123, 135, 93, 3] or detecting counterexamples that violate DP guarantees [28, 9, 10, 8] through annotated type systems and static/dynamic analysis of the program execution.

3.2.2 Techniques for How-DP-Provenance

Transformation Tracking. Complex database queries often consist of a series of transformations over the private data, e.g., selection, projection, group-by, join, union, and aggregation. In order to calibrate noise to the query answers, the system needs to analyze the sensitivity of the query. We categorize the approaches to estimate the sensitivity of a query into three levels, from coarse-grained to fine-grained: tracking sensitivity directly, tracking transformation stability, and tracking query structures.

Tracking Sensitivity (Directly). The coarsest level of how-DP-provenance is to directly compute an upper bound of the query sensitivity. For simple queries, the sensitivities are well understood and can be computed easily. However, for complex queries, without making clever use of the properties of the query, the sensitivity could be overestimated or cumbersome to compute.

Tracking Transformation Stability. Existing systems like PINQ [88] and wPINQ [102] keep track of the *transformation stability* to calculate the amount of noise needed for the queries. For a transformation $T : \mathcal{D} \to \mathcal{D}$, it is c-stable if $\forall$ two input databases $D, D' \in \mathcal{D}$, we have $|T(D) \triangle T(D')| \leq c \times |D \triangle D'|$, where $\triangle$ denotes the symmetric difference between two databases, i.e., $D \triangle D' = (D \backslash D') \cup (D' \backslash D)$. For example, common SQL transformations selection, projection, and counting queries have stability of 1, and group-by has a stability of 2. It has been shown that for a ϵ-differentially private mechanism $\mathcal{M}$ and a c-stable transformation series T, the composition $\mathcal{M} \circ T$ will be $(c \cdot \epsilon)$-differentially private.

Tracking Query Structures. Keeping the transformation stability metadata makes it easy to analyze the privacy loss for a series of linear transformations before histogram queries, but may not be sufficient for highly sensitive queries like aggregation and join. The query sensitivity for aggregations could be as large as the size of the domain product (e.g., sum over selected attributes). The approach to handling aggregation queries is to truncate the attribute domain [66, 45, 31] and/or perform aggregations on disjoint subsamples [95, 89, 110, 66], which requires the system to keep track of the truncation information per query and per attribute and the information about how the truncation transfers over queries. Chorus [66] enables a query rewriting and annotation technique,

which could be seen as using *how-DP-provenance*, to automatically trace and analyze the query stability and truncated domain when processing aggregation queries. Similarly, the sensitivity of join queries can be even unbounded — changing one tuple in an input table can cause unbounded changes in the join output since this tuple can match an arbitrary number of tuples in another input table. DP mechanisms for answering join queries clip the maximum number of tuples that a tuple can match in a join [75], or add data-dependent noises [95, 65, 29, 30], or a mixture of both [32]. Indeed, the recent residual sensitivity mechanism [29] analyzes the multi-way join topology metadata and traces a smooth upper bound of local sensitivity across this join topology to *efficiently* calculate a *tighter* noise.

Noise Tracking. While the basic DP system does not track the noise added to each query answer (i.e., queries are regarded as independent), a number of recent work [138, 87, 133, 81, 56, 74] inject correlated noise to the answers. They optimize the accuracy of the query results by batching the query workload and adding correlated noise with the workload of one single data analyst [132, 81, 34], or, in a more fine-grained way, maintaining a stateful cache of the historical query answers [87, 74, 75] so that answers to the new queries can reuse the cached noises. The DP caches are extended from answering one data analyst's queries to mitigating privacy loss across multiple data analysts [138, 43, 78, 61] or tight adaptive composition [119, 120] across analysts when multiple analysts ask the same or similar queries. Other multi-analyst systems [132, 133, 73, 103, 104] tracks per-analyst privacy [132] or accuracy [73] constraints to optimize the privacy-accuracy trade-off or fairly answer queries among analysts [103, 104]. Among all the different settings in multi-analyst DP, additional fine-grained requirements or privacy guarantees regarding different analysts' queries are recorded for the DP mechanism designs, which is in line with how-DP-provenance.

3.2.3 Techniques for Where-DP-Provenance

Privacy Resolutions. Depending on the data model and the intended privacy goals to deliver, a DP system can achieve the notion of *event-level DP*, *user-level DP*, multi-resolution DP (defined as per policies) [59, 75] and other extended notions of DP, e.g., personalized user-level DP [67]. Event-level DP assumes that in the private input data, each individual contributes only one record, while it is a special case of user-level DP, which allows each individual to make multiple contributions and reasons about privacy at the user level. In particular, the neighboring database definition is different in the two settings, which changes the way sensitivity is analyzed. Other DP notions, such as Pufferfish privacy [72], Blowfish privacy [59], multi-resolution privacy [75], per-attribute DP[51], Metric DP [17], geo-indistinguishability [134], etc., relax and extend DP to more general settings for example with correlations.

Privacy Accounting. Accounting for privacy loss over the input data is challenging. The basic DP system performs the privacy composition at the table level. However, it is unlikely that every query touches the entire table. Accounting privacy at the table level would waste privacy budgets for the part of data that is not used for computations. More fine-grained privacy accounting considers splitting databases into disjoint blocks and sub-tables, and only the block that is used for answering a query will be deducted for privacy consumption. This approach is called parallel composition or block composition [88, 80]. This has also been extended in recent work [78] that a finer-grained user-level partitioning is available at the column level so that the wasted privacy budget can be minimized. Other than privacy accounting for different parts of the data, the input data can arrive at different timestamps. Based on the DP models used, existing work proposes mechanisms for sliding windows [70], the entire streams [41, 35, 33] or with historical data [22].

Table 4 summarizes the techniques and the granularities of the provenances used in some existing DP systems for DP with trusted data curators (i.e., in the central setting). We discuss other cases that enforce local DP or other models of DP in Section 5.

Table 4: Taxonomy of Privacy Provenance and Evaluations of Implemented DP Systems. ✓=with, ✗=without, ○=Coarse-Grained, ◐=Moderate-Grained, ●=Fine-Grained. Shaded rows indicate systems surveyed in case studies.

Systems	Why-DP-Provenance			How-DP-Provenance		Where-DP-Provenance	
	Query Spec	Err Spec	Budget Spec	Transformation	Noise (Trackings)	Priv Resolutions	Priv Accountant
Basic	Privacy-First	✗	○ Per Query	○ Sens Only	✗ No	○ Event-DP	○ Table Level
PINQ (2009)	Privacy-First	✗	○ Per Query	◐ Sens+Stability	✗ No	○ Event-DP	◐ Sub-Table Level
ProPer (2015)	Privacy-First	✗	○ Per Query	◐ Sens+Stability	✗ No	● User-DP[†]	◐ Sub-Table Level
PSIΨ (2016)	Privacy-First	◐	◐ Per Query[*]	○ Sens Only	✗ No	○ Event-DP	○ Table Level
ϵKTELO (2018)	Privacy-First	✗	○ Per Query	◐ Sens+Stability	◐ Workload	○ Event-DP	◐ Sub-Table Level
APEx (2019)	Accuracy-First	○	○ Per Query	○ Sens Only	◐ Workload	○ Event-DP	○ Table Level
PrivateSQL (2019)	Privacy-First	✗	○ Per Query	○ Sens Only	● Caching	● Policy-DP	○ Table Level
Sage (2019)	Privacy-First	✗	○ Per Query	○ Sens Only	✗ No	○ Event-DP	◐ Sub-Table Level
Chorus (2020)	Privacy-First	✗	○ Per Query	● Query Struct	✗ No	○ Event-DP	○ Table Level
CacheDP (2022)	Accuracy-First	○	○ Per Query	○ Sens Only	● Caching	○ Event-DP	○ Table Level
ViP (2022)	Privacy-First	●	◐ Per Query[*]	○ Sens Only	✗ No	○ Event-DP	○ Table Level
DProvDB (2024)	Both	○	● Query+Analyst	◐ Sens+Stability	● Caching	○ Event-DP	○ Table Level
Cohere (2024)	Privacy-First	✗	● Query+Analyst	○ Sens Only	◐ Workload	◐ User-DP	● Column Level

[†] ProPer achieves user-level DP with personalized privacy guarantees.
[*] ViP and PSIΨ both support an interface to help analysts split privacy budget across multiple queries. ViP supports more types of error specifications (confidence intervals, quantiles, etc.).

4 Case Studies on Systems with Fine-Grained Privacy Provenance

This section dives into several representative DP systems that utilize fine-grained why-, how-, and where-provenance techniques via case studies. Each case study overviews the related systems, summarizes their provenance techniques, and discusses their strengths and weaknesses.

4.1 Case Study: APEx for Accuracy-Aware DP Data Exploration

APEx prioritizes accuracy in query specification through why-DP-provenance techniques. Additionally, it employs noise tracking for processing exploration workloads, leveraging how-DP-provenance. These features enhance the usability of the basic DP system.

4.1.1 Problem and Technical Brief

APEx is among the first systems to empower data analysts, even those without prior DP expertise, to easily specify private queries. It achieves this by allowing analysts to focus on their desired outcome, the answer's accuracy, rather than needing to grapple with complex privacy budgets. This ability to specify accuracy expectations exemplifies why-DP-provenance in action, as it provides valuable information about the usefulness and confidence intervals of query results. This feature ultimately enhances the user experience with DP systems. To achieve this user-centric approach, APEx made several vital contributions:

- A New Language for DP Queries: APEx designed a new, SQL-like query language tailored for DP tasks. This language simplifies query specification for analysts.
- Accuracy-Privacy Translation Framework: APEx developed a framework that automatically translates an analyst's desired accuracy level into an optimal DP mechanism. This framework eliminates the need for analysts to choose a mechanism themselves. APEx also compiled a comprehensive library of the latest DP mechanisms, ensuring the framework has the best one to achieve the desired accuracy-privacy trade-off.

