Privacy in Web Advertising: Analytics & Modeling

Badih Ghazi & Pasin Manurangsi
Google Research

Joint work with colleagues at Google

We will discuss tools for private analytics and learning for web advertising applications.

Outline

Part I: Basic & Analytics

- Privacy Attacks
- Differential Privacy: Basics
- DP Properties
- Noise Addition Mechanisms
- Threat Models
  - Central vs Local DP
  - Shuffle DP, SMPC and TEEs
- DP Ad Analytics
  - Conversion Measurement
  - Reach and Frequency

Part II: Learning & Other Topics

- DP Ad Modeling
  - Full DP
  - Label DP
  - DP with partially known features
- Future Directions

Online Advertising: Terminology

Entities

Advertiser

An entity / website / app that pays to have their ads shown to the users

Publisher

The website that shows the ads to the users

Will not focus on ad-tech in this tutorial

Ad-Tech

Entity that helps advertiser / publisher buying / selling / optimizing ads

Events

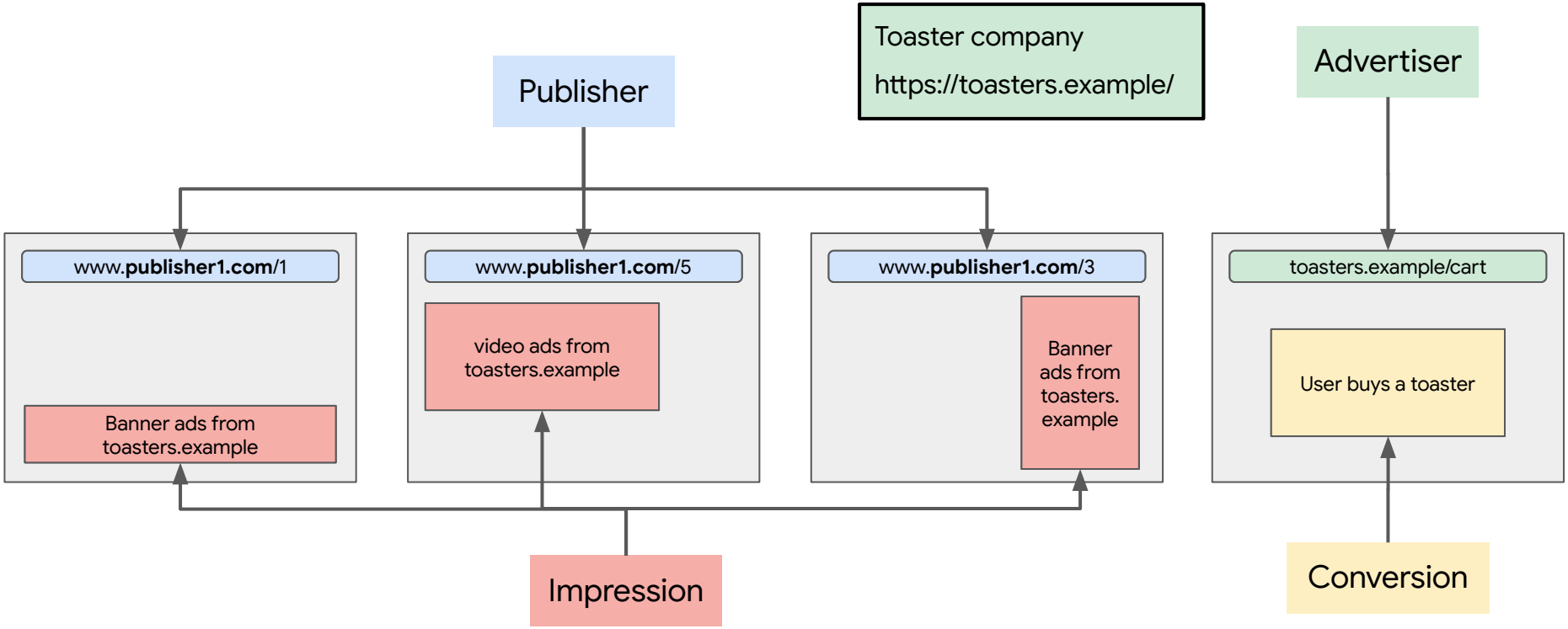
Impression

Event indicating that an ad is viewed or clicked by a user

Conversion

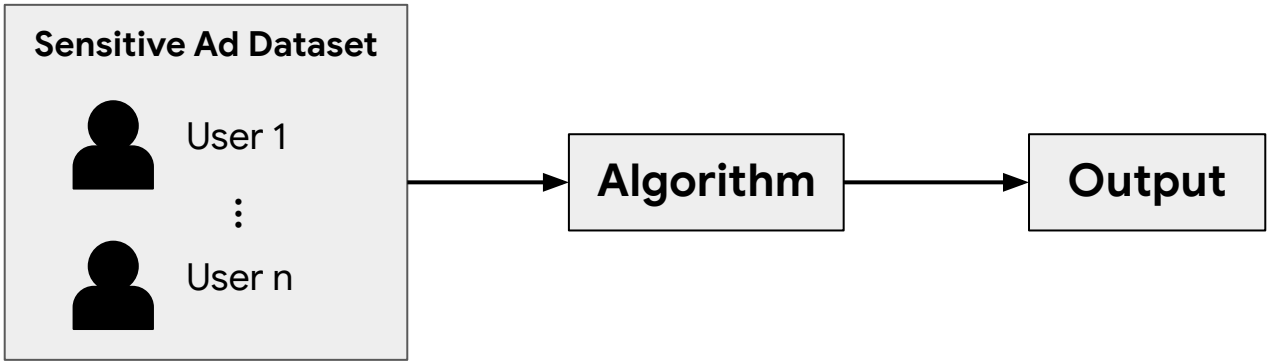
Action by the users that is valuable to advertisers

Online Advertising: Example



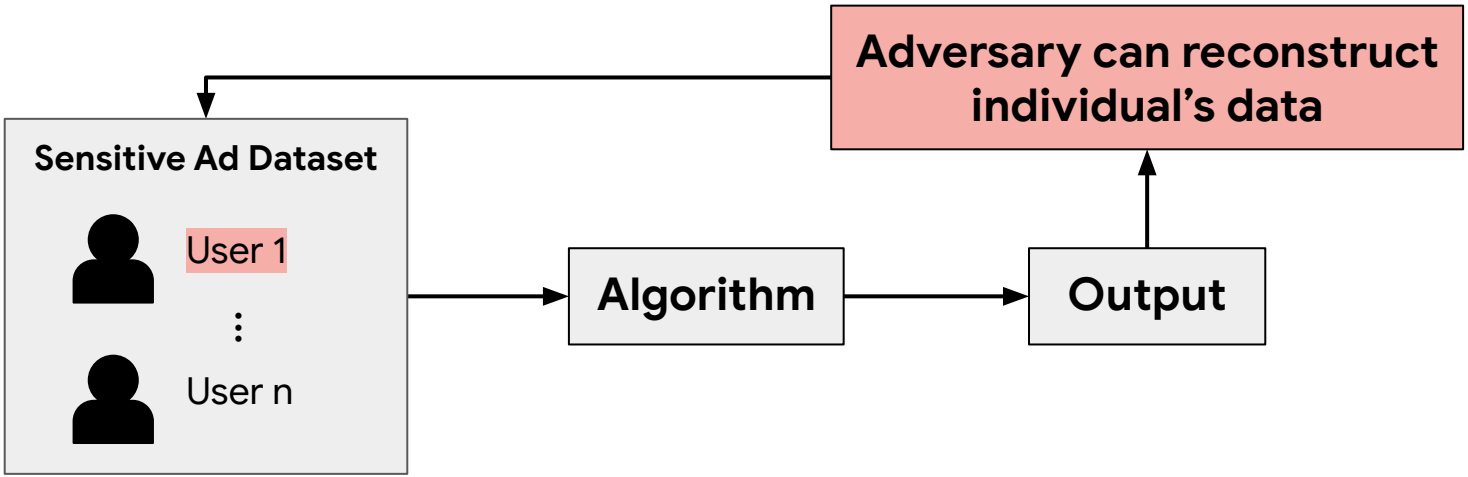
Privacy Attacks

Basic Setting



Is the algorithm *not* privacy-preserving?

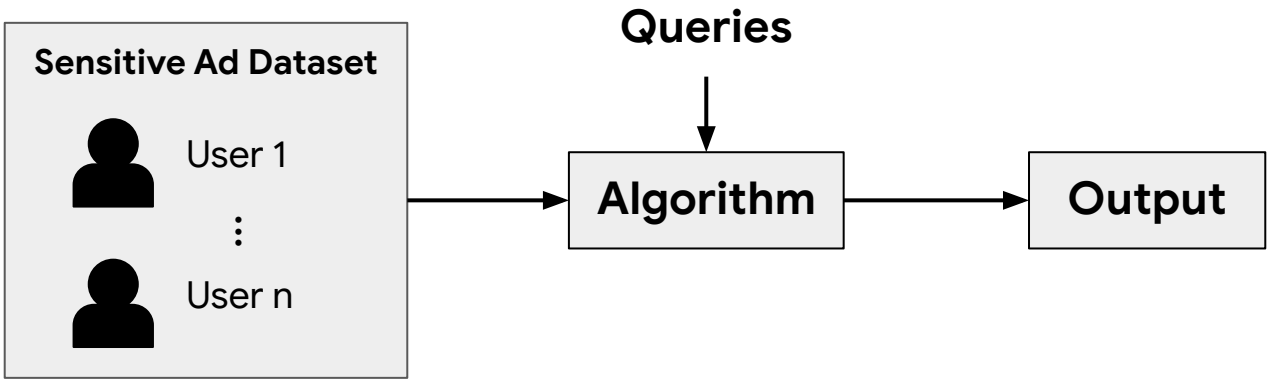
Reconstruction Attacks



Is the algorithm *not* privacy-preserving?

Attacks Against Query-Answering System

Query-Answering System

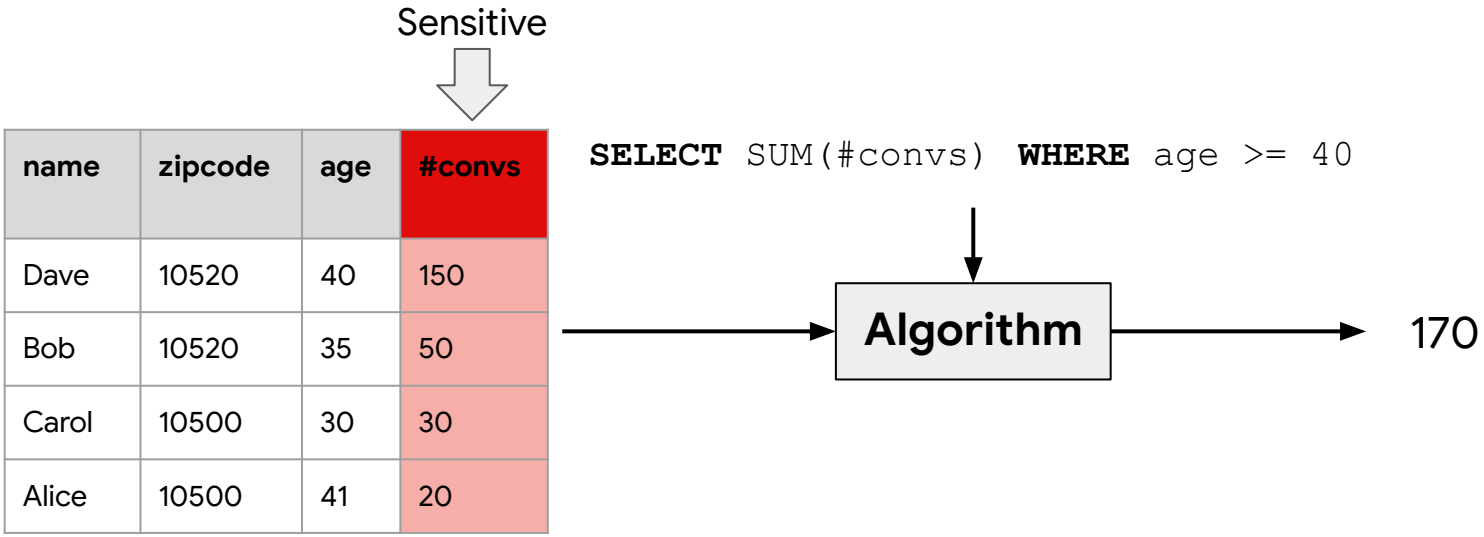


Reconstruction Attacks

name	zipcode	age	#convs
Dave	10520	40	150
Bob	10520	35	50
Carol	10500	30	30
Alice	10500	41	20

