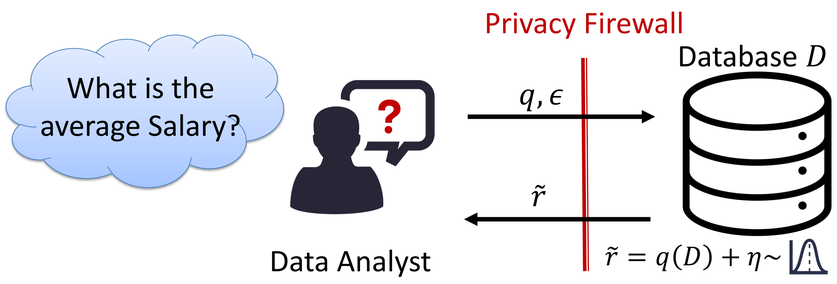
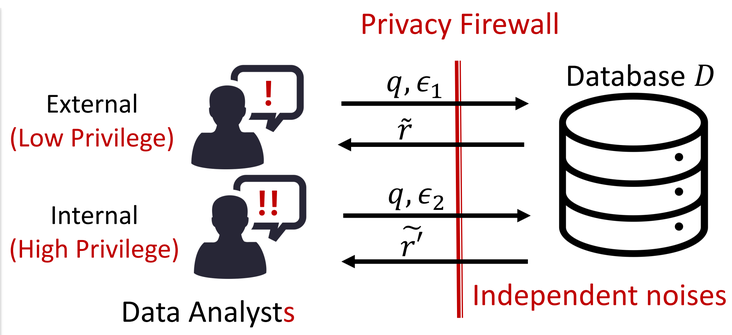
DProvDB: Differentially Private Query Processing with Multi-Analyst Provenance

Shufan Zhang and Xi He
University of Waterloo

Problem Setup

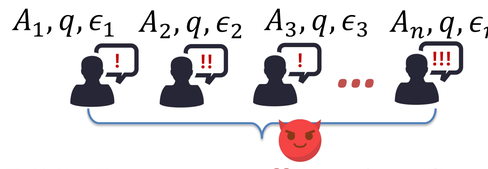


DP Threat Model
[SIGMOD'09] [VLDB'18] [VLDB'19]
[PoPETs'19] [EuroSys'20] ...



But What is in ***Practice***...

RQ1. Worst-case privacy bound across analysts?



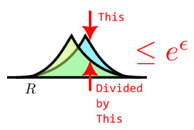
[**Upper bound**] $\sum_{j=1}^{n} \epsilon_j$
[**Lower bound**] $\max_j \epsilon_j$
GAP!

RQ2. Resource allocation & management:
- Maximize query answering?
- Fair query answering for Online System?

Differential Privacy (DP)

[**DP**] A mechanism M is (ϵ, δ)-DP, if for any $D \cong D'$, and all $O \subseteq \mathcal{O}$, we have

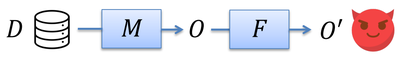
$$\Pr[M(D) \in O] \leq e^{\epsilon} \Pr[M(D') \in O] + \delta.$$



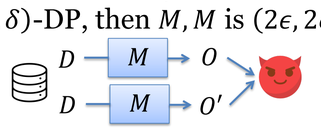
[(Analytic) **Gaussian Mechanism**]
$M(D) = q(D) + \eta \sim N(0, \sigma^2 I)$ satisfies (ϵ, δ)-DP, if

$$\Phi_N\left(\frac{\Delta q}{2\sigma} - \frac{\epsilon\sigma}{\Delta q}\right) - e^{\epsilon}\Phi_N\left(-\frac{\Delta q}{2\sigma} - \frac{\epsilon\delta}{\Delta q}\right) \leq \delta$$

[**DP Properties**]
1.Post-Processing: if M is (ϵ, δ)-DP, then $F \circ M$ is (ϵ, δ)-DP as well!



2.Sequential Composition: if M is (ϵ, δ)-DP, then M, M is $(2\epsilon, 2\delta)$-DP.



Multi-Analyst DP (Our New DP Variant)

[**Multi-analyst DP**] A mechanism M is $[(A_1, \epsilon_1, \delta_1), \ldots, (A_n, \epsilon_n, \delta_n)]$-multi-analyst-DP, if for any $D \cong D'$, any j $\in [n]$, and all $O_j \subseteq \mathcal{O}$, we have

$$\Pr[M(D) \in O_j] \leq e^{\epsilon_j} \Pr[M(D') \in O_j] + \delta_j.$$

[**Multi-analyst DP Properties**]
1.Post-Processing: hold;
2.Sequential Composition: hold on each coordinate.

[**DP vs. Multi-analyst DP**]
- DP guarantees an overall bound by privacy budget;
- Multi-analyst DP guarantees an individual privacy bound for each data analyst.