In addition to these why-provenance advancements, APEx also introduced a novel data-dependent DP mechanism that leverages how-DP-provenance. This mechanism tracks the noise added during query processing to minimize privacy costs while still meeting the specified accuracy requirements. We will delve deeper into each of these contributions in the following sections.

Query Language with Accuracy Measures. APEx aims to support SQL-like declarative query languages with accuracy specifications. The syntax of the query language is defined in the following format.

$$\begin{aligned}&\text{BIN } D \text{ ON } f(\cdot) \text{ WHERE } W = \{\phi_1, \ldots, \phi_L\}\\ &[\text{HAVING } f(\cdot) > c]\\ &[\text{ORDER BY } f(\cdot) \text{ LIMIT } k]\\ &\text{ERROR } \alpha \text{ CONFIDENCE } 1-\beta;\end{aligned}$$

The meaning of this query syntax is to map a database D into bins of rows $\{b_1, \ldots, b_L\}$ based on a workload of predicates $W = \{\phi_1, \ldots, \phi_L\}$, where each bin b_i contains rows that satisfy the predicate ϕ_i. Then it applies the aggregation function $f()$ (e.g., count and sum) over each bin b_i. The two lines in the brackets are optional. A *workload counting query* (WCQ) is simply the query when $f()$ is a counting function that returns the bin size without the two optional lines. A similar query with the HAVING clause is an *iceberg counting query* (ICQ) that returns a list of bin identifiers b_i for which $f(b_i) > c$, and a query with the ORDER BY ... LIMIT clause is called the *top-k counting query* (TCQ) that returns the k bins that have the largest values for $f(b_i)$.

For a query with a numerical output like WCQ, there are multiple accuracy semantics considered by the literature, such as mean square error (MSE) [131], relative error [131], and (α, β)-accuracy [39]. APEx considers (α, β)-accuracy for WCQ and extends it to non-numerical queries like ICQ and TCQ. In particular, we say a mechanism M satisfies (α, β)-WCQ accuracy, if with a high probability $1-\beta$, for each predicate $\phi \in W$, the absolute difference between its noisy answer returned by M and its true answer is bounded by α, i.e.,

$$\Pr[\max_{\phi \in W} |M_\phi(D) - c_\phi(D)| \leq \alpha] \geq 1 - \beta.$$

For a non-numerical query like ICQ, APEx has two parts for its accuracy requirement:

$$\Pr[|\{\phi \in M(D) \mid c_\phi(D) < c - \alpha\}| > 0] \leq \beta$$

$$\Pr[|\{\phi \in (W - M(D)) \mid c_\phi(D) > c + \alpha\}| > 0] \leq \beta$$

simultaneously. This first part corresponds to the type I error that wrongly labels predicates with a true count less than c as $> c$. The second part is for the type II error that labels the predicates with a true count greater than c as $< c$. The accuracy definition is satisfied if, with a high probability of $1-\beta$, all predicates with true counts greater than $c + \alpha$ or less than $c - \alpha$ are correctly labelled. APEx extends the same logic to the accuracy requirement for TCQ in a similar flavor. Note that both (α, β)-ICQ accuracy and (α, β)-TCQ accuracy definitions consider *symmetric* errors. Definitions and mechanisms that extend to *asymmetric* errors are discussed and proposed in the MIDE system for private decision-making [50].

Accuracy-Privacy Translation Framework. For each query and its accuracy specification, APEx automatically finds the best mechanism that satisfies the accuracy specification of the query with the least privacy budget. First, APEx prepares all the existing DP mechanisms and executes them in two phases: (i) privacy cost estimation, which simulates how much privacy budget would be needed if the mechanism were used, and (ii) running the algorithm, which actually runs the mechanism and returns the answer to the data analysts. Second, APEx stores all the relevant state-of-the-art DP mechanisms for each query type since the best one depends on the query and the data. For example, for WCQ, APEx provides two data-independent translation mechanisms. The first data-independent mechanism is the baseline *Laplace mechanism*, which analyzes the error bounds of Laplace noise and obtains directly a closed-form expression of bounds on the translated privacy budgets. The second mechanism, *strategy-based mechanism*, uses the matrix mechanism [81, 82] to analyze the overlapping parts of the

workload W and reuse the noisy intermediate results for the overlapping part to translate the same accuracy target into, in many cases, a tighter bound on privacy budget. The strategy-based mechanism is also data-independent and can be generalized to all the query types.

Data Dependent Accuracy-Privacy Translation. APEx considers a data-dependent accuracy-privacy translation mechanism for ICQ, an example of how-DP-provenance. An ICQ involves comparing the bin sizes with the threshold value c. For example, the data-dependent Laplace mechanism adds noise to the true bin sizes and compares the noisy bin sizes with the threshold. For example, given an accuracy target $\alpha = 10, \beta = 0.00001$, APEx will translate this into budget $\frac{\ln(1/2\beta)}{\alpha} = 0.85$ for the Laplace mechanism. However, if the bin sizes are far away from the threshold, let's say $c_\phi(D) = 1000$ and the threshold is $c = 100$ where the difference is 90 times α, then it may be sufficient only to use $\frac{0.85}{90} \approx 0.01$ privacy budget. As APEx does not know the distance between the true bin sizes and the threshold, it proposes a *multi-poking mechanism* that starts with a small privacy budget and tests multiple times with increasing privacy budgets until it can determine the noisy answer would satisfy the accuracy target confidently. Rather than drawing independent Laplace noise in each iteration, this multi-poking mechanism stores the privacy budgets and the noise used in previous "pokings" and samples correlated noise each time [77]. This correlated noise allows the overall privacy loss to be bounded by the privacy budget of the last poking instead of the sum of the privacy budgets over all the poking steps.

4.1.2 Improvements and Limitations

APEx offers several advantages that make it easier to conduct private data analysis: 1) *usability improvement via why-DP-provenance*, where APEx empowers data analysts, even those without prior privacy expertise, to achieve high accuracy in tasks like entity resolution, as shown in its user studies; 2) privacy improvement via how-DP provenance, where the new multi-poking mechanism significantly reduces the privacy budget needed for specific workloads (4 out of 8 ICQ workloads in studies) compared to traditional approaches. While APEx offers significant benefits, it is essential to acknowledge some limitations: 1) *compatibility with accuracy bounds*, for not all data privacy algorithms have clearly defined accuracy limitations, which restricts APEx's ability to incorporate them into its framework; 2) *runtime overheads*, for APEx requires running all the relevant mechanisms and storing/tracing the correlated noise calibration in the multi-poking mechanism, though they are relatively small and are only needed at runtime.

4.2 Case Study: DProvDB for Multi-Analyst DP

Unlike prior systems, DProvDB enforces privacy constraints, not only on the queries and the input data but also on each analyst. This design aims to achieve a fair distribution of privacy budget among data analysts and a tight privacy control per data analyst. DProvDB also leverages multiple how-DP provenance techniques for its DP mechanisms, including tracking sensitivity and stability of queries like Chorus [66] and tracking noise to responses for different analysts, considering how these responses might be interrelated over time. In this case study, we will highlight the privacy constraints for why-DP-provenance and the new noise-tracking technique for how-DP-provenance.

4.2.1 Problem and Technical Brief

DProvDB considers the problem of building an online query processing system for multiple data analysts, who are regulated not to collude but may break the regulation and collude. These data analysts also have different trust/privilege levels when accessing the data; for example, internal analysts shall use more global privacy budgets than external data analysts. DP systems before DProvDB do not distinguish data analysts, and naively tracing each analyst's queries independent of others can waste the global budget — if collusion happens, the privacy loss

across data analysts is upper bounded by $\sum \epsilon_i, \sum \delta_i$ while it is lower bounded by $\max \epsilon_i, \max \delta_i$, where ϵ_i, δ_i is the privacy budget spent on each data analyst.

Unlike the basic DP system described in Section 3.1, DProvDB might reject an analyst's query if answering it would exhaust either the analyst's individual privacy budget or the total budget shared by all analysts. This ensures fair and controlled use of privacy resources. To enforce these privacy constraints (why-DP-provenance), DProvDB utilizes two key components:

- Privacy Provenance Table: This table tracks past queries and the privacy budget spent on each. It allows DProvDB to monitor individual and overall budget consumption.
- Custom DP Mechanisms: DProvDB designs specialized DP mechanisms for this multi-analyst environment. These mechanisms consider budget limitations when determining whether to answer a query.

Furthermore, DProvDB tackles minimizing the overall privacy loss even in scenarios where analysts might collaborate (collusion). Here, DProvDB leverages a technique based on the additive Gaussian mechanism. This technique reuses previously generated noisy outputs (how-DP-provenance) to answer new queries from other analysts to achieve the lower bound for the privacy loss at collusion while still providing useful results.

Next, we will introduce the building block DP mechanism in DProvDB that achieves the lower privacy bound when all analysts collude for a simple query. We will then present how it is used for online query processing and the necessary provenance information for its deployment in DProvDB.