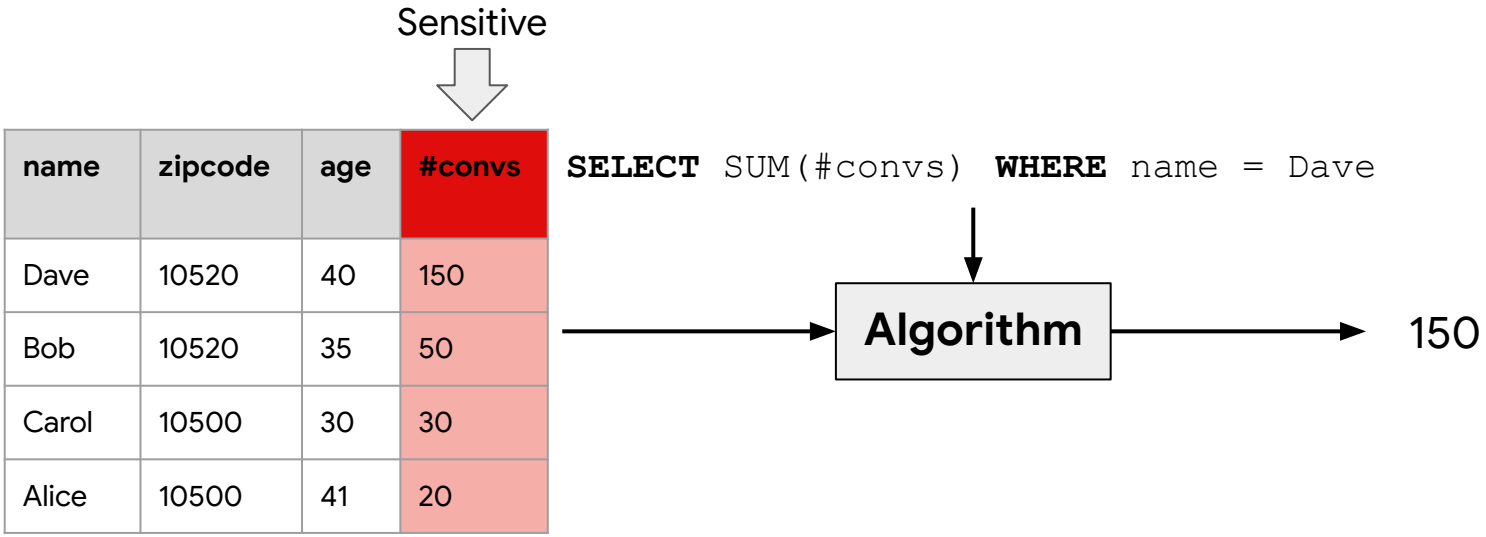
- Simplifying assumptions: only allow **`SUM`** (or **`COUNT`**) queries
- Only one sensitive column

Reconstruction Attacks

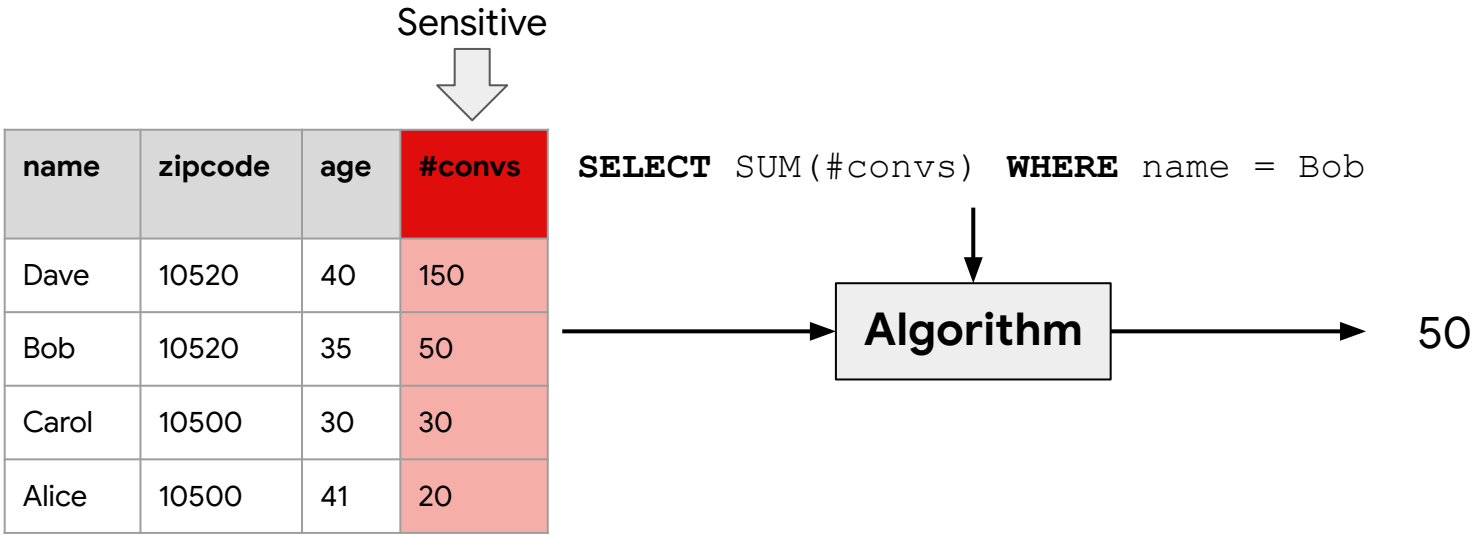


- Simplifying assumptions: only allow **SUM** (or **COUNT**) queries
- Only one sensitive column

Setting I: Adversary Can Select Queries

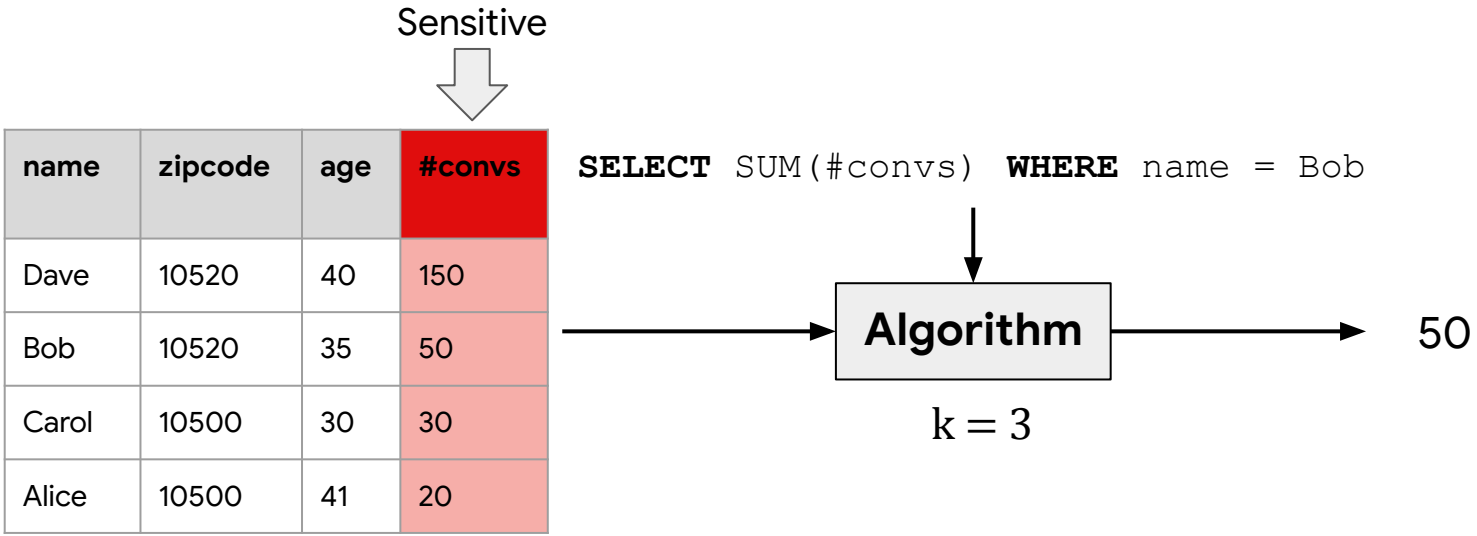


Setting I: Adversary Can Select Queries



The adversary can construct the entire dataset!

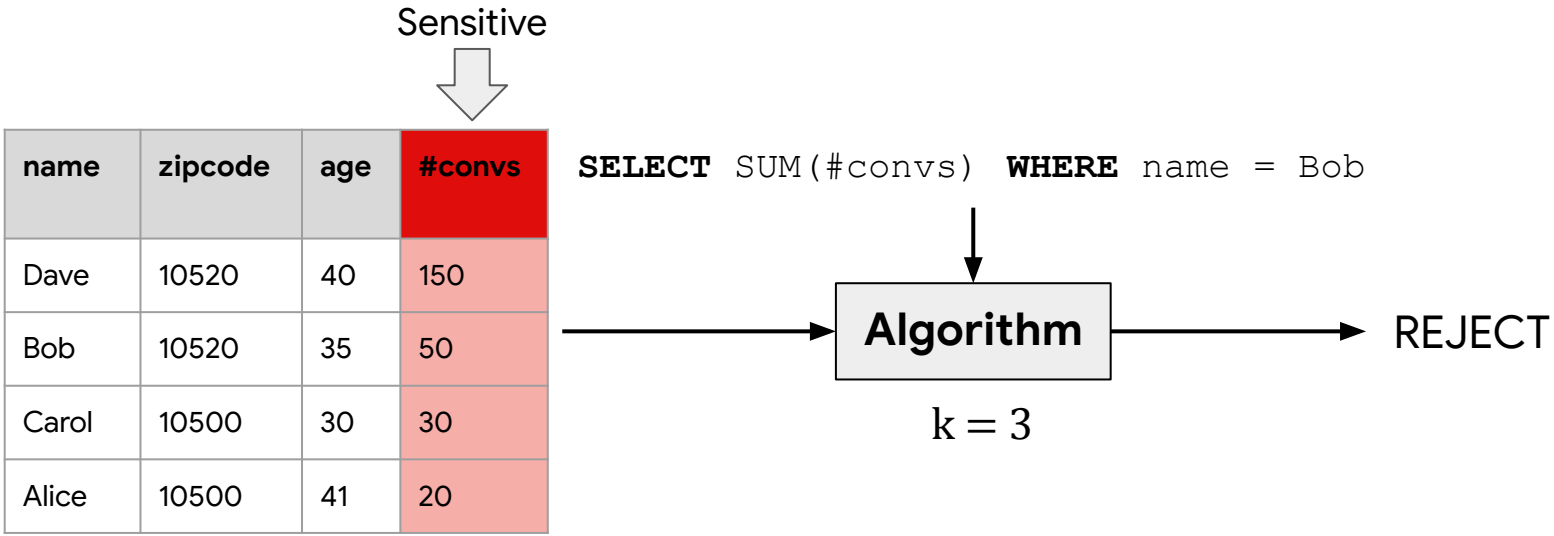
Mitigation I: "k-Anonymity"



k-Anonymity

Algorithm rejects if query involves $< k$ individuals

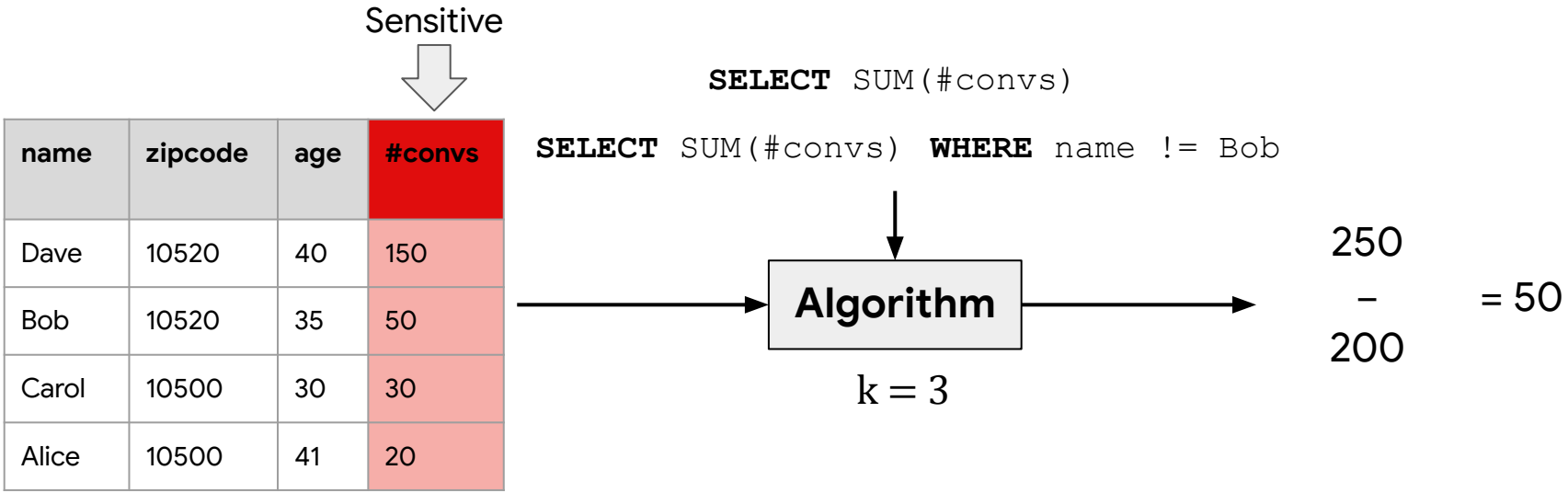
Mitigation I: "k-Anonymity"