[**Multi-analyst DP implies DP**] By applying sequential composition, multi-analyst DP trivially implies a DP bound.

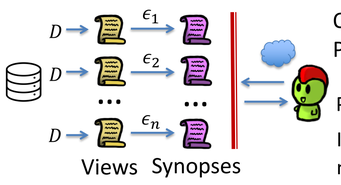
Our New DP System — DProvDB

Design Principles

- P1. View-based privacy management
- P2. Fine-grained privacy provenance
- P3. Dual query submission mode (c.f. our paper)
- P4. Maximum and fair query answering

Answering Queries on *Views*

- Directly answering queries on fresh DB is not good [CIDR'19]
- Instead, answer queries over private snapshots [VLDB'19, VLDB'23]



Query answering becomes Post-Processing! [VLDB'19]

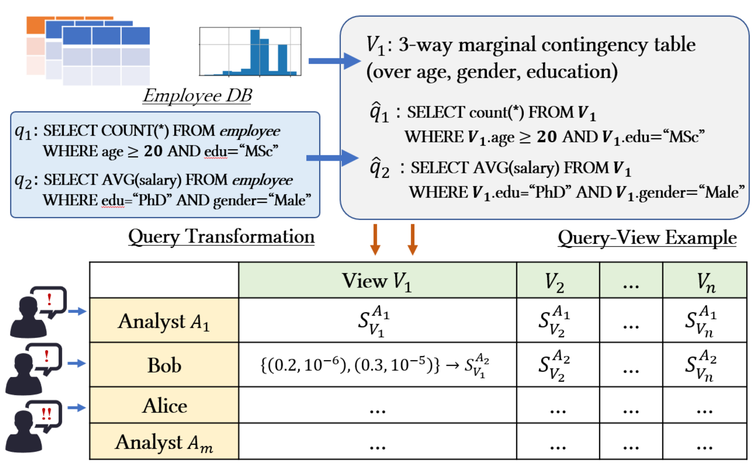
Pre-allocation s.t. $\epsilon = \sum \epsilon_i$.

If incoming $\epsilon_q > \epsilon_i$, query not answerable.

Dynamically!

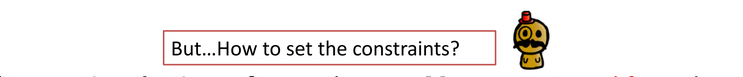
But how could we better allocate budgets? (RQ2)

Privacy Provenance Framework



[**State of Privacy Loss**] S_i^j, i.e. the entry of the provenance table. The current consumed privacy budget on View i by Analyst j.

[**Privacy Constraints**] Privacy constraints are max allowed budget consumption. The privacy provenance table is set with 3 types of constraints: table, column, and analyst constraints. If any one is not satisfied, the query will be rejected.



But...How to set the constraints?

[**Proportional Fairness**] A mechanism M is proportional fair, where each analyst A_i is with privilege l_i, if $\forall A_i, A_j\ (i \neq j), l_i < l_j$, we have

$$\frac{Err_i(M, A_i, Q)}{\mu(l_i)} \leq \frac{Err_j(M, A_j, Q)}{\mu(l_j)}.$$

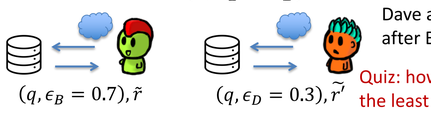
This is about analyst constraint...How about column and table constraints?

Quiz: Could you help our admin, Carol?

Our New DP Mechanism (Additive Gaussian Approach)

Additive Gaussian Mechanism (additive GM)

[**Sum of Gaussian**] $X \sim N(0, \sigma_1^2), Y \sim N(0, \sigma_2^2)$, then $Z = X + Y \sim N(0, \sigma_1^2 + \sigma_2^2)$.



Dave asks the same query after Bob with less budget.

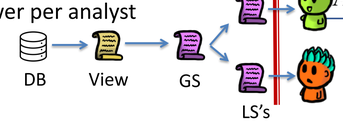
Quiz: how to answer Dave with the least budget consumption?