Building Block: Additive Gaussian Mechanism. Consider two analysts A_1 and A_2 send the same query q with two different privacy budgets (ϵ_1, δ) and (ϵ_2, δ), respectively (W.L.O.G, assuming $\epsilon_1 > \epsilon_2$). If responding to each analyst separately with an independent Gaussian mechanism (Definition 2.1), i.e., $q(D) + \eta_1$ for A_1 and $q(D) + \eta_2$ for A_2, where $\eta_1 \sim \mathcal{N}(0, \sigma_1^2 I)$ for (ϵ_1, δ)-DP and $\eta_2 \sim \mathcal{N}(0, \sigma_2^2 I)$ for (ϵ_2, δ)-DP, then the overall privacy loss if these two analysts collude will be $(\epsilon_1 + \epsilon_2, 2\delta)$.

The additive Gaussian mechanism first processes the noisy response to A_1 with the standard Gaussian mechanism, $q(D) + \eta_1$ from the above distribution. Then, it reuses η_1 in its response to A_2 by returning $q(D) + \eta_1 + \eta'$, where $\eta_1' \sim \mathcal{N}(0, \sigma_2^2 - \sigma_1^2)$. If A_1 and A_2 do not collude, the privacy loss to each one of them is (ϵ_1, δ)-DP and (ϵ_2, δ)-DP respectively. However, if they collude, the overall privacy loss is bounded by (ϵ_1, δ)-DP as the most accurate response they can come up with is the noisy response to A_1.

Note that the additive Gaussian mechanism described above only works for *the identical queries* when the *privacy budget of the first processed query is always greater than that of the future queries*. This limitation poses challenges in online database systems when 1) two analysts' queries *only overlaps* (i.e., not exactly the same), and 2) a query received at a *later timestamp* has a larger privacy budget compared to the processed historical query. To address these challenges, DProvDB devised the additive Gaussian mechanism by carefully selecting, maintaining, and updating a set of historical query answers to different data analysts and their respective privacy consumption over time.

Query Answering using Views/Synopses. To solve the first problem of overlapping queries, DProvDB does not directly apply the additive Gaussian mechanism to the queries from the data analysts. Instead, it creates materialized private views or synopses of the data using the additive Gaussian mechanism and post-processes queries on these synopses. These views are essentially histograms (or contingency tables for multiple columns). They capture the distribution of data for specific attributes and allow for processing queries that involve linear combinations of the data points (like finding averages or sums). Synopses are formed by adding different noises to the true answer of each view, and post-processing these noisy synopses does not consume an additional privacy budget. Hence, even if queries from analysts partially overlap or differ entirely, as long as they can be processed using the same view, DProvDB will update the corresponding synopses for this view with the additive Gaussian mechanism and use the updated synopses to answer the queries.

Incremental Synopses Maintenance. To tackle the second problem on dynamic budget, DProvDB maintains the noisy synopses *adaptively* based on incoming queries submitted to the system. DProvDB, in particular, has

two layers of synopses: 1) a *global synopsis* per view, and 2) a *local synopsis* per view and per analyst. The local synopsis is always generated from the global synopsis using the additive Gaussian mechanism, and the analyst's queries are always processed on their corresponding local synopses (viz., post-processing). Therefore, privacy loss across data analysts is always bounded by the privacy budget used for generating global synopses.

To answer queries with a higher privacy budget (i.e., the analyst wants the query answer to be more accurate) while reusing existing synopses, DProvDB updates the global synopses using the following approach. When the global DP synopsis V^{ϵ} does not provide enough accuracy to handle a local synopsis request at privacy budget ϵ_t, DProvDB spends additional privacy budget $\Delta\epsilon$ to update the global DP synopsis to $V^{\epsilon+\Delta\epsilon}$, where $\Delta\epsilon = \epsilon_t - \epsilon$. Here, DProvDB uses the standard Gaussian mechanism, which generates an intermediate DP synopsis $V^{\Delta\epsilon}$ with a budget $\Delta\epsilon$, and then combines the previous synopses with this intermediate synopsis into an updated one. The key insight of the combination is to properly involve the fresh noisy synopses by assigning each synopsis with a weight proportional to the inverse of its noise variance, which gives the smallest expected square error based on UMVUE [71, 105]. That is, for the t-th release, we combine these two synopses $V^{\epsilon_t} = (1 - w_t)V^{\epsilon_{t-1}} + w_t V^{\Delta\epsilon}$. The resulted expected square error for V^{ϵ_t} is $v_t = (1 - w_t)^2 v_{t-1} + w_t^2 v_{\Delta}$, where v_{t-1} is the noise variance of view $V^{\epsilon_{t-1}}$, and v_{Δ} is derived from $V^{\Delta\epsilon}$. The error is minimized at $w_t = \frac{v_{t-1}}{v_{\Delta}+v_{t-1}}$.

Privacy Provenance Table. Besides maintaining the global and local synopses, DProvDB keeps a privacy provenance table to manage the privacy budgets. The privacy provenance table $\mathcal{P}$ consists of (i) a provenance matrix P that tracks the privacy loss of a view in $\mathcal{V}$ to each data analyst in $\mathcal{A}$, where each entry of the matrix $P[A_i, V_j]$ records the current cumulative privacy loss $S_{V_j}^{A_i}$, on view V_j to analyst A_i; (ii) a set of row/column/table constraints, Ψ: a row constraint for i-th row of P, denoted by ψ_{A_i}, refers to the allowed maximum privacy loss to a data analyst $A_i \in \mathcal{A}$ (according to his/her privilege level); a column constraint for the j-th column, denoted by ψ_{V_j} refers to as the allowed maximum privacy loss to a specific view V_j; the table constraint over P, denoted by ψ_P, specifies the overall privacy loss allowed for the protected database. Due to the privacy constraints imposed by the privacy provenance table, queries can be rejected when the cumulative privacy cost exceeds the constraints. The overall privacy guarantee of the system is then implied by the three levels of privacy constraints over the provenance table. Given the privacy provenance table and its constraint specifications, $\Psi = \{\psi_{A_i} | A_i \in \mathcal{A}\} \cup \{\psi_{V_j} | V_j \in \mathcal{V}\} \cup \{\psi_P\}$, DProvDB ensures $[\ldots, (A_i, \psi_{A_i}, \delta), \ldots]$-multi-analyst-DP; it also ensures $\min(\psi_{V_j}, \psi_P)$-DP for view $V_j \in \mathcal{V}$ and overall ψ_P-DP if all the data analysts collude.

4.2.2 Improvements and Limitations

DProvDB is built as a middleware or a multi-analyst interface that works on top of the existing Chorus system [66]. This allows DProvDB to leverage Chorus's functionalities while adding its own capabilities. Experiments of DProvDB are tested over the Adult census dataset [38] and the TPC-H synthetic dataset [20] with two types of workloads, one of which consists of randomized range queries over random attributes while the other simulates traversing a decomposition tree of the domain of selected attributes. With more than one data analyst in the experimental setup, empirical results show that DProvDB dominates existing DP query processing systems by answering 2.5x-1000x more queries given the same privacy budget.

One current limitation of DProvDB is the overhead associated with storing, querying, and updating the privacy provenance information (privacy provenance table). Future work will focus on optimizing these operations for better efficiency. Interestingly, the way DProvDB updates synopses based on analyst queries is similar to the problem of *incremental view maintenance* from the field of data provenance. While the current algorithm in DProvDB does not modify the actual queries, there is potential to explore how incremental view maintenance techniques could inspire new and more efficient algorithms for private data management. There are also several interesting future directions related to privacy provenance for multi-analyst DP. For analyst provenance tracking, research questions and works may be spawned by a deeper intertwinement between privacy provenance and access/leakage control [98] or focus on a more expressive model for privacy provenance. For example, an analyst may temporarily delegate his/her privacy privilege to other analysts.

4.3 Case Study: User-Level Adaptive Block Composition in Sage and Cohere

Systems like Sage [80] and Cohere [78] aim to build a DP system that can continuously run with a finite global budget. To achieve this, Sage and Cohere, different from the basic DP system, enable more fine-grained privacy accounting (i.e., where-DP-provenance) at the level of subsets (i.e., blocks) of the data and replace the retired data blocks with new data. In addition, Cohere achieves user-level DP (privacy resolutions in where-DP-provenance) and enables budget allocation optimization over a batch of applications/queries (i.e., features how-DP-provenance).

4.3.1 Problem and Technical Brief

Sage and Cohere study the approaches to building DP systems that can continuously run with a finite global budget. They explore the heterogeneous input data streams and the parallel/block composition techniques that account for privacy loss over disjoint subsets of data.

Block Composition in Sage [80]. Block composition is an extension of parallel composition to the streaming data model. To apply block composition, at each timestamp, the system will create a new data block with disjoint sets of data from the previous blocks and also maintain the state of the block with a privacy filter [108]. Note that the creation/split of the block is based on some publicly known specifications while the data blocks after splitting is remaining secret. The specifications are criteria of how data is split over public domain values of certain attributes, e.g., timestamp, userID, geography. At query time, the system allows to run DP queries adaptively over the overlapping subsets of the blocks created so far. The privacy accounting is performed by updating the privacy filter per block, which is intuitively enforcing adaptive sequential composition (or other tighter composition bounds [79]) within a block and parallel composition across blocks (if a query is answered using multiple blocks)[3]. Sage applies block composition over time splits and hence guarantees event-level DP. The subsequent work, Cohere [78], extends the methodology of block composition and applies to user-level DP.