k-Anonymity

Algorithm rejects if query involves $< k$ individuals

Differencing Attacks

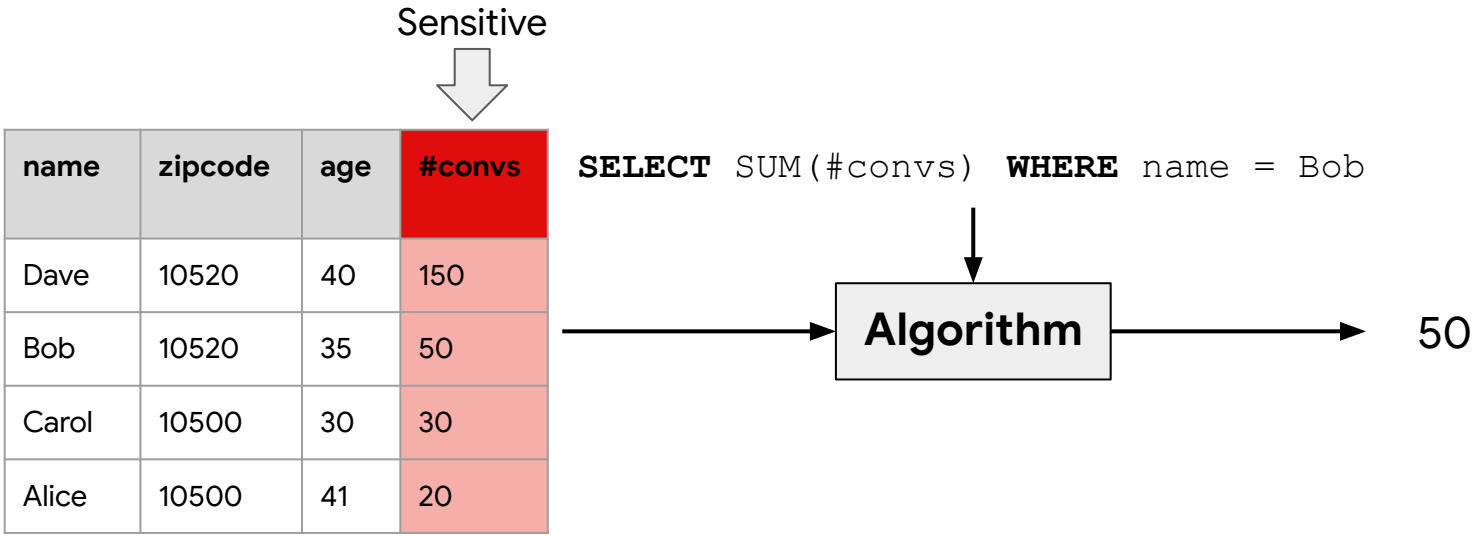


k-Anonymity

Algorithm rejects if query involves $< k$ individuals

Hard to extend if adversary can combine >1 queries

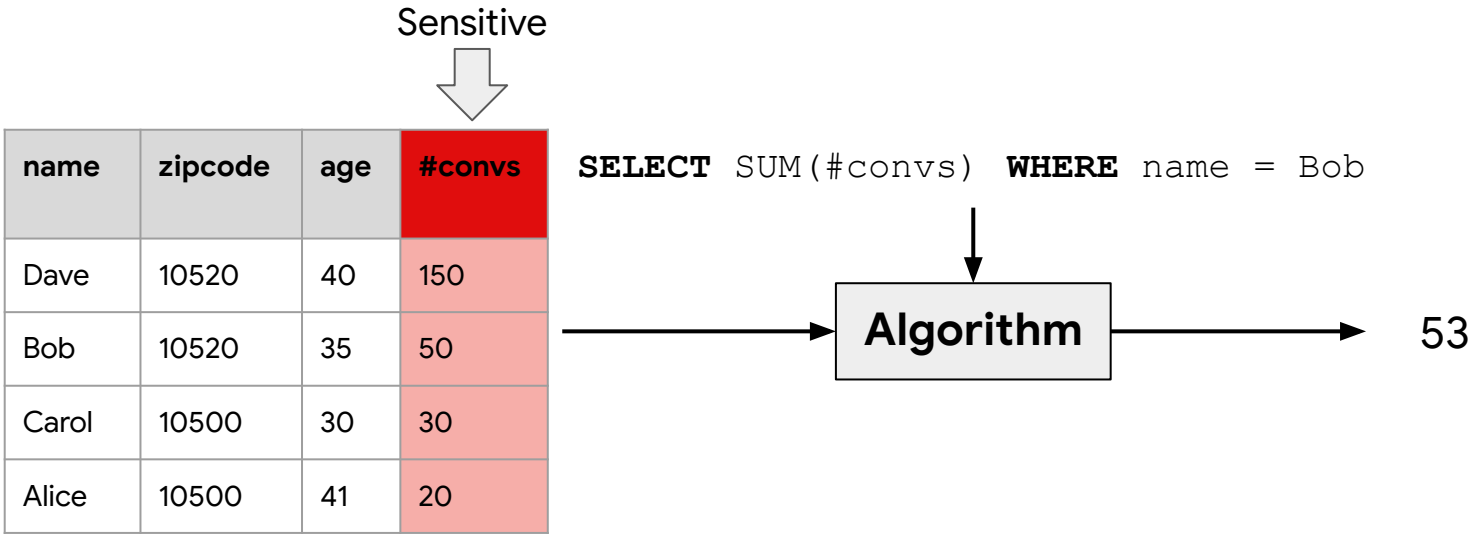
Mitigation II: Add Noise



Noise Addition

Add random noise to the answer

Mitigation II: Add Noise



Noise Addition
Add random noise to the answer

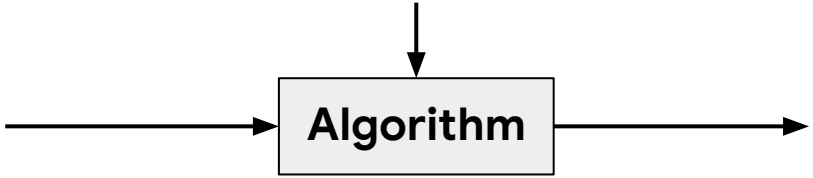
An approach taken by *differential privacy*

Setting II: Adversary *Cannot* Select Queries

Sensitive

name	zipcode	age	#convs
Dave	10520	40	150
Bob	10520	35	50
Carol	10500	30	30
Alice	10500	41	20

```
SELECT SUM(#convs) WHERE age >= 40
SELECT SUM(#convs) WHERE age < 40
SELECT SUM(#convs) WHERE zipcode = 10500
SELECT SUM(#convs) WHERE age != 41
```



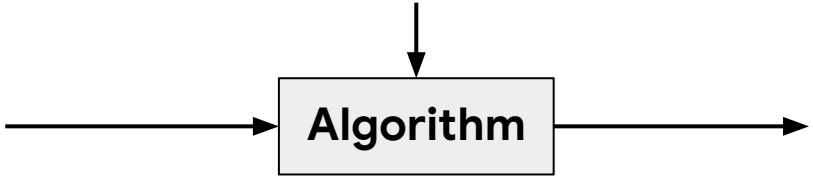
170
80
50
230

Setting II: Adversary *Cannot* Select Queries

Sensitive

name	zipcode	age	#convs
Dave	10520	40	x_1
Bob	10520	35	x_2
Carol	10500	30	x_3
Alice	10500	41	x_4

```
SELECT SUM(#convs) WHERE age >= 40
SELECT SUM(#convs) WHERE age < 40
SELECT SUM(#convs) WHERE zipcode = 10500
SELECT SUM(#convs) WHERE age != 41
```



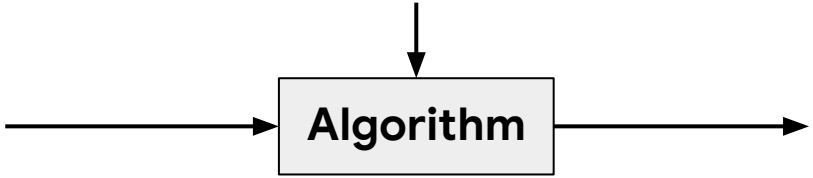
170
80
50
230

Setting II: Adversary *Cannot* Select Queries

Sensitive

name	zipcode	age	#convs
Dave	10520	40	x_1
Bob	10520	35	x_2
Carol	10500	30	x_3
Alice	10500	41	x_4

```
SELECT SUM(#convs) WHERE age >= 40
SELECT SUM(#convs) WHERE age < 40
SELECT SUM(#convs) WHERE zipcode = 10500
SELECT SUM(#convs) WHERE age != 41
```



170
80
50
230

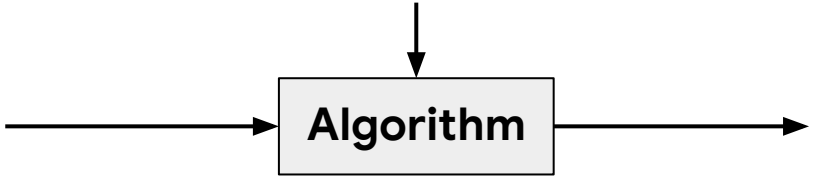
$$x_1 + x_4 = 170$$

Setting II: Adversary *Cannot* Select Queries

Sensitive

name	zipcode	age	#convs
Dave	10520	40	x_1
Bob	10520	35	x_2
Carol	10500	30	x_3
Alice	10500	41	x_4

```
SELECT SUM(#convs) WHERE age >= 40
SELECT SUM(#convs) WHERE age < 40
SELECT SUM(#convs) WHERE zipcode = 10500
SELECT SUM(#convs) WHERE age != 41
```



170
80
50
230

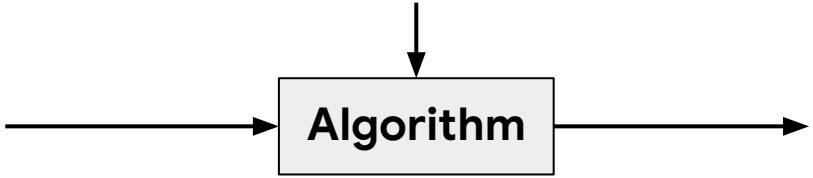
$$x_1 + x_4 = 170$$
$$x_2 + x_3 = 80$$

Setting II: Adversary *Cannot* Select Queries

Sensitive

name	zipcode	age	#convs
Dave	10520	40	x_1
Bob	10520	35	x_2
Carol	10500	30	x_3
Alice	10500	41	x_4

```
SELECT SUM(#convs) WHERE age >= 40
SELECT SUM(#convs) WHERE age < 40
SELECT SUM(#convs) WHERE zipcode = 10500
SELECT SUM(#convs) WHERE age != 41
```



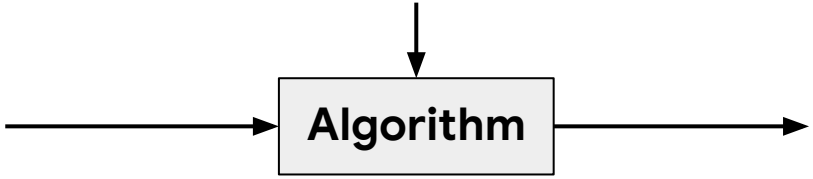
$$
\begin{aligned}
x_1 + x_4 &= 170 \\
x_2 + x_3 &= 80 \\
x_3 + x_4 &= 50 \\
x_1 + x_2 + x_3 &= 230
\end{aligned}
\quad \xrightarrow{\text{Solve}} \quad
\begin{aligned}
x_1 &= 150 \\
x_2 &= 50 \\
x_3 &= 30 \\
x_4 &= 20
\end{aligned}
$$

Setting II: Adversary *Cannot* Select Queries

Sensitive

name	zipcode	age	#convs
Dave	10520	40	x_1
Bob	10520	35	x_2
Carol	10500	30	x_3
Alice	10500	41	x_4

```
SELECT SUM(#convs) WHERE age >= 40
SELECT SUM(#convs) WHERE age < 40
SELECT SUM(#convs) WHERE zipcode = 10500
SELECT SUM(#convs) WHERE age != 41
```



173
79
51
232

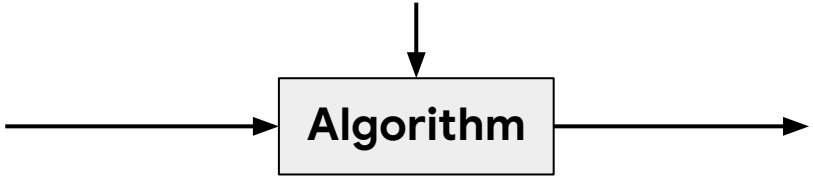
$$
\begin{aligned}
x_1 + x_4 &\approx 173 \\
x_2 + x_3 &\approx 79 \\
x_3 + x_4 &\approx 51 \\
x_1 + x_2 + x_3 &\approx 232
\end{aligned}
$$

Setting II: Adversary *Cannot* Select Queries

Sensitive

name	zipcode	age	#convs
Dave	10520	40	x_1
Bob	10520	35	x_2
Carol	10500	30	x_3
Alice	10500	41	x_4

```
SELECT SUM(#convs) WHERE age >= 40
SELECT SUM(#convs) WHERE age < 40
SELECT SUM(#convs) WHERE zipcode = 10500
SELECT SUM(#convs) WHERE age != 41
```



173
79
51
232

minimize $(x_1 + x_4 - 173)^2 + (x_2 + x_3 - 79)^2 + (x_3 + x_4 - 51)^2 + (x_1 + x_2 + x_3 - 232)^2$

subject to $x_1, x_2, x_3, x_4 \geq 0$

Solve

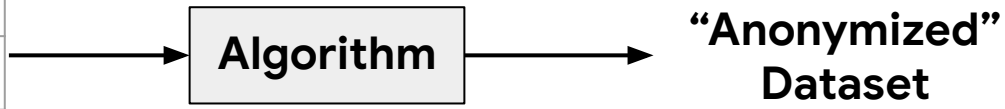
$x_1 = 153$
$x_2 = 48$
$x_3 = 31$
$x_4 = 20$

Attacks Against Anonymized Dataset

Data Anonymization

Raw dataset

name	zipcode	age	#convs
Dave	10520	40	150
Bob	10520	35	50
Carol	10500	30	30
Alice	10500	41	20



Data Anonymization

Raw dataset

name	zipcode	age	#convs
Dave	10520	40	150
Bob	10520	35	50
Carol	10500	30	30
Alice	10500	41	20

name	zipcode	age	#convs
Dave	10520	40	150
Bob	10520	35	50
Carol	10500	30	30
Alice	10500	41	20

Common techniques

- ***Suppression:*** erasing the entries

Data Anonymization

Raw dataset

name	zipcode	age	#convs
Dave	10520	40	150
Bob	10520	35	50
Carol	10500	30	30
Alice	10500	41	20

name	zipcode	age	#convs
*	10520	40	150
*	10520	35	50
*	10500	30	30
*	10500	41	20

Common techniques

- ***Suppression:*** erasing the entries

Data Anonymization

Raw dataset

name	zipcode	age	#convs
Dave	10520	40	150
Bob	10520	35	50
Carol	10500	30	30
Alice	10500	41	20

name	zipcode	age	#convs
*	10520	40	150
*	10520	35	50
*	10500	30	30
*	10500	41	20

Common techniques

- ***Suppression:*** erasing the entries
- ***Generalization:*** replacing entries with more general value

Data Anonymization

Raw dataset

name	zipcode	age	#convs
Dave	10520	40	150
Bob	10520	35	50
Carol	10500	30	30
Alice	10500	41	20

Is this privacy-preserving?