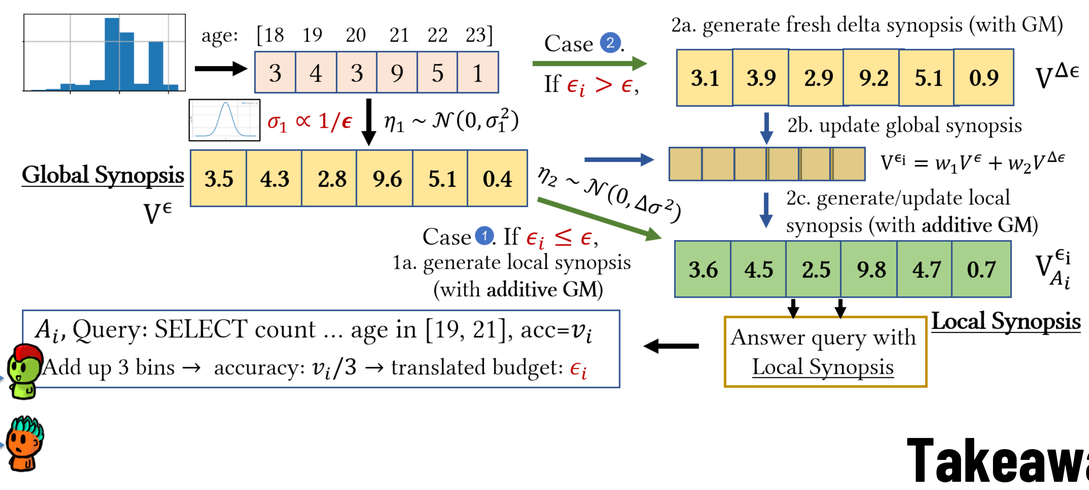
Solution: Record $\tilde{r}$, and calculate $\tilde{r}' = \tilde{r} + \eta \sim N(\sigma_D^2 - \sigma_B^2)$.
No additional budget consumed!

Wait...Record too many data?

[**Global Synopsis**] The private answer to a view of DB.

[**Local Synopsis**] The private answer per analyst generated from global synopsis.



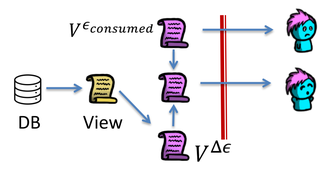


[**Analysis of additive GM**] Additive GM is $[(A_1, \epsilon_1, \delta_1), \ldots, (A_n, \epsilon_n, \delta_n)]$-multi-analyst-DP, and guarantees $\max_j \epsilon_j$-DP.

GAP Closed!

Yes, additive GM gives us nice bound to generate answers when $\epsilon_i < \epsilon_{consumed}$, but how do we do if later, Alice asks a query with higher budget?

[**Synopsis Update**] When $\epsilon_i > \epsilon_{consumed}$, we update the current synopsis based UMVUE, i.e., $V^{\epsilon_i} = w_1 V^{\epsilon_{consumed}} + w_2 V^{\Delta\epsilon}$.

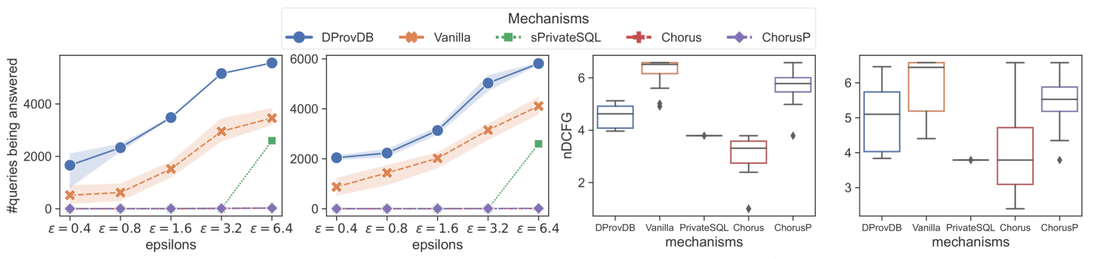


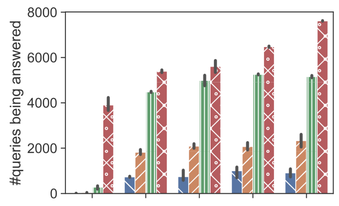
But...What if this query is asked by another analyst?

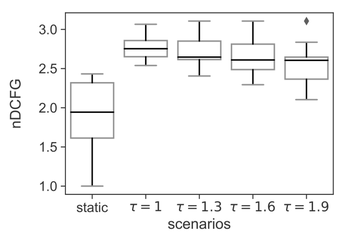
Solution: See the figure in the middle.

Evaluation

Dataset: Adult, TPC-H
Baseline: sPrivateSQL [VLDB'19], Chorus [EuroSys'20]
Goal: 1) End-to-End Comparison, on Utility and Fairness (Bottom↓)
2) Trading-off Fairness for Utility (Right→)







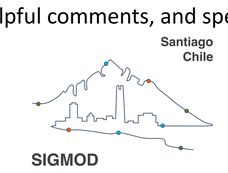
Takeaways

- A serious step to make DP query processing more practical!
- A multi-analyst interface can improve the system utility over existing DP approaches based on standard composition.
- DProvDB is the first "stateful" DP query-processing system.
- DProvDB can be benefit most, if not all, exiting DP query systems, and can be integrated as a middleware solution.
- Blue ocean for future work in DP + access control
  - We are happy to see more research join the discussion!

Acknowledgement

Thank Runchao Jiang, Florian Kerschbaum, Semih Salihoglu for helpful comments, and special thanks to Simon Oya for lovely cartoon drawings!

UNIVERSITY OF WATERLOO | DSG Data Systems Group



SIGMOD 2024
Santiago, Chile