Partitioning Attributes and User Rotation in Cohere [78]. Cohere creates new blocks based on split over both userIDs and a (set of) given attribute(s). That is, each block generated contains a new batch of users that never appear in the previous blocks and a value in the selected partitioning attributes. For example, data block 1 contains users 1-3 all with region A, and data block 2 has users 4-6 all with region B, and block 3 consists of users 7-9 with region A, etc. Cohere then applies block composition over the user data blocks and retires users with replacement of new users to keep the system continuously running. Cohere also adopts a user rotation mechanism to prevent users from retiring too quickly from certain subpopulations (in terms of some values of the partitioning attributes), which reduces biases in answering queries or running applications. The approach is based on sliding windows (or, in essence, the least recently used strategy) so that it is independent of the user attributes. In particular, at each timestamp, the newly joined users are randomly partitioned/assigned into groups, and the group that was active the longest will be retired (tentatively if the budget over this group is not depleted), and a new group will be activated. The budget spent by queries at this timestamp will be capped with $1/K$ of the global budget where K is the number of active groups.

Formalizing Optimization Problem in Cohere [78]. Cohere further uses the tracked budget allocation history per block (or where-DP-provenance) to develop an optimization problem, as a variant of the multidimensional knapsack problem, for query answering. Each query R_i in the Cohere system is annotated with a propositional formula Φ_i over the partitioning attributes, a privacy budget $\mathbf{C}_i \in \mathbb{R}_{\geq 0}^{|\mathcal{A}|}$, and a weight $W_i \in \mathbb{N}$. The goal of Cohere's query processing is to batch all the queries $\mathcal{R}$ received at the current timestamp, and find an optimal (in terms of the weights) set of queries to answer while subject to privacy constraints due to partitioning attributes. To formulate the optimization problem, Cohere introduces decision variables $y_i \in \{0, 1\}$ for $i \in \mathcal{R}$, where $y_i = 1$ means the request R_i is accepted, and $y_i = 0$ means the request has been rejected. The privacy constraints are defined over blocks $\mathcal{S}$. A block $S_j \in \mathcal{S}$ is denoted by $(groupid, \Psi_j, \mathbf{B}_j)$, where Ψ_j is a propositional formula

[3] By using privacy filter, Sage can support strong composition [42, 69] with block composition.

over the partitioning attributes and $\mathbf{B}_j \in \mathbb{R}^A_{\geq 0}$ is the remaining budget for this block. Given the demand for a block, written as $d_{ij} = \mathbf{C}_i$ if $\Phi_i \wedge \Psi_j$ is satisfiable, and 0 otherwise, the optimization problem is formalized as $\max \sum_{i \in \mathcal{R}} y_i \cdot W_i$, s.t. $\sum_{i \in \mathcal{R}} d_{ij} y_i \leq B_j$, $[\forall j \in \mathcal{S}]$. Cohere generalizes the problem into an integer linear programming (ILP) problem and solves it with an ILP solver. Cohere also shows an optimization technique to reduce the dimensionalities to scale the solving process.

4.3.2 Improvements and Limitations

Sage and Cohere report the benefit of enabling block composition and recording per block budget consumption (viz., where-DP-provenance) through extensive experimental evaluations. Sage shows that with block composition, the system can train a machine learning model with a lower mean squared error (MSE) that cannot be achieved by sequential composition (Δ=0.0002); to achieve the same MSE, block composition requires much less data than sequential composition (10x-100x less in certain cases). On the other hand, Cohere is compared with PrivateKube [84], a privacy budget scheduler built based on Sage. In an end-to-end comparison with PrivateKube, Cohere shows an improvement in handling 1.5x-2.0x more queries and a 6.4x–28x better utility due to the partitioning attributes approach and the more fine-grained privacy analysis. All these experimental results on Sage and Cohere validate the effectiveness of enabling where-DP-provenance in a DP system.

However, Cohere cannot be run in real-time systems since maintaining the block composition and solving the optimization problem requires, on average, 48 minutes and around 1.3 GB of memory. Empirical results also observe an increasing runtime when using the partitioning attribute approach. Another limitation of using block composition or partitioning attributes with where-DP-provenance is that they assume the criteria for creating the partitioning is publicly known; otherwise, it will cause problems in the privacy analysis. Removing the assumption may be a future direction to explore for where-DP-provenance.

5 Discussion and Open Questions

This section discusses the fine-grained provenance for DP systems related to traditional data provenance literature and the development of DP with their respective open questions.

5.1 Relationship with Data Provenance

The well-established field of data provenance offers valuable insights for the future development of DP provenance. In this work, we explored why, how, and where provenance for DP systems. To delve deeper, let's discuss two additional areas for consideration.

Representations of Privacy Provenance. In traditional databases, provenance information can be represented using either an eager or a lazy approach. The eager approach [15] attaches extra metadata (annotations) to queries and propagates it to the results, e.g., based on the provenance semirings [54] or calculation of Shapley value [85, 86]. While this allows for direct retrieval of provenance information, it can incur performance overhead and require additional storage for the metadata. The lazy approach [18] relies on properties of specific transformations to identify the source data behind the output without annotations. This method has lower overhead but limited applicability. In the context of DP provenance, we have primarily focused on leveraging and storing additional metadata, similar to the eager approach. It would be interesting to explore the feasibility of a lazy approach for DP provenance. This could involve developing mechanisms to answer why-, how-, and where-provenance queries without needing constant metadata storage and updates.

Scalable Privacy Provenance Tracking. Research in database provenance has addressed the challenge of scalability in managing provenance information [49, 112, 16, 106]. These methods aim to reduce the cost of tracking provenance by either minimizing the amount of extra metadata required or employing compression techniques to approximate provenance. For DP provenance, the level of granularity (detail) directly impacts

the amount of data that needs to be tracked. Finer-grained provenance necessitates tracking more data. There are two key areas for further exploration. First, we would like to have a better understanding of the trade-off between privacy granularity and the associated storage and processing overhead. For instance, DProvDB can use a finer-grained caching mechanism like CacheDP [87] to further save privacy budget per query, but maintaining and updating such cache structures for all analysts can be very expensive. This will guide future research efforts. Second, developing techniques to reduce the cost of privacy provenance tracking is a promising research direction. Existing systems, like DProvDB, which tracks analyst provenance at the view level for efficiency, offer valuable insights. Future work could explore compression and approximation techniques specifically tailored for efficient privacy provenance management.

5.2 Relationship with the Development of DP

Differential privacy has been around since 2006 and has evolved significantly. Today, we have a vast array of DP algorithms for various uses, different privacy definitions for various scenarios, and even prototype programming tools and systems. This article focuses on DP systems that leverage provenance techniques to improve usability or performance. The effectiveness of these provenance techniques is directly related to the development of DP algorithms and definitions. Let's explore some key areas for further exploration.

Optimal Algorithm Design. Accuracy-first mechanisms are less understood compared to privacy-first mechanisms. For accuracy-first mechanisms, how queries translate into privacy guarantees can vary depending on the specific queries, how accuracy is measured by the system, and even the data (e.g., joining tables or measuring relative error). Existing research for privacy-first mechanisms has produced optimal solutions for specific queries like joins [32, 30, 29, 131]. However, there is a gap in understanding how to achieve optimal results for accuracy-first mechanisms, particularly those that depend on the data. Closing this gap is crucial for developing user-friendly DP systems that leverage why-DP-provenance.

Recent advancements in DP mechanisms [87, 48, 100, 126] involve using correlated noise drawn from different points over time. This approach can lead to tighter privacy analysis or improved utility for the results. However, effectively and securely maintaining these noise sequences is critical for successful deployment. This necessitates the development of systematic how-DP-provenance techniques in the future.

Limited DP algorithms have been developed specifically for growing data models, and they often overlook how data evolves over time (e.g., their temporal properties). In machine learning, for instance, data patterns and learned models can change over time (i.e., the concept drift). Factoring in concept drift will likely require even finer-grained how/where-DP-provenance tracking for effective solutions.

Mixing DP Variants. DP provides some degree of freedom to allow system designers to "composite" privacy guarantees. However, a key challenge arises if one part of the data is released with DP while others are queried and processed with other privacy notions, such as OSDP [76], attribute privacy [140], pufferfish privacy [72], etc., each with its own strengths and use cases. While combining these use cases into a unified system might be desirable, a significant question remains: how do we account for the total privacy loss when mixing different privacy-preserving techniques? Recent research in encrypted databases [139] explores similar challenges in reasoning about security when combining multiple encryption techniques. This offers valuable insights for the DP domain. An interesting future direction would be to develop techniques specifically for mixing different privacy definitions and calculating the resulting privacy loss. One possibility is to leverage existing DP auditing techniques [101, 62, 91]. These techniques can help us establish a lower bound for the overall privacy guarantee, even when combining different privacy-preserving methods.

Removing Trusted Curators in DP. Our discussion of privacy provenance has so far focused on centralized DP systems, which rely on a trusted curator to oversee the entire process. However, this centralized approach may not be practical in all real-world scenarios. Removing the need for a trusted party is an active area of research with several promising directions. First, local DP empowers data owners to add noise to their own data before it is used in queries. While this offers greater privacy control, it can lead to lower accuracy than centralized DP.

Existing research has explored using anonymous shufflers [11, 52] to improve utility in local DP settings. An interesting future direction would be to investigate how other privacy provenance information, besides shuffling, can be leveraged to enhance utility in local DP systems. Second, combining DP with cryptography [122, 109] offers another approach to reduce the noise needed for the local or federated settings [13, 7]. Third, enabling a DP system with trusted hardware, e.g., SGX [94], can simulate the trusted curator in the untrusted settings. However, as shown in recent work [63], the last two approaches have to maintain and track a significant amount of additional metadata about the state of the running environment. Without this metadata tracking, the system remains vulnerable. Privacy provenance can be crucial in future work on these decentralized privacy-preserving techniques. Providing a systematic view of data usage and privacy guarantees can help address the challenges associated with removing the need for a trusted central authority in DP systems.