Anonymized dataset

name	zipcode	age	#convs
*	10520	≥40	150
*	10520	<40	50
*	10500	<40	30
*	10500	≥40	20

Common techniques

- ***Suppression:*** erasing the entries
- ***Generalization:*** replacing entries with more general value

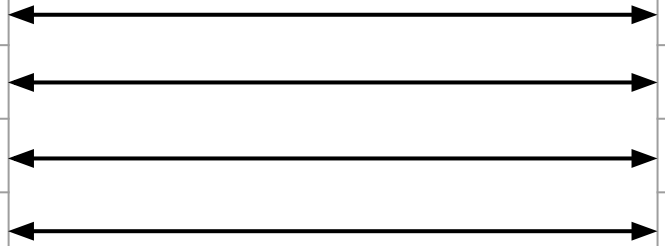
Linkage Attacks

Adversary's Auxiliary data

name	zipcode	age
Dave	10520	40
Bob	10520	35
Carol	10500	30
Alice	10500	41

Anonymized dataset

name	zipcode	age	#convs
*	10520	≥40	150
*	10520	<40	50
*	10500	<40	30
*	10500	≥40	20



Linkage Attack (aka De-Anonymization Attack)
Adversary links record of anonymized dataset to auxiliary dataset

Linkage Attacks

Adversary computes
maximum matching

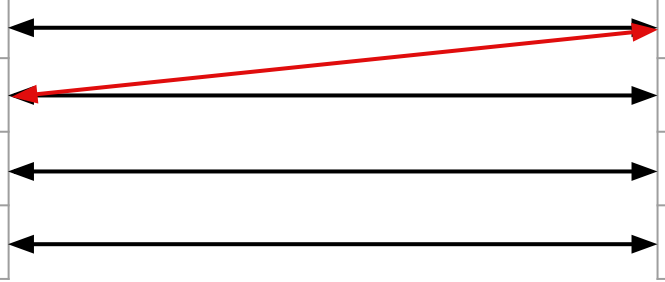
Compatibility Graph

Adversary's Auxiliary data

name	zipcode	age
Dave	10520	40
Bob	10520	35
Carol	10500	30
Alice	10500	41

Anonymized dataset

name	zipcode	age	#convs
*	10520	*	150
*	10520	<40	50
*	10500	<40	30
*	10500	≥40	20



Linkage Attack (aka De-Anonymization Attack)
Adversary links record of anonymized dataset to auxiliary dataset

Linkage Attacks

Can handle uncertain data with weights & min-weight matching

Adversary computes **maximum matching**

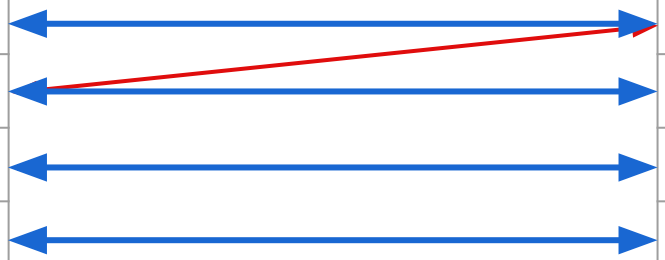
Compatibility Graph

Adversary's Auxiliary data

name	zipcode	age
Dave	10520	40
Bob	10520	35
Carol	10500	30
Alice	10500	41

Anonymized dataset

name	zipcode	age	#convs
*	10520	*	150
*	10520	<40	50
*	10500	<40	30
*	10500	≥40	20



Linkage Attack (aka De-Anonymization Attack)

Adversary links record of anonymized dataset to auxiliary dataset

Linkage Attacks: Practical Examples

Example I: Massachusetts Health Data

Medical data of state employees in the state of Massachusetts

- Personal identifiers (eg, name, SSN) are removed

[Sweeney] managed to identify the governor's medical records

Example II: Netflix Prize

"Anonymized" user ratings

- Each row is (user, movie, rating, time)
- User id is re-randomized

99% re-identified by *[Narayanan, Shmatikov'08]*

Example III: EdX Dataset

5-anonymized student data

- Each row has demographics, course activities/outcomes
- Anonymization using subset of attributes

Using LinkedIn side information, a few students re-identified *[Cohen'22]*

k-Anonymity

Known | Unknown

name	zipcode	age	#convs
Dave	10520	40	150
Bob	10520	35	50
Carol	10500	30	30
Alice	10500	41	20

No knowledge of unknown columns
⇒ cannot distinguish each row from
$k-1$ other rows

Algorithm

2-Anonymity

Known | Unknown

name	zipcode	age	#convs
*	10520	*	150
*	10520	*	50
*	10500	*	30
*	10500	*	20

Limitations

- Requires adversary to know *nothing* about unknown columns
- k-Anonymity does not hold if consider two releases

k-Anonymity
Every combination of known values appear at least k times

k-Anonymity

Known | Unknown

name	zipcode	age	#convs
Dave	10520	40	150
Bob	10520	35	50
Carol	10500	30	30
Alice	10500	41	20

No knowledge of unknown columns
⇒ cannot distinguish each row from
$k-1$ other rows

Algorithm

name	zipcode	age	#convs
*	10520	*	150
*	10520	*	50
*	10500	*	30
*	10500	*	20

name	zipcode	age	#convs
*	*	≥40	150
*	*	<40	50
*	*	<40	30
*	*	≥40	20

Limitations

- Requires adversary to know *nothing* about unknown columns
- k-Anonymity does not hold if consider two releases

k-Anonymity
Every combination of known values appear at least k times

k-Anonymity

Known | Unknown

name	zipcode	age	#convs
Dave	10520	40	150
Bob	10520	35	50
Carol	10500	30	30
Alice	10500	41	20

No knowledge of unknown columns
⇒ cannot distinguish each row from
$k-1$ other rows

Algorithm

name	zipcode	age	#convs
*	10520	*	150
*	10520	*	50
*	10500	*	30
*	10500	*	20

name	zipcode	age	#convs
*	*	≥40	150
*	*	<40	50
*	*	<40	30
*	*	≥40	20

Limitations

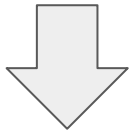
- Requires adversary to know *nothing* about unknown columns
- k-Anonymity does not hold if consider two releases
- Computing "optimal" anonymized dataset is NP-hard

k-Anonymity
Every combination of known values appear at least k times

What is *not* a privacy violation?

Inference ≠ Privacy Violation

Study in 1950s



“Heavy Smoking Cause Cancer”

Public | Sensitive

name	Heavy smoker	age	Has Cancer
Dave	NO	40	-
Bob	YES	60	-
Carol	NO	55	-
Alice	NO	41	-

Inference ≠ Privacy Violation

Did not rely on Bob's data at all

↓

Cannot be privacy violation

Study in 1950s

↓

"Heavy Smoking Cause Cancer"

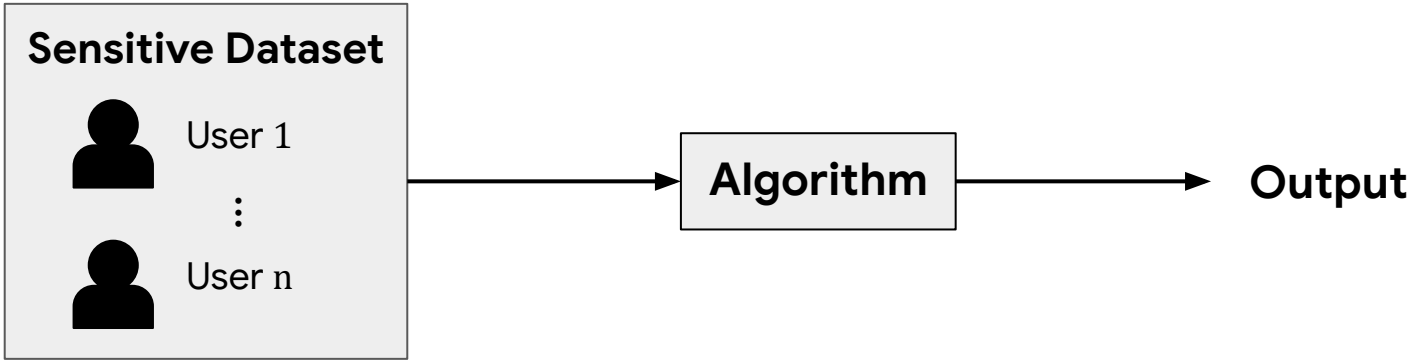
Public | Sensitive

name	Heavy smoker	age	Has Cancer
Dave	NO	40	-
Bob	YES	60	YES
Carol	NO	55	-
Alice	NO	41	-

Adversary correctly guesses that Bob has cancer

Conclusion of the study affects Bob not his presence or absence in the dataset

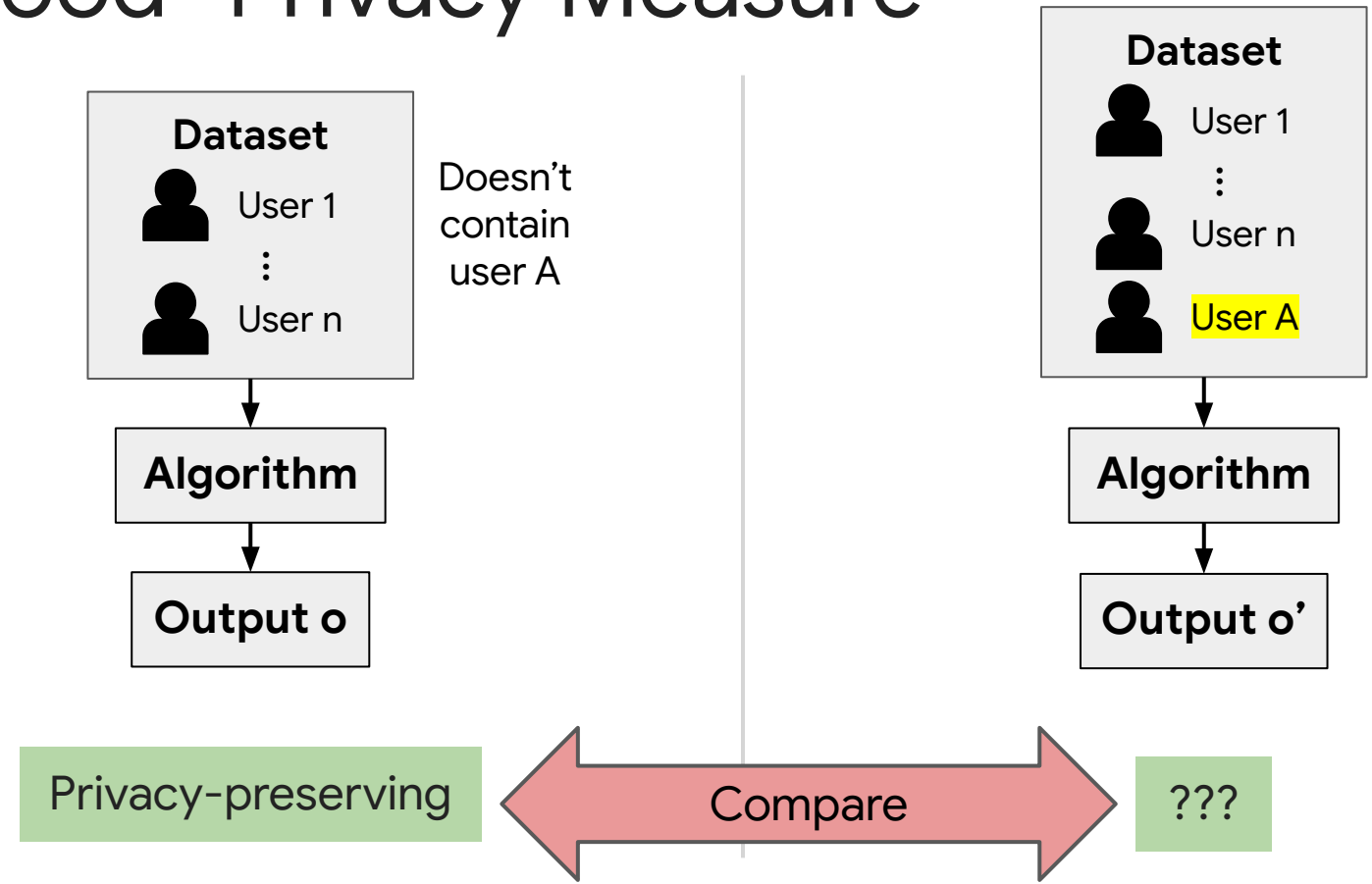
Inference ≠ Privacy Violation



- If a user chooses not to be in the dataset, then need to be more careful
- Can model as user is in the dataset, but with a column saying “OPT OUT”

If a user’s data is not in the input,
⇒ releasing the output does ***not*** violate the user’s privacy

“Good” Privacy Measure



Differential Privacy

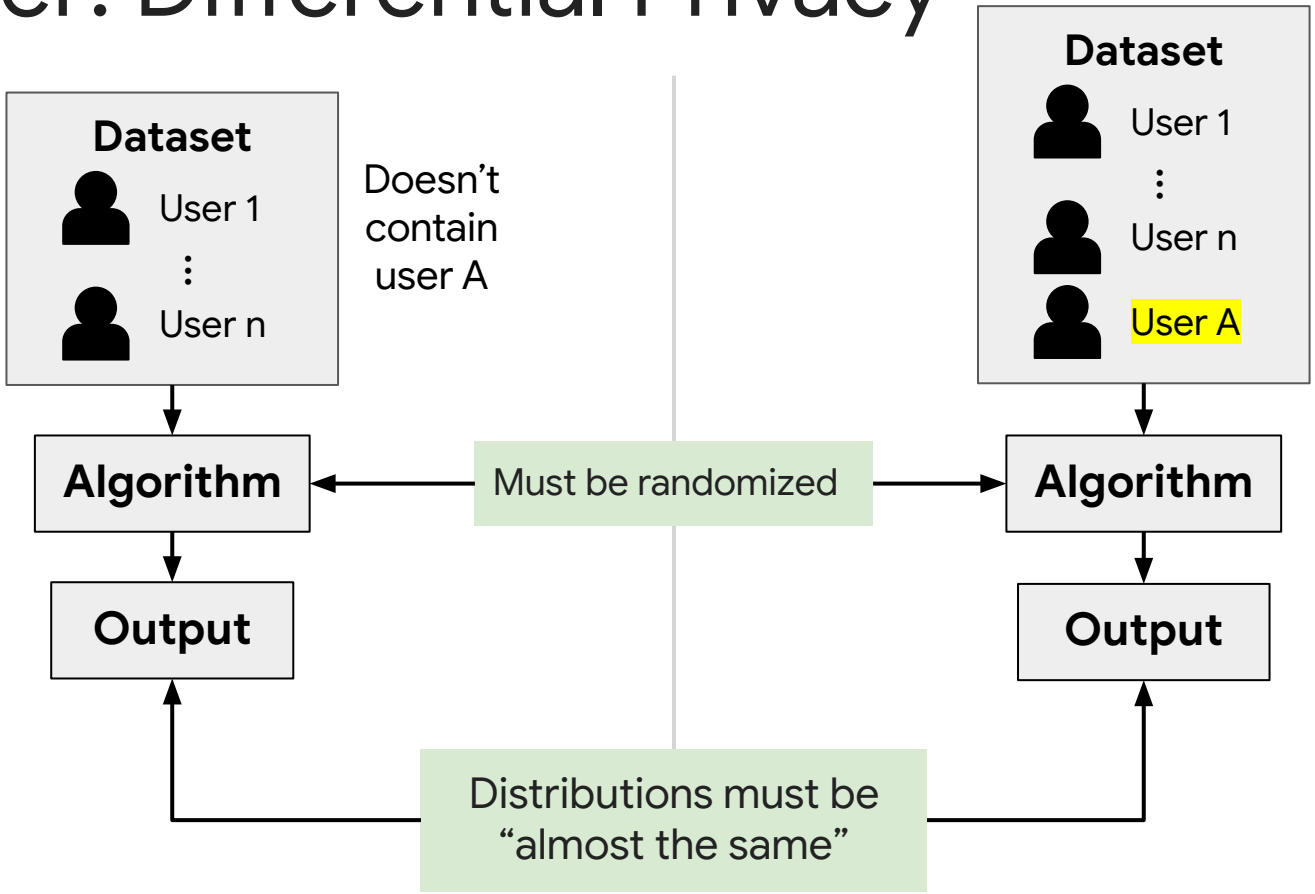
Why Differential Privacy (DP)?

- DP is only privacy notion that is robust (unlike k-anonymity) and has intuitive, practically appealing properties
  - DP give rigourous guarantees that an individual user will not be adversely affected by allowing their data to be used
  - Learns little about an individual but useful things about a population
- Other notions of security/privacy (eg, FL or MPC) have to be combined with DP to provide anonymity guarantees

- DP is the only *anonymity* notion broadly adopted by multiple organizations
  - Google, Apple, Microsoft, Meta, LinkedIn, Uber, Amazon
  - US Census Bureau
  - Academia
- Privacy research is dominated by DP
  - Eg, NeurIPS'22 had 50 papers with "priv" in title and 49 were on DP

Differential Privacy: Basics

Teaser: Differential Privacy



Differential Privacy: Definition

Intuition:

Adding or removing a single user should not change the output distribution too much

Smaller ε, δ

⇕

More privacy

(ε, δ)-Differential Privacy

[Dwork et al.'06]

For every neighboring datasets $\mathbf{X}, \mathbf{X'}$ and every set $\mathbf{S}$ of outputs,

$$\Pr[M(X) \in S] \leq e^{\varepsilon} \cdot \Pr[M(X') \in S] + \delta$$

Pure-DP

ε-DP ≡ $(\varepsilon, 0)$-DP

Approx-DP

$\delta > 0$

A property not an algorithm

ε = small constant

δ = negligible in #users

Privacy budget

Differential Privacy: Definition

Intuition:

Adding or removing a single user should not change the output distribution too much

Smaller ε, δ

⇕

More privacy

(ε, δ)-Differential Privacy

[Dwork et al.'06]

For every neighboring datasets $\mathbf{X}, \mathbf{X'}$ and every set $\mathbf{S}$ of outputs,

$$\Pr[M(X) \in S] \leq e^{\varepsilon} \cdot \Pr[M(X') \in S] + \delta$$

Pure-DP

ε-DP $\equiv$ $(\varepsilon, 0)$-DP

Approx-DP

$\delta > 0$

- "neighboring" notion can be broader than intuition
- $\asymp$: abbrv for neighboring

ε = small constant

δ = negligible in #users

Differential Privacy: Neighboring Notions

User-Level DP

$X \asymp^u X'$ ⟺ X is X' with an individual's data added / removed

Dataset X

	name	zipcode	age	Purchase value
x_1	Dave	10520	40	150
x_2	Bob	10520	35	50
x_3	Carol	10500	30	30
	Carol	10500	30	70

Dataset X'

	name	zipcode	age	#convs
x_1	Dave	10520	40	150
x_2	Bob	10520	35	50

Differential Privacy: Neighboring Notions

Event-Level DP

$X \simeq^e X'$ ⟺ X is X' with a single event is added / removed

Dataset X

	name	zipcode	age	Purchase value
x_1	Dave	10520	40	150
x_2	Bob	10520	35	50
x_3	Carol	10500	30	30
x_4	Carol	10500	30	70

Dataset X

	name	zipcode	age	Conv Value
x_1	Dave	10520	40	150
x_2	Bob	10520	35	50
x_3	Carol	10500	30	30

Differential Privacy: Neighboring Notions

Add/Remove-DP

$X \simeq^r X'$ ⟺ X is X' with an individual's data added / removed

Dataset X

	name	zipcode	age	#convs
x_1	Dave	10520	40	150
x_2	Bob	10520	35	50
x_3	Carol	10500	30	30

Dataset X'

	name	zipcode	age	#convs
x_1	Dave	10520	40	150
x_2	Bob	10520	35	50

Differential Privacy: Neighboring Notions

Substitution-DP

$X \asymp^s X'$ ⟺ X is X' with an individual's data changed

Dataset X

	name	zipcode	age	#convs
x_1	Dave	10520	40	150
x_2	Bob	10520	35	50
x_3	Carol	10500	30	30

Dataset X'

	name	zipcode	age	#convs
x_1	Dave	10520	40	150
x_2	Bob	10520	35	50
x_3	Carol	10520	32	150

Differential Privacy: Neighboring Notions

Add/Remove-DP

$X \asymp^r X'$ ⟺ X is X' with an individual's data added / removed

Substitution-DP

$X \asymp^s X'$ ⟺ X is X' with an individual's data changed

These notions are mostly interchangeable up to parameters

Unless otherwise stated, assume
substitution-DP

Pure vs Approximate DP

(ε, δ)-Differential Privacy

[Dwork et al.'06]

For every neighboring datasets X, X' and every set S of outputs,

$$\Pr[M(X) \in S] \leq e^{\varepsilon} \cdot \Pr[M(X') \in S] + \delta$$

Pure-DP
$\delta = 0$

- Desirable
- Worst privacy-utility trade-off for many problems
- Weaker composability

Approx-DP
$\delta > 0$

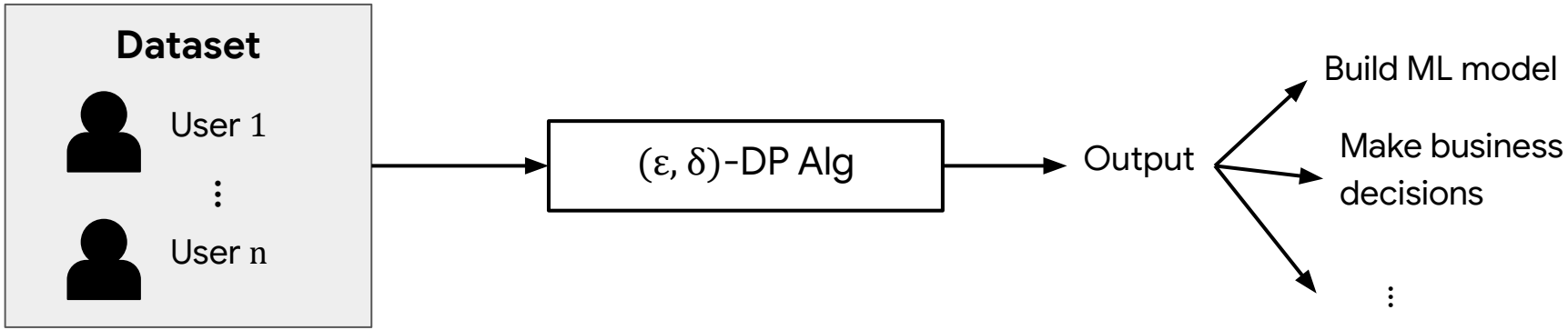
- Definition allows catastrophic leak
- δ ideally $o(1/n)$
- Most practical mechanisms behave gracefully

DP Properties

Post-Processing

"Result of DP algorithm can be used in arbitrary manner and it remains DP."

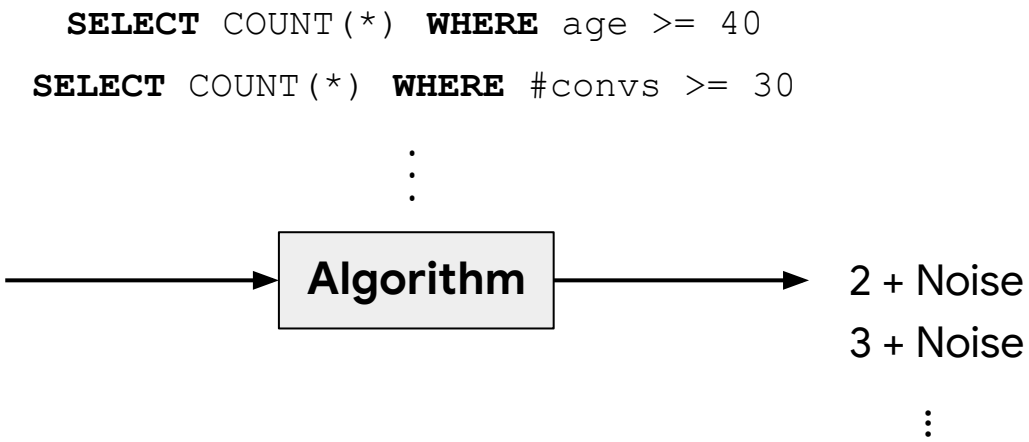
Theorem If M is (ε, δ)-DP and $h(\cdot)$ is any function, then $h(M(X))$ remains (ε, δ)-DP.



Can we use the output safely in downstream applications?

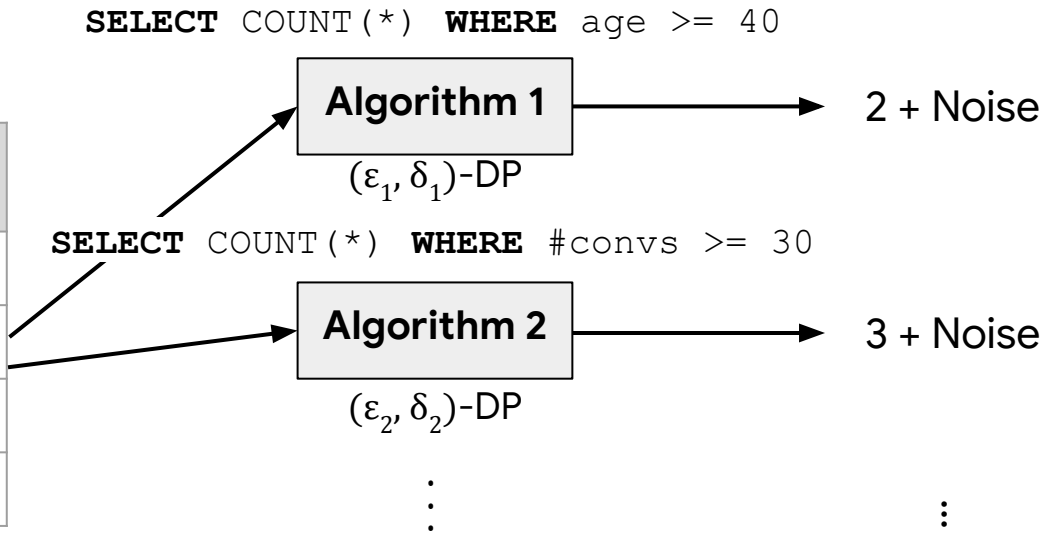
Composition

name	zipcode	age	#convs
Dave	10520	40	150
Bob	10520	35	50
Carol	10500	30	30
Alice	10500	41	20

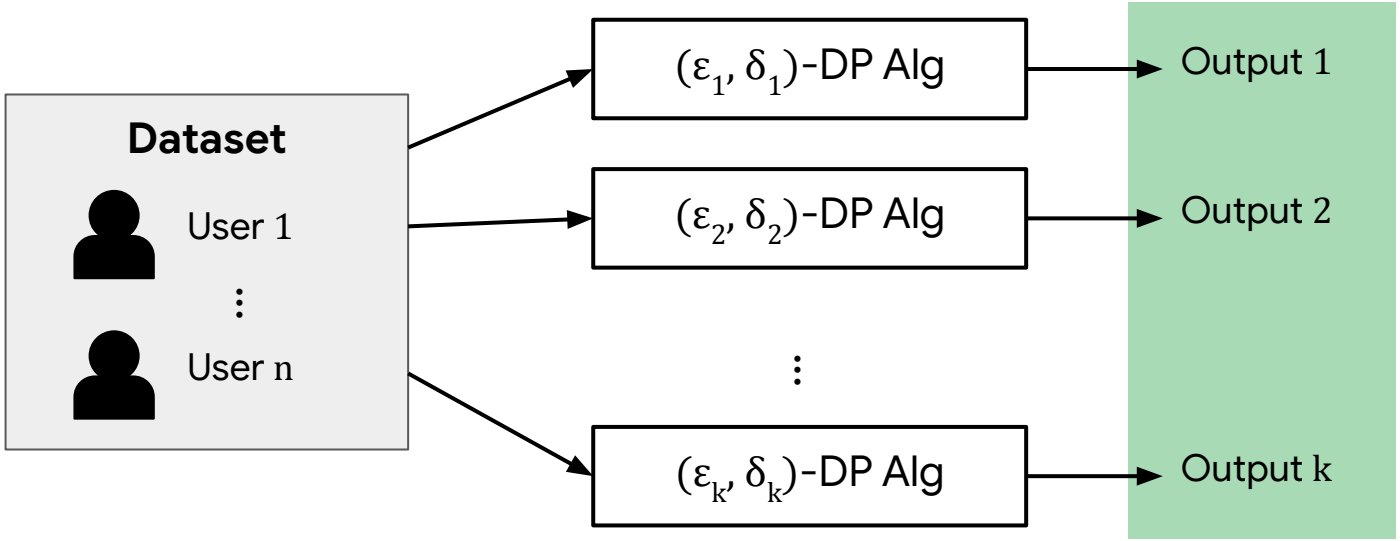


Composition

name	zipcode	age	#convs
Dave	10520	40	150
Bob	10520	35	50
Carol	10500	30	30
Alice	10500	41	20