6 Conclusion

This article explores how provenance techniques can empower differential privacy (DP) systems. We introduce a novel taxonomy for three crucial DP provenance types (why-provenance, how-provenance, and where-provenance). We then unpack existing techniques for each type, leveraging case studies to illuminate their advantages and drawbacks. Finally, we establish the link between fine-grained DP provenance and traditional data provenance, investigating how both can propel advancements in DP across various domains. This work is the first attempt to bridge these two critical areas. We hope it offers a unique perspective and paves the way for further research in this exciting direction.

References

[1] J. M. Abowd. The us census bureau adopts differential privacy. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pages 2867–2867, 2018.

[2] C. Abuah, A. Silence, D. Darais, and J. P. Near. Dduo: General-purpose dynamic analysis for differential privacy. In 2021 IEEE 34th Computer Security Foundations Symposium (CSF), pages 1–15. IEEE, 2021.

[3] C. Abuah, D. Darais, and J. P. Near. Solo: A lightweight static analysis for differential privacy. Proc. ACM Program. Lang., 6(OOPSLA2), oct 2022. doi: 10.1145/3563313. URL https://doi.org/10.1145/3563313.

[4] Amazon Inc. AWS Clean Rooms differential privacy. https://aws.amazon.com/clean-rooms/differential-privacy/. Accessed: 2024-05-31.

[5] K. Amin, J. Gillenwater, M. Joseph, A. Kulesza, and S. Vassilvitskii. Plume: Differential privacy at scale. CoRR, abs/2201.11603, 2022. URL https://arxiv.org/abs/2201.11603.

[6] B. Balle and Y. Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In J. G. Dy and A. Krause, editors, Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of Proceedings of Machine Learning Research, pages 403–412. PMLR, 2018. URL http://proceedings.mlr.press/v80/balle18a.html.

[7] E. Bao, Y. Zhu, X. Xiao, Y. Yang, B. C. Ooi, B. H. M. Tan, and K. M. M. Aung. Skellam mixture mechanism: a novel approach to federated learning with differential privacy. Proc. VLDB Endow., 15(11): 2348–2360, 2022. URL https://www.vldb.org/pvldb/vol15/p2348-bao.pdf.

[8] G. Barthe, R. Chadha, V. Jagannath, A. P. Sistla, and M. Viswanathan. Deciding differential privacy for programs with finite inputs and outputs. In Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science, pages 141–154, 2020.

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

[10] B. Bichsel, S. Steffen, I. Bogunovic, and M. Vechev. Dp-sniper: Black-box discovery of differential privacy violations using classifiers. In 2021 IEEE Symposium on Security and Privacy (SP), pages 391–409. IEEE, 2021.

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

[12] D. M. Bittner, A. E. Brito, M. Ghassemi, S. Rane, A. D. Sarwate, and R. N. Wright. Understanding privacy-utility tradeoffs in differentially private online active learning. Journal of Privacy and Confidentiality, 10 (2), 2020.

[13] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical secure aggregation for privacy-preserving machine learning. In proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pages 1175–1191, 2017.

[14] P. Buneman, S. Khanna, and W. C. Tan. Why and where: A characterization of data provenance. In J. V. den Bussche and V. Vianu, editors, Database Theory - ICDT 2001, 8th International Conference, London, UK, January 4-6, 2001, Proceedings, volume 1973 of Lecture Notes in Computer Science, pages 316–330. Springer, 2001. doi: 10.1007/3-540-44503-X_20. URL https://doi.org/10.1007/3-540-44503-X_20.

[15] P. Buneman, S. Khanna, and W. C. Tan. On propagation of deletions and annotations through views. In L. Popa, S. Abiteboul, and P. G. Kolaitis, editors, Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 3-5, Madison, Wisconsin, USA, pages 150–158. ACM, 2002. doi: 10.1145/543613.543633. URL https://doi.org/10.1145/543613.543633.

[16] A. P. Chapman, H. V. Jagadish, and P. Ramanan. Efficient provenance storage. In Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pages 993–1006, 2008.

[17] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi. Broadening the scope of differential privacy using metrics. In E. D. Cristofaro and M. K. Wright, editors, Privacy Enhancing Technologies - 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013. Proceedings, volume 7981 of Lecture Notes in Computer Science, pages 82–102. Springer, 2013. doi: 10.1007/978-3-642-39077-7_5. URL https://doi.org/10.1007/978-3-642-39077-7_5.

[18] J. Cheney, L. Chiticariu, and W. C. Tan. Provenance in databases: Why, how, and where. Found. Trends Databases, 1(4):379–474, 2009. doi: 10.1561/1900000006. URL https://doi.org/10.1561/1900000006.

[19] E. Cohen, X. Lyu, J. Nelson, T. Sarlós, and U. Stemmer. Optimal differentially private learning of thresholds and quasi-concave optimization. In Proceedings of the 55th Annual ACM Symposium on Theory of Computing, pages 472–482, 2023.

[20] T. T. P. P. Council. The tpc benchmark h (tpc-h)., 2008. URL https://www.tpc.org/tpch/.

[21] C. Covington, X. He, J. Honaker, and G. Kamath. Unbiased statistical estimation and valid confidence intervals under differential privacy. Statistica Sinica, to appear.

[22] R. Cummings, S. Krehbiel, K. A. Lai, and U. Tantipongpipat. Differential privacy for growing databases. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18, page 8878–8887, Red Hook, NY, USA, 2018. Curran Associates Inc.

[23] R. Cummings, D. Desfontaines, D. Evans, R. Geambasu, Y. Huang, M. Jagielski, P. Kairouz, G. Kamath, S. Oh, O. Ohrimenko, N. Papernot, R. Rogers, M. Shen, S. Song, W. Su, A. Terzis, A. Thakurta, S. Vassilvitskii, Y.-X. Wang, L. Xiong, S. Yekhanin, D. Yu, H. Zhang, and W. Zhang. Advancing Differential Privacy: Where We Are Now and Future Directions for Real-World Deployment. Harvard Data Science Review, 6(1), jan 16 2024. https://hdsr.mitpress.mit.edu/pub/sl9we8gh.

[24] S. B. Davidson, S. Khanna, T. Milo, D. Panigrahi, and S. Roy. Provenance views for module privacy. In M. Lenzerini and T. Schwentick, editors, Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2011, June 12-16, 2011, Athens, Greece, pages 175–186. ACM, 2011. doi: 10.1145/1989284.1989305. URL https://doi.org/10.1145/1989284.1989305.

[25] S. B. Davidson, S. Khanna, S. Roy, J. Stoyanovich, V. Tannen, and Y. Chen. On provenance and privacy. In T. Milo, editor, Database Theory - ICDT 2011, 14th International Conference, Uppsala, Sweden, March 21-24, 2011, Proceedings, pages 3–10. ACM, 2011. doi: 10.1145/1938551.1938554. URL https://doi.org/10.1145/1938551.1938554.

[26] D. Deutch, A. Frankenthal, A. Gilad, and Y. Moskovitch. On optimizing the trade-off between privacy and utility in data provenance. In G. Li, Z. Li, S. Idreos, and D. Srivastava, editors, SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021, pages 379–391. ACM, 2021. doi: 10.1145/3448016.3452835. URL https://doi.org/10.1145/3448016.3452835.

[27] B. Ding, J. Kulkarni, and S. Yekhanin. Collecting telemetry data privately. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 3571–3580, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/253614bbac999b38b5b60cae531c4969-Abstract.html.

[28] Z. Ding, Y. Wang, G. Wang, D. Zhang, and D. Kifer. Detecting violations of differential privacy. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pages 475–489, 2018.

[29] W. Dong and K. Yi. Residual sensitivity for differentially private multi-way joins. In G. Li, Z. Li, S. Idreos, and D. Srivastava, editors, SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021, pages 432–444. ACM, 2021. doi: 10.1145/3448016.3452813. URL https://doi.org/10.1145/3448016.3452813.

[30] W. Dong and K. Yi. A nearly instance-optimal differentially private mechanism for conjunctive queries. In L. Libkin and P. Barceló, editors, PODS '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022, pages 213–225. ACM, 2022. doi: 10.1145/3517804.3524143. URL https://doi.org/10.1145/3517804.3524143.

[31] W. Dong and K. Yi. Universal private estimators. In Proceedings of the 42nd ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, pages 195–206, 2023.

[32] W. Dong, J. Fang, K. Yi, Y. Tao, and A. Machanavajjhala. R2T: instance-optimal truncation for differentially private query evaluation with foreign keys. In Z. G. Ives, A. Bonifati, and A. E. Abbadi, editors, SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022, pages 759–772. ACM, 2022. doi: 10.1145/3514221.3517844. URL https://doi.org/10.1145/3514221.3517844.

[33] W. Dong, Q. Luo, and K. Yi. Continual observation under user-level differential privacy. In 44th IEEE Symposium on Security and Privacy, SP 2023, San Francisco, CA, USA, May 21-25, 2023, pages 2190–2207. IEEE, 2023. doi: 10.1109/SP46215.2023.10179466. URL https://doi.org/10.1109/SP46215.2023.10179466.