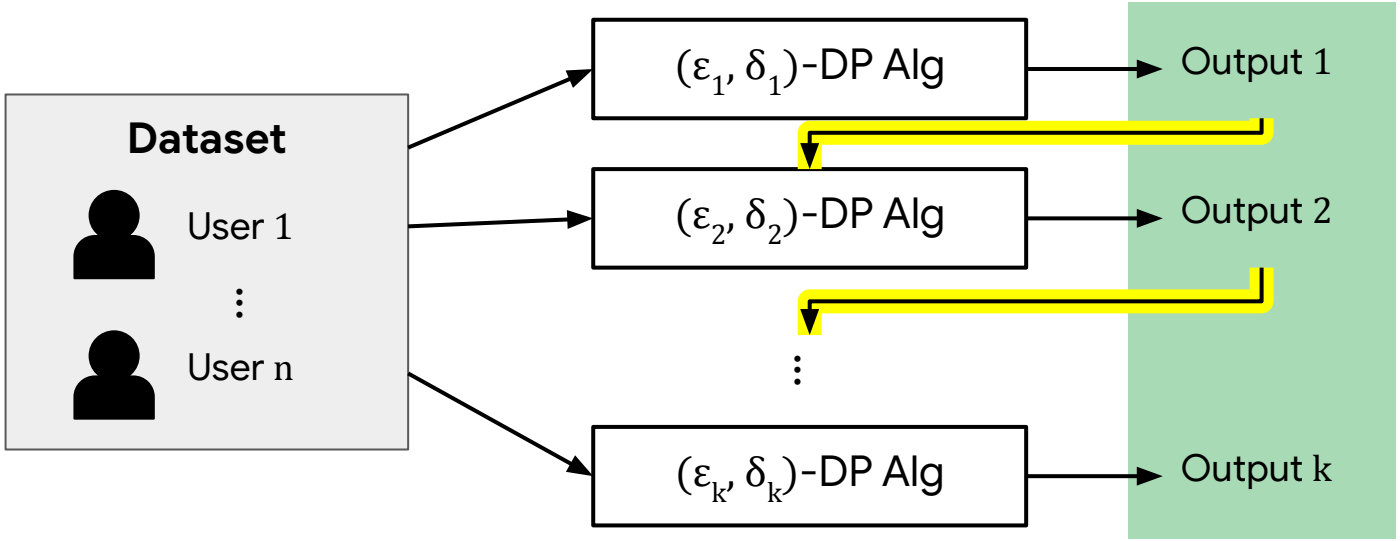
Composition



Helps modular algorithm design

"Running DP algorithms multiple times remain DP, *but with worse parameters*"

Basic Composition



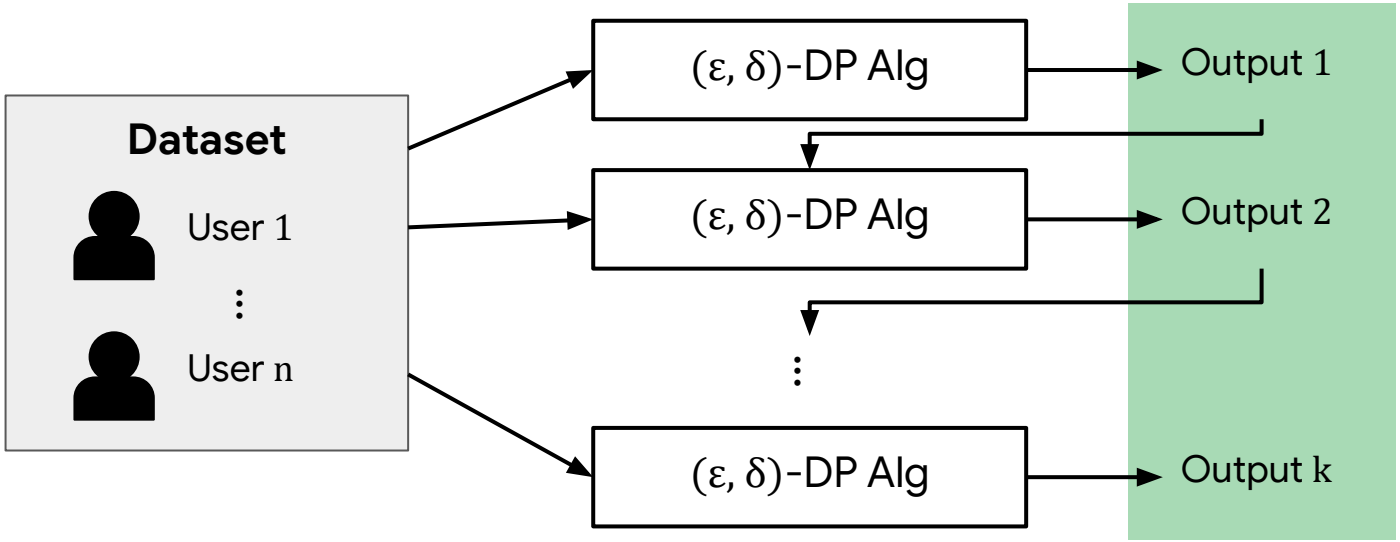
"Adaptive" setting: previous outputs are used in subsequent algorithms

Basic Composition Theorem *[Dwork et al.]*

All the outputs combined remain $(\varepsilon_1 + \cdots + \varepsilon_k, \delta_1 + \cdots + \delta_k)$-DP

Works even for "adaptive" setting

Advanced Composition



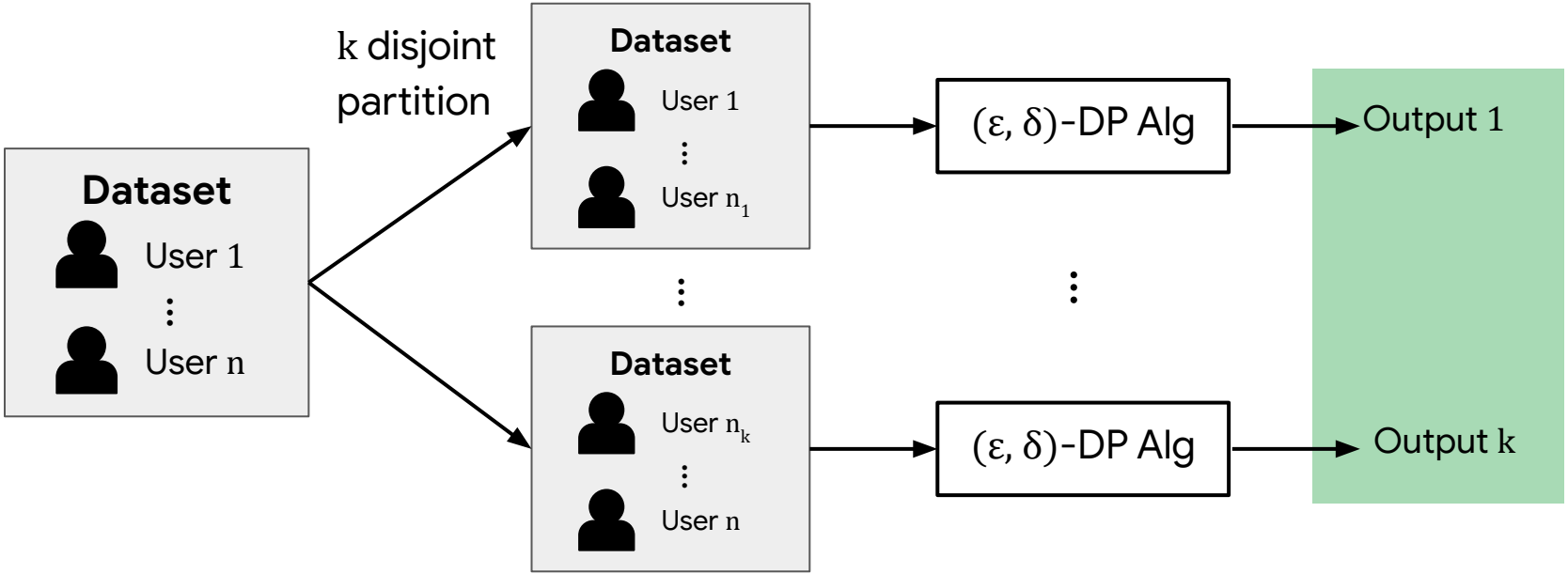
Advanced Composition Theorem *[Dwork et al.]*

All the outputs combined is $(\varepsilon', k\delta + \delta')$-DP where

$$\varepsilon' = \sqrt{2k \ln(1/\delta')} \cdot \varepsilon + k\,\varepsilon\,(e^{\varepsilon} - 1)$$

- For small ε and large k, ε' is only $\approx \varepsilon\sqrt{k}$
- Better than $\varepsilon' = \varepsilon k$ from Basic composition

Parallel Composition



Parallel Composition Theorem *[McSherry]*
All the outputs combined remain (ε, δ)-DP

Group Privacy

Example:

Email	zipcode	Purchase value
dave123@gmail.com	10520	500
dave1993@gmail.com	10520	25
alice@gmail.com	10500	15
bob@gmail.com	10500	40

Group Privacy

k-Neighboring

X, X' are k-neighboring, denoted by $\asymp_k$ iff there exists a sequence $X = X_0 \asymp X_1 \asymp \cdots \asymp X_t = X', t \leq k$.

Theorem If M is (ε, δ)-DP under neighboring notion $\asymp$, then $M(X)$ is (ε', δ')-DP for neighboring notion $\asymp_k$, where $\varepsilon' = k\varepsilon$ and $\delta' = \delta \cdot (e^{k\varepsilon} - 1) / (e^{\varepsilon} - 1)$.

Example:

Same individual

Email	zipcode	Purchase value
dave123@gmail.com	10520	500
dave1993@gmail.com	10520	25
alice@gmail.com	10500	15
bob@gmail.com	10500	40

Is this still DP? ⇨ Yes, it is 2ε-DP

User-level vs event-level

Group Privacy

k-Neighboring

X, X' are k-neighboring, denoted by $\asymp_k$ iff there exists a sequence $X = X_0 \asymp X_1 \asymp \cdots \asymp X_t = X', t \leq k.$

Theorem If M is (ε, δ)-DP under neighboring notion $\asymp$, then $M(X)$ is (ε', δ')-DP for neighboring notion $\asymp_k$, where $\varepsilon' = k\varepsilon$ and $\delta' = \delta \cdot (e^{k\varepsilon} - 1) / (e^{\varepsilon} - 1)$.

Example:

Add/remove $X \asymp^r X'$ ⟺ X is X' with an individual's data added / removed

Substitution $X \asymp^s X'$ ⟺ X is X' with an individual's data changed

Observation

If $X \asymp^s X'$, then $X \asymp^r_2 X'$

"Every substitution neighbor is a 2-add/remove neighbor"

Group Privacy

k-Neighboring

X, X' are k-neighboring, denoted by $\asymp_k$ iff there exists a sequence $X = X_0 \asymp X_1 \asymp \cdots \asymp X_t = X', t \leq k$.

Theorem If M is (ε, δ)-DP under neighboring notion $\asymp$, then $M(X)$ is (ε', δ')-DP for neighboring notion $\asymp_k$, where $\varepsilon' = k\varepsilon$ and $\delta' = \delta \cdot (e^{k\varepsilon} - 1) / (e^{\varepsilon} - 1)$.

Example:

Lemma

If M is ε-add/remove-DP, then it is 2ε-substitution-DP

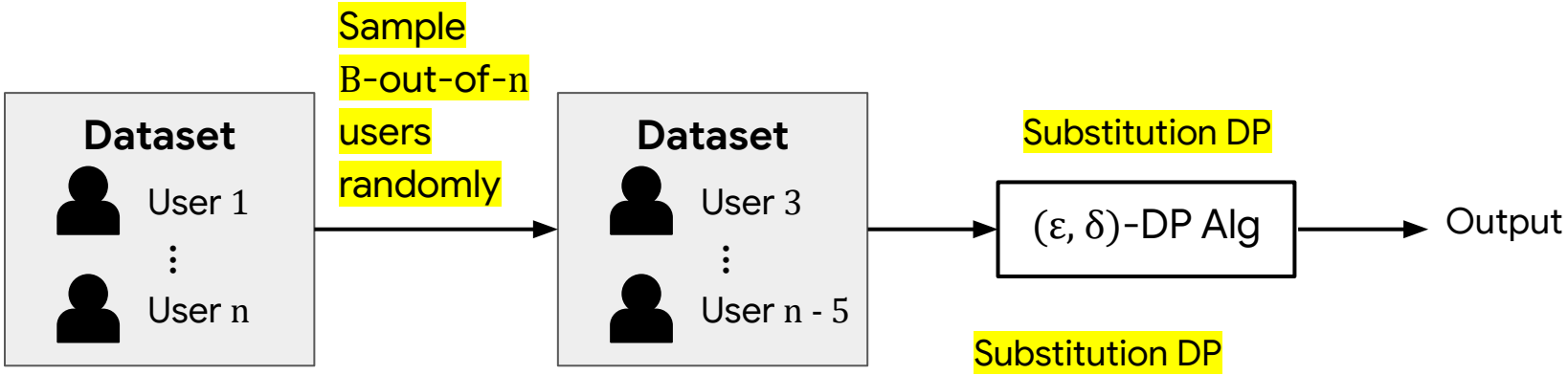
Observation

If $X \asymp^s X'$, then $X \asymp^r_2 X'$

"Every substitution neighbor is a 2-add/remove neighbor"

Amplification by Subsampling

"Subsampling makes the algorithm more private."



Amplification-by-subsampling Theorem

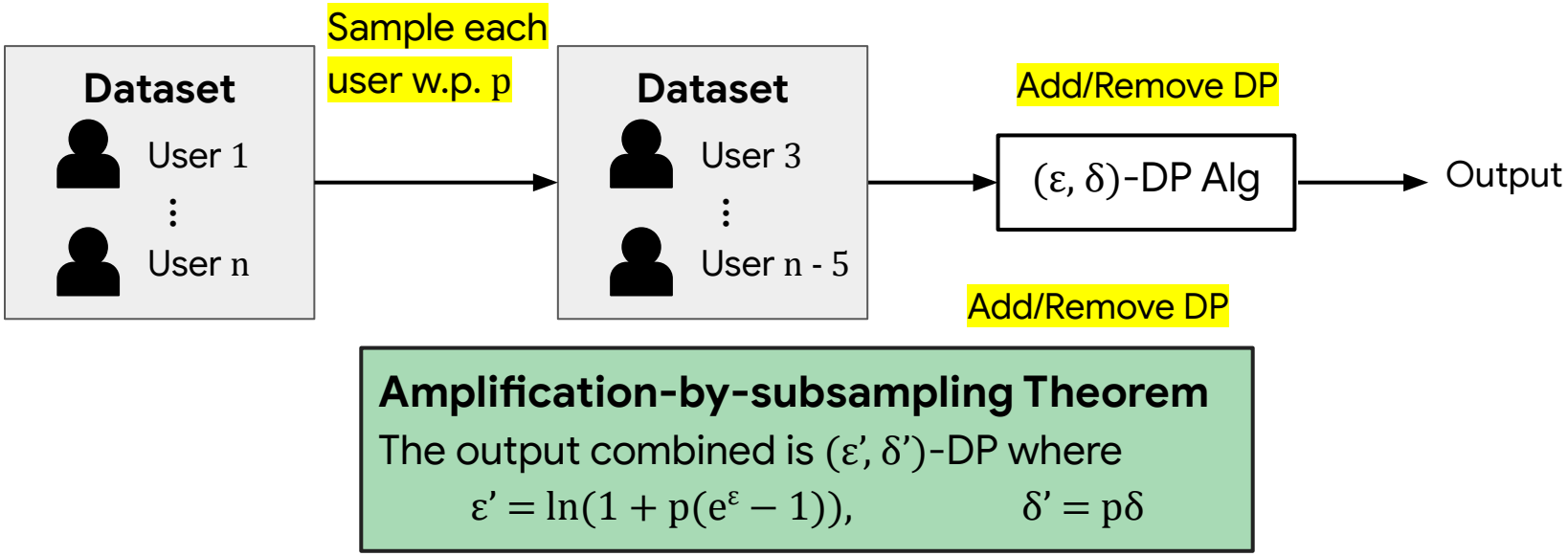
The output combined is (ε', δ')-DP where

$$\varepsilon' = \ln(1 + p(e^{\varepsilon} - 1)), \qquad \delta' = p\delta$$

with $p = B / n$

Amplification by Subsampling

"Subsampling makes the algorithm more private."



Amplification-by-subsampling Theorem

The output combined is (ε', δ')-DP where

$$\varepsilon' = \ln(1 + p(e^{\varepsilon} - 1)), \qquad \delta' = p\delta$$

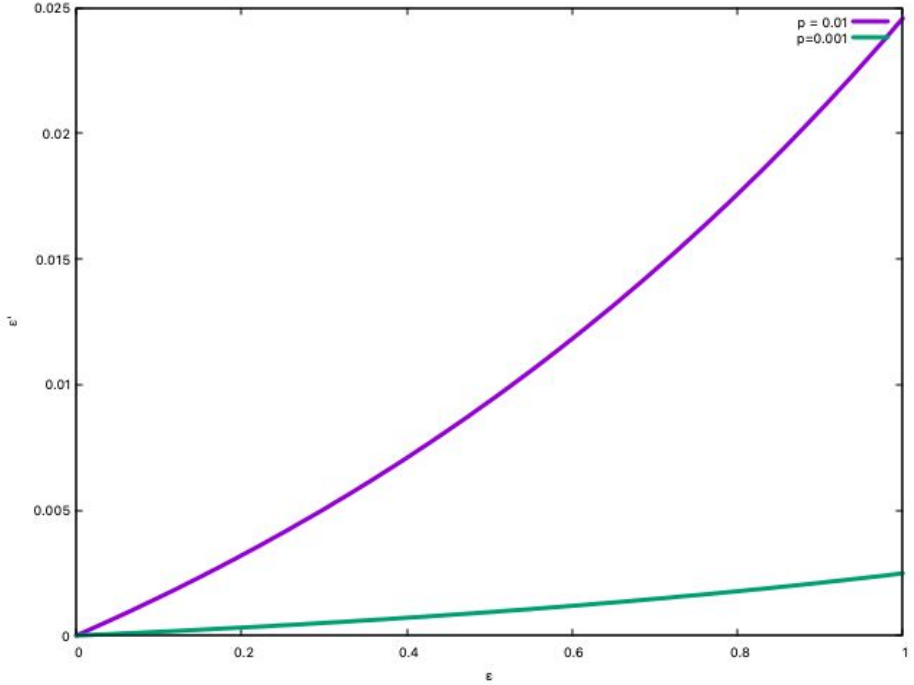
Amplification by Subsampling

"Subsampling makes the algorithm more private."

Amplification-by-subsampling Thm
The output combined is (ε', δ')-DP where

$$\varepsilon' = \ln(1 + p(e^{\varepsilon} - 1)), \qquad \delta' = p\delta$$

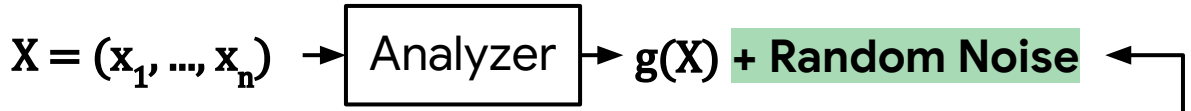
with $p = B / n$



Basic Mechanism: Noise Addition

Noise Addition Mechanism

Noise Addition Mechanism

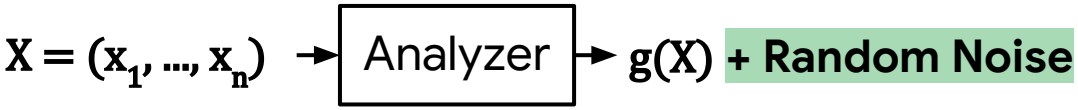


Intuition:
Noise should be large enough to hide a user's contribution

What noise distribution should we use?

- Depends on $\mathrm{Range}(g)$
- Depends on how "sensitive" g is

Discrete Laplace Mechanism



Assumption: $\text{Range}(g) \subseteq \mathbb{Z}$

(ie, g is integer-valued)

Sensitivity

$\Delta(g) = \max_{X \asymp X'} |g(X) - g(X')|$

Larger sensitivity ⇒
More noise required

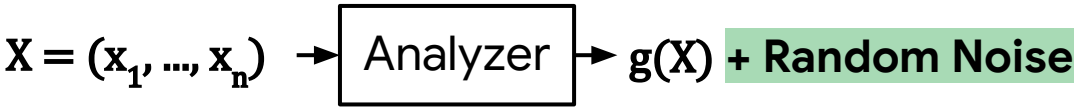
Examples: COUNT Query

$$g = \text{COUNT}(*) \text{ WHERE } \#\text{convs} > 40$$

	name	zipcode	age	#convs
x_1	Dave	10520	40	150
x_2	Bob	10520	35	50
x_3	Carol	10500	30	30

$g(X) = 2$

Discrete Laplace Mechanism



Assumption: $\text{Range}(g) \subseteq \mathbb{Z}$

(ie, g is integer-valued)

Sensitivity

$\Delta(g) = \max_{X \asymp X'} |g(X) - g(X')|$

Larger sensitivity ⇒
More noise required

Examples: COUNT Query

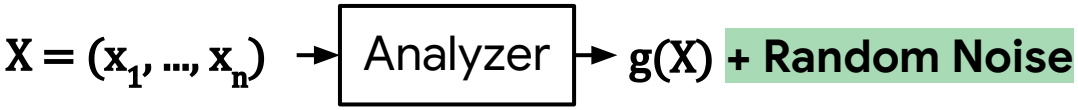
$$g = \text{COUNT}(*) \text{ WHERE } \#\text{convs} > 40$$

	name	zipcode	age	#convs
x_1	Dave	10520	40	150
x_2	Bob	10520	35	50
x_3	Carol	10500	30	60

g(X) = 3

$\Delta(g) = 1$

Discrete Laplace Mechanism



Assumption: $\text{Range}(g) \subseteq \mathbb{Z}$

(ie, g is integer-valued)

Sensitivity

$\Delta(g) = \max_{X \asymp X'} |g(X) - g(X')|$

Larger sensitivity ⇒
More noise required

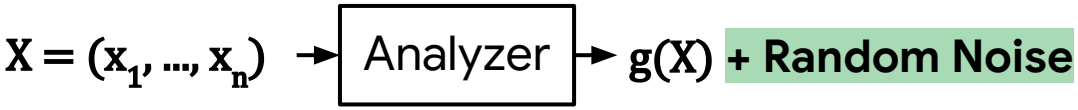
Examples: SUM Query

$$g = \text{SUM}(\#\text{convs})$$

	name	zipcode	age	#convs
x_1	Dave	10520	40	150
x_2	Bob	10520	35	50
x_3	Carol	10500	30	60

$g(X) = 260$

Discrete Laplace Mechanism



Assumption: $\text{Range}(g) \subseteq \mathbb{Z}$

(ie, g is integer-valued)

Sensitivity

$\Delta(g) = \max_{X \simeq X'} |g(X) - g(X')|$

Larger sensitivity ⇒
More noise required

Examples: SUM Query

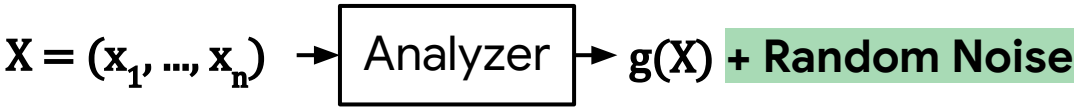
$$g = \text{SUM}(\#\text{convs})$$

	name	zipcode	age	#convs
x_1	Dave	10520	40	150
x_2	Bob	10520	35	50
x_3	Carol	10500	30	190

$g(X) = 390$

Assume $\#\text{convs} \leq c$ ⇒ $\Delta(g) \leq c$

Discrete Laplace Mechanism



Assumption: $\text{Range}(g) \subseteq \mathbb{Z}$

(ie, g is integer-valued)

Sensitivity

$\Delta(g) = \max_{X \asymp X'} |g(X) - g(X')|$

Larger sensitivity ⇒
More noise required

Examples: AVERAGE Query

$$g = \text{AVG}(\#\text{convs})$$

	name	zipcode	age	#convs
x_1	Dave	10520	40	150
x_2	Bob	10520	35	50
x_3	Carol	10500	30	190

$g(X) = 130$

Assume $\#\text{convs} \leq c$ ⇒ $\Delta(g) \leq c/n$

Discrete Laplace Mechanism

$X = (x_1, ..., x_n)$ → Analyzer → $g(X)$ + **Random Noise**

Assumption: $\text{Range}(g) \subseteq \mathbb{Z}$

(ie, g is integer-valued)

Sensitivity

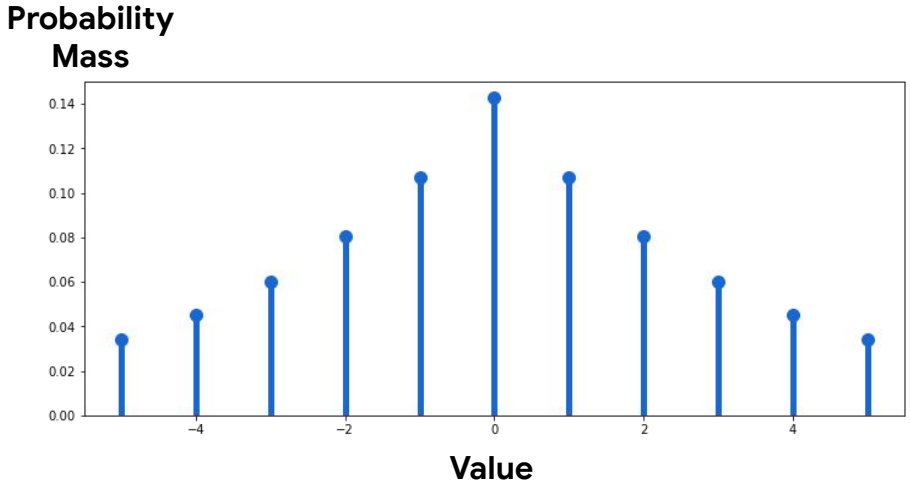
$\Delta(g) = \max_{X \asymp X'} |g(X) - g(X')|$