[34] W. Dong, D. Sun, and K. Yi. Better than composition: How to answer multiple relational queries under differential privacy. Proc. ACM Manag. Data, 1(2):123:1–123:26, 2023. doi: 10.1145/3589268. URL https://doi.org/10.1145/3589268.

[35] W. Dong, Z. Chen, Q. Luo, E. Shi, and K. Yi. Continual observation of joins under differential privacy. Proceedings of the ACM on Management of Data, 2(3):1–27, 2024.

[36] J. Drechsler, I. Globus-Harris, A. Mcmillan, J. Sarathy, and A. Smith. Nonparametric differentially private confidence intervals for the median. Journal of Survey Statistics and Methodology, 10(3):804–829, 2022.

[37] W. Du, C. Foot, M. Moniot, A. Bray, and A. Groce. Differentially private confidence intervals. arXiv preprint arXiv:2001.02285, 2020.

[38] D. Dua and C. Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[39] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci., 9(3-4):211–407, 2014. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

[40] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, editors, Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings, volume 3876 of Lecture Notes in Computer Science, pages 265–284. Springer, 2006. doi: 10.1007/11681878_14. URL https://doi.org/10.1007/11681878_14.

[41] C. Dwork, M. Naor, T. Pitassi, G. N. Rothblum, and S. Yekhanin. Pan-private streaming algorithms. In A. C. Yao, editor, Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings, pages 66–80. Tsinghua University Press, 2010. URL http://conference.iiis.tsinghua.edu.cn/ICS2010/content/papers/6.html.

[42] C. Dwork, G. N. Rothblum, and S. P. Vadhan. Boosting and differential privacy. In 51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA, pages 51–60. IEEE Computer Society, 2010. doi: 10.1109/FOCS.2010.12. URL https://doi.org/10.1109/FOCS.2010.12.

[43] C. Dwork, M. Naor, and S. P. Vadhan. The privacy of the analyst and the power of the state. In 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA,

October 20-23, 2012, pages 400–409. IEEE Computer Society, 2012. doi: 10.1109/FOCS.2012.87. URL https://doi.org/10.1109/FOCS.2012.87.

[44] H. Ebadi, D. Sands, and G. Schneider. Differential privacy: Now it's getting personal. In S. K. Rajamani and D. Walker, editors, Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015, Mumbai, India, January 15-17, 2015, pages 69–81. ACM, 2015. doi: 10.1145/2676726.2677005. URL https://doi.org/10.1145/2676726.2677005.

[45] J. Fang, W. Dong, and K. Yi. Shifted inverse: A general mechanism for monotonic functions under user differential privacy. In H. Yin, A. Stavrou, C. Cremers, and E. Shi, editors, Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022, pages 1009–1022. ACM, 2022. doi: 10.1145/3548606.3560567. URL https://doi.org/10.1145/3548606.3560567.

[46] M. Gaboardi, J. Honaker, G. King, J. Murtagh, K. Nissim, J. Ullman, and S. Vadhan. Psi ({\Psi}): a private data sharing interface. arXiv preprint arXiv:1609.04340, 2016.

[47] A. Gadotti, F. Houssiau, M. S. M. S. Annamalai, and Y. de Montjoye. Pool inference attacks on local differential privacy: Quantifying the privacy guarantees of apple's count mean sketch in practice. In K. R. B. Butler and K. Thomas, editors, 31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022, pages 501–518. USENIX Association, 2022. URL https://www.usenix.org/conference/usenixsecurity22/presentation/gadotti.

[48] C. Ge, X. He, I. F. Ilyas, and A. Machanavajjhala. Apex: Accuracy-aware differentially private data exploration. In P. A. Boncz, S. Manegold, A. Ailamaki, A. Deshpande, and T. Kraska, editors, Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019, pages 177–194. ACM, 2019. doi: 10.1145/3299869.3300092. URL https://doi.org/10.1145/3299869.3300092.

[49] F. Geerts, A. Kementsietsidis, and D. Milano. Mondrian: Annotating and querying databases through colors and blocks. In 22nd International Conference on Data Engineering (ICDE'06), pages 82–82. IEEE, 2006.

[50] S. Ghayyur, D. Ghosh, X. He, and S. Mehrotra. MIDE: accuracy aware minimally invasive data exploration for decision support. Proc. VLDB Endow., 15(11):2653–2665, 2022. URL https://www.vldb.org/pvldb/vol15/p2653-ghayyur.pdf.

[51] B. Ghazi, R. Kumar, P. Manurangsi, and T. Steinke. Algorithms with more granular differential privacy guarantees. In Y. T. Kalai, editor, 14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA, volume 251 of LIPIcs, pages 54:1–54:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi: 10.4230/LIPICS.ITCS.2023.54. URL https://doi.org/10.4230/LIPIcs.ITCS.2023.54.

[52] A. M. Girgis, D. Data, S. Diggavi, A. T. Suresh, and P. Kairouz. On the renyi differential privacy of the shuffle model. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pages 2321–2341, 2021.

[53] Google Inc. Use differential privacy – Big Query Documentations. https://cloud.google.com/bigquery/docs/differential-privacy. Accessed: 2024-05-31.

[54] T. J. Green and V. Tannen. The semiring framework for database provenance. In E. Sallinger, J. V. den Bussche, and F. Geerts, editors, Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on

Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017, pages 93–99. ACM, 2017. doi: 10.1145/3034786.3056125. URL https://doi.org/10.1145/3034786.3056125.

[55] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In L. Libkin, editor, Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11-13, 2007, Beijing, China, pages 31–40. ACM, 2007. doi: 10.1145/1265530.1265535. URL https://doi.org/10.1145/1265530.1265535.

[56] M. Hardt and G. N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In 51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA, pages 61–70. IEEE Computer Society, 2010. doi: 10.1109/FOCS.2010.85. URL https://doi.org/10.1109/FOCS.2010.85.

[57] M. B. Hawes. Implementing differential privacy: Seven lessons from the 2020 united states census. Harvard Data Science Review, 2(2):4, 2020.

[58] M. Hay, A. Machanavajjhala, G. Miklau, Y. Chen, D. Zhang, and G. Bissias. Exploring privacy-accuracy tradeoffs using dpcomp. In Proceedings of the 2016 International Conference on Management of Data, pages 2101–2104, 2016.

[59] X. He, A. Machanavajjhala, and B. Ding. Blowfish privacy: tuning privacy-utility trade-offs using policies. In C. E. Dyreson, F. Li, and M. T. Özsu, editors, International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014, pages 1447–1458. ACM, 2014. doi: 10.1145/2588555.2588581. URL https://doi.org/10.1145/2588555.2588581.

[60] F. Houssiau, L. Rocher, and Y.-A. de Montjoye. On the difficulty of achieving differential privacy in practice: user-level guarantees in aggregate location data. Nature communications, 13(1):29, 2022.

[61] J. Hsu, A. Roth, and J. R. Ullman. Differential privacy for the analyst via private equilibrium computation. In D. Boneh, T. Roughgarden, and J. Feigenbaum, editors, Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013, pages 341–350. ACM, 2013. doi: 10.1145/2488608.2488651. URL https://doi.org/10.1145/2488608.2488651.

[62] M. Jagielski, J. R. Ullman, and A. Oprea. Auditing differentially private machine learning: How private is private sgd? In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/fc4ddc15f9f4b4b06ef7844d6bb53abf-Abstract.html.

[63] J. Jin, C. Chuengsatiansup, T. Murray, B. I. P. Rubinstein, Y. Yarom, and O. Ohrimenko. Elephants do not forget: Differential privacy with state continuity for privacy budget. CoRR, abs/2401.17628, 2024. doi: 10.48550/ARXIV.2401.17628. URL https://doi.org/10.48550/arXiv.2401.17628.

[64] M. F. S. John, G. Denker, P. Laud, K. Martiny, A. Pankova, and D. Pavlovic. Decision support for sharing data using differential privacy. In 2021 IEEE Symposium on Visualization for Cyber Security (VizSec), pages 26–35. IEEE, 2021.

[65] N. M. Johnson, J. P. Near, and D. Song. Towards practical differential privacy for SQL queries. Proc. VLDB Endow., 11(5):526–539, 2018. doi: 10.1145/3187009.3177733. URL http://www.vldb.org/pvldb/vol11/p526-johnson.pdf.

[66] N. M. Johnson, J. P. Near, J. M. Hellerstein, and D. Song. Chorus: a programming framework for building scalable differential privacy mechanisms. In IEEE European Symposium on Security and Privacy, EuroS&P 2020, Genoa, Italy, September 7-11, 2020, pages 535–551. IEEE, 2020. doi: 10.1109/EuroSP48549.2020.00041. URL https://doi.org/10.1109/EuroSP48549.2020.00041.

[67] Z. Jorgensen, T. Yu, and G. Cormode. Conservative or liberal? personalized differential privacy. In J. Gehrke, W. Lehner, K. Shim, S. K. Cha, and G. M. Lohman, editors, 31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13-17, 2015, pages 1023–1034. IEEE Computer Society, 2015. doi: 10.1109/ICDE.2015.7113353. URL https://doi.org/10.1109/ICDE.2015.7113353.

[68] P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. In F. Bach and D. Blei, editors, Proceedings of the 32nd International Conference on Machine Learning, volume 37 of Proceedings of Machine Learning Research, pages 1376–1385, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/kairouz15.html.

[69] P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. IEEE Trans. Inf. Theory, 63(6):4037–4049, 2017. doi: 10.1109/TIT.2017.2685505. URL https://doi.org/10.1109/TIT.2017.2685505.

[70] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias. Differentially private event sequences over infinite streams. Proceedings of the VLDB Endowment, 7(12), 2014.

[71] J. Kiefer. On minimum variance estimators. The Annals of Mathematical Statistics, 23(4):627–629, 1952.

[72] D. Kifer and A. Machanavajjhala. Pufferfish: A framework for mathematical privacy definitions. ACM Transactions on Database Systems (TODS), 39(1):1–36, 2014.

[73] K. Knopf. Framework for differentially private data analysis with multiple accuracy requirements. In G. Li, Z. Li, S. Idreos, and D. Srivastava, editors, SIGMOD '21: International Conference on Management of Data, Virtual Event, China, June 20-25, 2021, pages 2890–2892. ACM, 2021. doi: 10.1145/3448016.3450587. URL https://doi.org/10.1145/3448016.3450587.

[74] K. Kostopoulou, P. Tholoniat, A. Cidon, R. Geambasu, and M. Lécuyer. Turbo: Effective caching in differentially-private databases. In J. Flinn, M. I. Seltzer, P. Druschel, A. Kaufmann, and J. Mace, editors, Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023, pages 579–594. ACM, 2023. doi: 10.1145/3600006.3613174. URL https://doi.org/10.1145/3600006.3613174.

[75] I. Kotsogiannis, Y. Tao, X. He, M. Fanaeepour, A. Machanavajjhala, M. Hay, and G. Miklau. Privatesql: A differentially private SQL query engine. Proc. VLDB Endow., 12(11):1371–1384, 2019. doi: 10.14778/3342263.3342274. URL http://www.vldb.org/pvldb/vol12/p1371-kotsogiannis.pdf.

[76] I. Kotsogiannis, S. Doudalis, S. Haney, A. Machanavajjhala, and S. Mehrotra. One-sided differential privacy. In 36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020, pages 493–504. IEEE, 2020. doi: 10.1109/ICDE48307.2020.00049. URL https://doi.org/10.1109/ICDE48307.2020.00049.

[77] F. Koufogiannis, S. Han, and G. J. Pappas. Gradual release of sensitive data under differential privacy. J. Priv. Confidentiality, 7(2), 2016. doi: 10.29012/jpc.v7i2.649. URL https://doi.org/10.29012/jpc.v7i2.649.

[78] N. Küchler, E. Opel, H. Lycklama, A. Viand, and A. Hithnawi. Cohere: Managing differential privacy in large scale systems. IEEE Symposium on Security and Privacy, 2024. doi: 10.48550/arXiv.2301.08517. URL https://doi.org/10.48550/arXiv.2301.08517.

[79] M. Lécuyer. Practical privacy filters and odometers with rényi differential privacy and applications to differentially private deep learning. CoRR, abs/2103.01379, 2021. URL https://arxiv.org/abs/2103.01379.

[80] M. Lécuyer, R. Spahn, K. Vodrahalli, R. Geambasu, and D. Hsu. Privacy accounting and quality control in the sage differentially private ML platform. In T. Brecht and C. Williamson, editors, Proceedings of the 27th ACM Symposium on Operating Systems Principles, SOSP 2019, Huntsville, ON, Canada, October 27-30, 2019, pages 181–195. ACM, 2019. doi: 10.1145/3341301.3359639. URL https://doi.org/10.1145/3341301.3359639.

[81] C. Li, M. Hay, G. Miklau, and Y. Wang. A data- and workload-aware query answering algorithm for range queries under differential privacy. Proc. VLDB Endow., 7(5):341–352, 2014. doi: 10.14778/2732269.2732271. URL http://www.vldb.org/pvldb/vol7/p341-li.pdf.

[82] C. Li, G. Miklau, M. Hay, A. McGregor, and V. Rastogi. The matrix mechanism: optimizing linear counting queries under differential privacy. VLDB J., 24(6):757–781, 2015. doi: 10.1007/s00778-015-0398-x. URL https://doi.org/10.1007/s00778-015-0398-x.

[83] K. Ligett, S. Neel, A. Roth, B. Waggoner, and Z. S. Wu. Accuracy first: Selecting a differential privacy level for accuracy constrained erm. In Neural Information Processing Systems, 2017. URL https://api.semanticscholar.org/CorpusID:19294675.

[84] T. Luo, M. Pan, P. Tholoniat, A. Cidon, R. Geambasu, and M. Lécuyer. Privacy budget scheduling. In A. D. Brown and J. R. Lorch, editors, 15th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2021, July 14-16, 2021, pages 55–74. USENIX Association, 2021. URL https://www.usenix.org/conference/osdi21/presentation/luo.

[85] X. Luo, J. Pei, Z. Cong, and C. Xu. On shapley value in data assemblage under independent utility. Proc. VLDB Endow., 15(11):2761–2773, 2022. doi: 10.14778/3551793.3551829. URL https://www.vldb.org/pvldb/vol15/p2761-luo.pdf.

[86] X. Luo, J. Pei, C. Xu, W. Zhang, and J. Xu. Fast shapley value computation in data assemblage tasks as cooperative simple games. Proc. ACM Manag. Data, 2(1):56:1–56:28, 2024. doi: 10.1145/3639311. URL https://doi.org/10.1145/3639311.

[87] M. Mazmudar, T. Humphries, J. Liu, M. Rafuse, and X. He. Cache me if you can: Accuracy-aware inference engine for differentially private data exploration. Proc. VLDB Endow., 16(4):574–586, 2022. URL https://www.vldb.org/pvldb/vol16/p574-mazmudar.pdf.

[88] F. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In U. Çetintemel, S. B. Zdonik, D. Kossmann, and N. Tatbul, editors, Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2009, Providence, Rhode Island, USA, June 29 - July 2, 2009, pages 19–30. ACM, 2009. doi: 10.1145/1559845.1559850. URL https://doi.org/10.1145/1559845.1559850.

[89] P. Mohan, A. Thakurta, E. Shi, D. Song, and D. E. Culler. GUPT: privacy preserving data analysis made easy. In K. S. Candan, Y. Chen, R. T. Snodgrass, L. Gravano, and A. Fuxman, editors, Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2012, Scottsdale,

AZ, USA, May 20-24, 2012, pages 349–360. ACM, 2012. doi: 10.1145/2213836.2213876. URL https://doi.org/10.1145/2213836.2213876.

[90] P. Nanayakkara, J. Bater, X. He, J. Hullman, and J. Rogers. Visualizing privacy-utility trade-offs in differentially private data releases. Proc. Priv. Enhancing Technol., 2022(2):601–618, 2022. doi: 10.2478/POPETS-2022-0058. URL https://doi.org/10.2478/popets-2022-0058.

[91] M. Nasr, J. Hayes, T. Steinke, B. Balle, F. Tramèr, M. Jagielski, N. Carlini, and A. Terzis. Tight auditing of differentially private machine learning. In J. A. Calandrino and C. Troncoso, editors, 32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023, pages 1631–1648. USENIX Association, 2023. URL https://www.usenix.org/conference/usenixsecurity23/presentation/nasr.

[92] J. P. Near and X. He. Differential privacy for databases. Found. Trends Databases, 11(2):109–225, 2021. doi: 10.1561/1900000066. URL https://doi.org/10.1561/1900000066.

[93] J. P. Near, D. Darais, C. Abuah, T. Stevens, P. Gaddamadugu, L. Wang, N. Somani, M. Zhang, N. Sharma, A. Shan, et al. Duet: an expressive higher-order language and linear type system for statically enforcing differential privacy. Proceedings of the ACM on Programming Languages, 3(OOPSLA):1–30, 2019.

[94] P. Nguyen, A. Silence, D. Darais, and J. P. Near. Duetsgx: Differential privacy with secure hardware. CoRR, abs/2010.10664, 2020. URL https://arxiv.org/abs/2010.10664.

[95] K. Nissim, S. Raskhodnikova, and A. D. Smith. Smooth sensitivity and sampling in private data analysis. In D. S. Johnson and U. Feige, editors, Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007, pages 75–84. ACM, 2007. doi: 10.1145/1250790.1250803. URL https://doi.org/10.1145/1250790.1250803.

[96] OpenDP. Developing open source tools for differential privacy. https://docs.opendp.org/en/stable/index.html. Accessed: 2024-05-31.

[97] B. Pan, N. Stakhanova, and S. Ray. Data provenance in security and privacy. ACM Comput. Surv., 55 (14s):323:1–323:35, 2023. doi: 10.1145/3593294. URL https://doi.org/10.1145/3593294.

[98] P. Pappachan, S. Zhang, X. He, and S. Mehrotra. Don't be a tattle-tale: Preventing leakages through data dependencies on access control protected data. Proc. VLDB Endow., 15(11):2437–2449, 2022. URL https://www.vldb.org/pvldb/vol15/p2437-pappachan.pdf.

[99] P. Pappachan, S. Zhang, X. He, and S. Mehrotra. Preventing inferences through data dependencies on sensitive data. IEEE Transactions on Knowledge and Data Engineering, to appear. URL https://doi.org/10.1109/TKDE.2023.3336630.

[100] S. Peng, Y. Yang, Z. Zhang, M. Winslett, and Y. Yu. Query optimization for differentially private data management systems. In C. S. Jensen, C. M. Jermaine, and X. Zhou, editors, 29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8-12, 2013, pages 1093–1104. IEEE Computer Society, 2013. doi: 10.1109/ICDE.2013.6544900. URL https://doi.org/10.1109/ICDE.2013.6544900.

[101] K. Pillutla, G. Andrew, P. Kairouz, H. B. McMahan, A. Oprea, and S. Oh. Unleashing the power of randomization in auditing differentially private ML. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December

10 - 16, 2023, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/d09ef5264966e17adffd3157265c9946-Abstract-Conference.html.

[102] D. Proserpio, S. Goldberg, and F. McSherry. Calibrating data to sensitivity in private data analysis. Proc. VLDB Endow., 7(8):637–648, 2014. doi: 10.14778/2732296.2732300. URL http://www.vldb.org/pvldb/vol7/p637-proserpio.pdf.

[103] D. Pujol, Y. Wu, B. Fain, and A. Machanavajjhala. Budget sharing for multi-analyst differential privacy. Proc. VLDB Endow., 14(10):1805–1817, 2021. doi: 10.14778/3467861.3467870. URL http://www.vldb.org/pvldb/vol14/p1805-pujol.pdf.

[104] D. Pujol, A. Sun, B. Fain, and A. Machanavajjhala. Multi-analyst differential privacy for online query answering. Proc. VLDB Endow., 16(4):816–828, 2022. URL https://www.vldb.org/pvldb/vol16/p816-pujol.pdf.

[105] C. R. Rao. Sufficient statistics and minimum variance estimates. In Mathematical Proceedings of the Cambridge Philosophical Society, volume 45, pages 213–218. Cambridge University Press, 1949.

[106] C. Ré and D. Suciu. Approximate lineage for probabilistic databases. Proceedings of the VLDB Endowment, 1(1):797–808, 2008.

[107] R. Rogers, S. Subramaniam, S. Peng, D. Durfee, S. Lee, S. K. Kancha, S. Sahay, and P. Ahammad. Linkedin's audience engagements api: A privacy preserving data analytics system at scale. arXiv preprint arXiv:2002.05839, 2020.

[108] R. M. Rogers, S. P. Vadhan, A. Roth, and J. R. Ullman. Privacy odometers and filters: Pay-as-you-go composition. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, pages 1921–1929, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/58c54802a9fb9526cd0923353a34a7ae-Abstract.html.

[109] A. Roy Chowdhury, C. Wang, X. He, A. Machanavajjhala, and S. Jha. Crypte: Crypto-assisted differential privacy on untrusted servers. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, pages 603–619, 2020.

[110] A. D. Smith. Privacy-preserving statistical estimation with optimal convergence rates. In L. Fortnow and S. P. Vadhan, editors, Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011, pages 813–822. ACM, 2011. doi: 10.1145/1993636.1993743. URL https://doi.org/10.1145/1993636.1993743.

[111] Snowflake Inc. Differential privacy in Snowflake data clean rooms. https://docs.snowflake.com/en/user-guide/cleanrooms/differential-privacy. Accessed: 2024-05-31.

[112] D. Srivastava and Y. Velegrakis. Intensional associations between data and metadata. In Proceedings of the 2007 ACM SIGMOD international conference on Management of data, pages 401–412, 2007.

[113] D. Sun, W. Dong, and K. Yi. Confidence intervals for private query processing. Proc. VLDB Endow., 17 (3):373–385, 2023. URL https://www.vldb.org/pvldb/vol17/p373-sun.pdf.

[114] J. Tang, A. Korolova, X. Bai, X. Wang, and X. Wang. Privacy loss in apple's implementation of differential privacy on macos 10.12. Theory and Practice of Differential Privacy (TPDP) Workshop, 2017.

[115] Y. Tao, A. Gilad, A. Machanavajjhala, and S. Roy. Dpxplain: Privately explaining aggregate query answers. Proc. VLDB Endow., 16(1):113–126, 2022. doi: 10.14778/3561261.3561271. URL https://www.vldb.org/pvldb/vol16/p113-tao.pdf.

[116] P. Thaker, M. Budiu, P. Gopalan, U. Wieder, and M. Zaharia. Overlook: Differentially private exploratory visualization for big data. arXiv preprint arXiv:2006.12018, 2020.

[117] Transcend Inc. Next-gen privacy management. https://transcend.io/. Accessed: 2024-05-31.

[118] TUMULT Labs. Tumult analysis. https://www.tmlt.dev/. Accessed: 2024-05-31.

[119] S. Vadhan and T. Wang. Concurrent composition of differential privacy. In Theory of Cryptography: 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8–11, 2021, Proceedings, Part II 19, pages 582–604. Springer, 2021.

[120] S. Vadhan and W. Zhang. Concurrent composition theorems for differential privacy. In Proceedings of the 55th Annual ACM Symposium on Theory of Computing, pages 507–519, 2023.

[121] E. L. Vesga, A. Russo, and M. Gaboardi. A programming framework for differential privacy with accuracy concentration bounds. In 2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020, pages 411–428. IEEE, 2020. doi: 10.1109/SP40000.2020.00086. URL https://doi.org/10.1109/SP40000.2020.00086.

[122] S. Wagh, X. He, A. Machanavajjhala, and P. Mittal. Dp-cryptography: marrying differential privacy and cryptography in emerging applications. Communications of the ACM, 64(2):84–93, 2021.

[123] Y. Wang, Z. Ding, G. Wang, D. Kifer, and D. Zhang. Proving differential privacy with shadow execution. In Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation, pages 655–669, 2019.

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

[125] Y. Wang, Z. Ding, Y. Xiao, D. Kifer, and D. Zhang. Dpgen: Automated program synthesis for differential privacy. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pages 393–411, 2021.

[126] J. Whitehouse, A. Ramdas, Z. S. Wu, and R. M. Rogers. Brownian noise reduction: Maximizing privacy subject to accuracy constraints. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, 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, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/48aaa5ea741ae8430bd58e25917d267d-Abstract-Conference.html.

[127] J. Whitehouse, A. Ramdas, R. Rogers, and S. Wu. Fully-adaptive composition in differential privacy. In International Conference on Machine Learning, pages 36990–37007. PMLR, 2023.

[128] R. J. Wilson, C. Y. Zhang, W. Lam, D. Desfontaines, D. Simmons-Marengo, and B. Gipson. Differentially private SQL with bounded user contribution. Proc. Priv. Enhancing Technol., 2020(2):230–250, 2020. doi: 10.2478/popets-2020-0025. URL https://doi.org/10.2478/popets-2020-0025.

[129] D. Winograd-Cort, A. Haeberlen, A. Roth, and B. C. Pierce. A framework for adaptive differential privacy. Proceedings of the ACM on Programming Languages, 1(ICFP):1–29, 2017.

[130] Y. Wu, V. Tannen, and S. B. Davidson. Provenance-based model maintenance: Implications for privacy. IEEE Data Eng. Bull., 45(1):37–49, 2022. URL http://sites.computer.org/debull/A22mar/p37.pdf.

[131] X. Xiao, G. Bender, M. Hay, and J. Gehrke. ireduct: differential privacy with reduced relative errors. In T. K. Sellis, R. J. Miller, A. Kementsietsidis, and Y. Velegrakis, editors, Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2011, Athens, Greece, June 12-16, 2011, pages 229–240. ACM, 2011. doi: 10.1145/1989323.1989348. URL https://doi.org/10.1145/1989323.1989348.

[132] Y. Xiao, Z. Ding, Y. Wang, D. Zhang, and D. Kifer. Optimizing fitness-for-use of differentially private linear queries. Proc. VLDB Endow., 14(10):1730–1742, 2021. doi: 10.14778/3467861.3467864. URL http://www.vldb.org/pvldb/vol14/p1730-xiao.pdf.

[133] Y. Xiao, G. Wang, D. Zhang, and D. Kifer. Answering private linear queries adaptively using the common mechanism. CoRR, abs/2212.00135, 2022. doi: 10.48550/arXiv.2212.00135. URL https://doi.org/10.48550/arXiv.2212.00135.

[134] L. Yu, S. Zhang, L. Zhou, Y. Meng, S. Du, and H. Zhu. Thwarting longitudinal location exposure attacks in advertising ecosystem via edge computing. In 42nd IEEE International Conference on Distributed Computing Systems, ICDCS 2022, Bologna, Italy, July 10-13, 2022, pages 470–480. IEEE, 2022. doi: 10.1109/ICDCS54860.2022.00052. URL https://doi.org/10.1109/ICDCS54860.2022.00052.

[135] D. Zhang and D. Kifer. Lightdp: Towards automating differential privacy proofs. In Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, pages 888–901, 2017.

[136] D. Zhang, R. McKenna, I. Kotsogiannis, M. Hay, A. Machanavajjhala, and G. Miklau. EKTELO: A framework for defining differentially-private computations. In G. Das, C. M. Jermaine, and P. A. Bernstein, editors, Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018, pages 115–130. ACM, 2018. doi: 10.1145/3183713.3196921. URL https://doi.org/10.1145/3183713.3196921.

[137] S. Zhang. DProvSQL: Accuracy-aware privacy provenance framework for differentially private sql engine. Master's thesis, University of Waterloo, 2022.

[138] S. Zhang and X. He. DProvDB: Differentially private query processing with multi-analyst provenance. In Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2024. ACM, 2024.

[139] S. Zhang, X. He, A. Kundu, S. Mehrotra, and S. Sharma. Secure normal form: Mediation among cross cryptographic leakages in encrypted databases. In IEEE International Conference on Data Engineering (ICDE). IEEE, 2024.

[140] W. Zhang, O. Ohrimenko, and R. Cummings. Attribute privacy: Framework and mechanisms. In FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022, pages 757–766. ACM, 2022. doi: 10.1145/3531146.3533139. URL https://doi.org/10.1145/3531146.3533139.