Larger sensitivity ⇒
More noise required

Discrete Laplace Distribution

For every integer i,

$$\Pr[i = \text{DLap}(b)] \propto e^{-|i|/b}$$



Discrete Laplace Mechanism

$$X = (x_1, \ldots, x_n) \rightarrow \boxed{\text{Analyzer}} \rightarrow g(X) + \mathbf{DLap}(\Delta(g)/\varepsilon)$$

Assumption: $\text{Range}(g) \subseteq \mathbb{Z}$

(ie, g is integer-valued)

Sensitivity

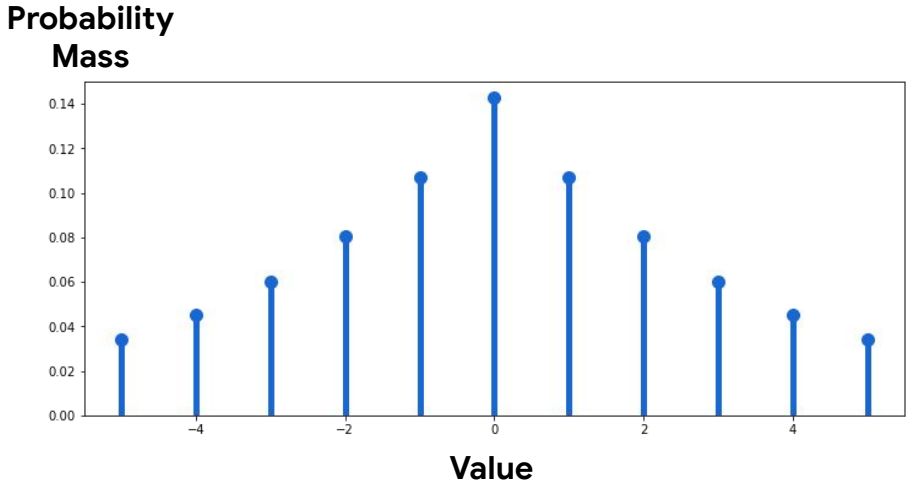
$\Delta(g) = \max_{X \asymp X'} |g(X) - g(X')|$

Larger sensitivity ⇒
More noise required

Discrete Laplace Distribution

For every integer i,

$$\Pr[i = \mathbf{DLap}(b)] \propto e^{-|i|/b}$$



Discrete Laplace Mechanism

$$X = (x_1, \ldots, x_n) \rightarrow \boxed{\text{Analyzer}} \rightarrow g(X) + \text{DLap}(\Delta(g)/\varepsilon)$$

Example:

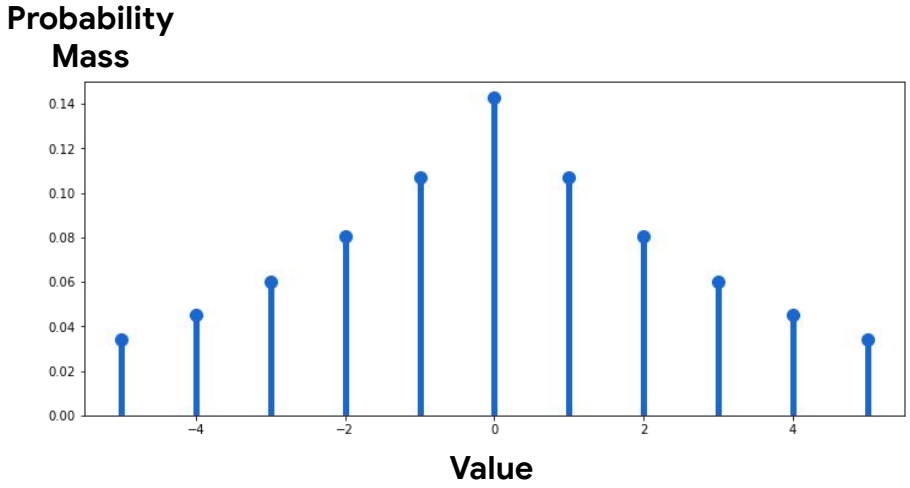
$\Delta(g) = 1, \quad \varepsilon = \ln(4/3), \quad g(X) = 3$

Output: $3 + \text{DLap}(1/\ln(4/3))$

Discrete Laplace Distribution

For every integer i,

$$\Pr[i = \text{DLap}(b)] \propto e^{-|i|/b}$$



Discrete Laplace Mechanism

$$X = (x_1, \ldots, x_n) \rightarrow \boxed{\text{Analyzer}} \rightarrow g(X) + \text{DLap}(\Delta(g)/\varepsilon)$$

Example:

$\Delta(g) = 1, \quad \varepsilon = \ln(4/3), \quad g(X) = 3$

Output: $3 + \text{DLap}(1/\ln(4/3))$

output =

$\vdots$

2 w.p. 3/28 = 0.1071...

3 w.p. 1/7 = 0.1428...

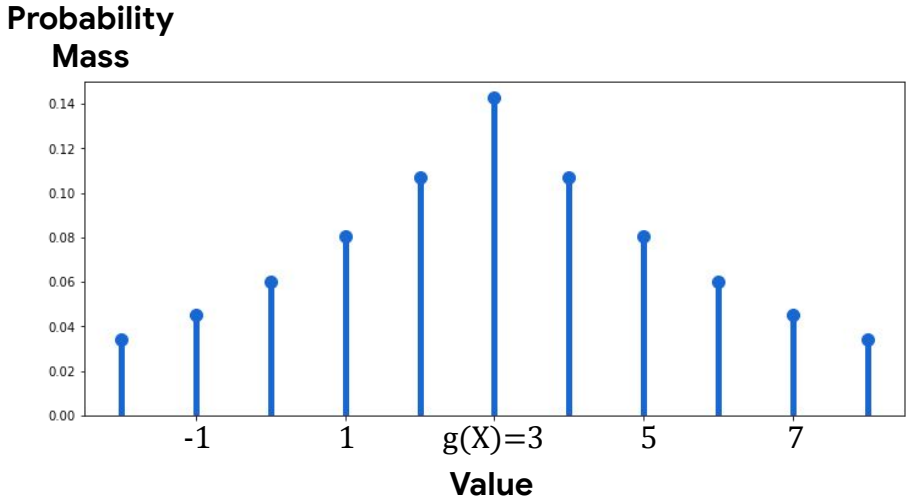
4 w.p. 3/28 = 0.1071...

$\vdots$

Discrete Laplace Distribution

For every integer i,

$$\Pr[i = \text{DLap}(b)] \propto e^{-|i|/b}$$

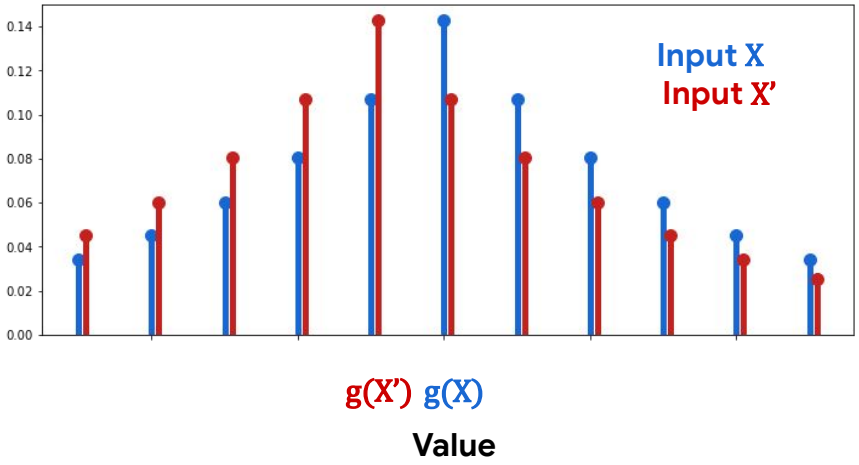


Discrete Laplace Mechanism: Privacy

$$X = (x_1, \ldots, x_n) \rightarrow \boxed{\text{Analyzer}} \rightarrow g(X) + \mathbf{DLap(\Delta(g)/\varepsilon)}$$

Theorem Assuming $\mathrm{Range}(g) \subseteq \mathbb{Z}$, Discrete Laplace Mechanism is ε-DP.

Illustration for $\Delta(g) = 1$:



Discrete Laplace Mechanism: Utility

$X = (x_1, ..., x_n)$ → Analyzer → $g(X) + DLap(\Delta(g)/\varepsilon)$

Theorem Discrete Laplace Mechanism has MSE = $2p / (1-p)^2$ where $p = \exp(-\varepsilon/\Delta(g))$

Utility Measures

- ***Mean Square Error (MSE)***: $\mathbb{E}[(\text{output} - \text{true})^2]$
  - ***Root Mean Square Error (RMSE)***: $\sqrt{\text{MSE}}$
- ***Mean Absolute Error (MAE)***: $\mathbb{E}[|\text{output} - \text{true}|]$

- We will mostly compute MSE since it is easier to deal with
- MAE is always ≤ RMSE

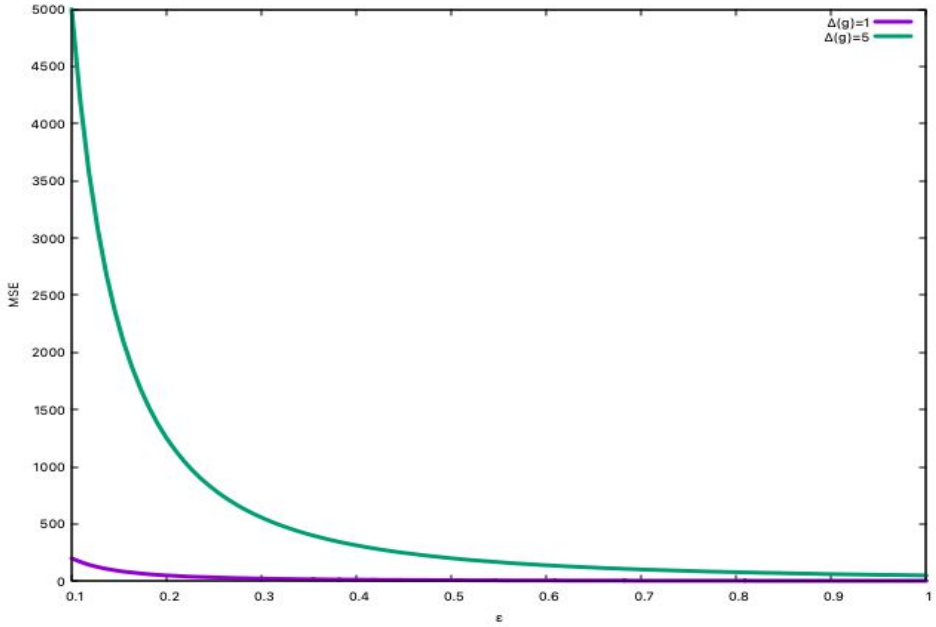
Discrete Laplace Mechanism: Utility

Theorem Discrete Laplace Mechanism has MSE = $2p / (1 - p)^2$ where $p = \exp(-\varepsilon/\Delta(g))$

Discrete Laplace Mechanism: Utility

$$X = (x_1, \ldots, x_n) \rightarrow \boxed{\text{Analyzer}} \rightarrow g(X) + \mathbf{DLap}(\Delta(g)/\varepsilon)$$

Theorem Discrete Laplace Mechanism has MSE = $2p / (1-p)^2$ where $p = \exp(-\varepsilon/\Delta(g))$



$O((\Delta(g)/\varepsilon)^2)$ for $\varepsilon \leq 1$

Discrete Laplace Mech: Multi-Dimension

$X = (x_1, \ldots, x_n)$ → Analyzer → $g(X) + \mathbf{DLap}(\Delta(g)/\varepsilon)$

Assumption: $\mathrm{Range}(g) \subseteq \mathbb{Z}^d$

(ie, g is vector-valued with integer entries)

ℓ_p-Sensitivity

$$\Delta_p(g) = \max_{X \asymp X'} \|g(X) - g(X')\|_p$$

ℓ_p-Norm

$$\|v\|_p = (|v_1|^p + \cdots + |v_d|^p)^{1/p}$$

Examples: Histogram

	name	zipcode	age	#convs
x_1	Dave	10520	40	150
x_2	Bob	10520	35	50
	...			
x_n	Carol	10500	30	30

Discrete Laplace Mech: Multi-Dimension

$\mathbf{X = (x_1, ..., x_n)}$ → Analyzer → $\mathbf{g(X) + DLap(\Delta(g)/\varepsilon)}$

Assumption: $\text{Range}(g) \subseteq \mathbb{Z}^d$

(ie, g is vector-valued with integer entries)

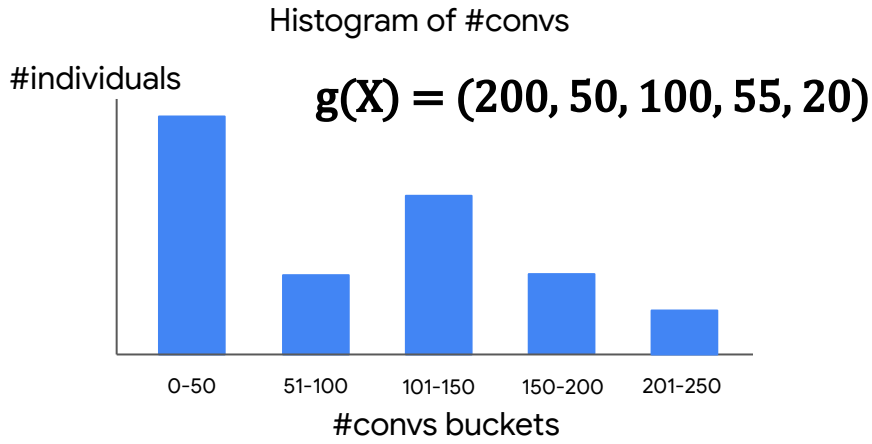
ℓ_p-Sensitivity

$$\Delta_p(g) = \max_{X \simeq X'} \|g(X) - g(X')\|_p$$

ℓ_p-Norm

$$\|v\|_p = (|v_1|^p + \cdots + |v_d|^p)^{1/p}$$

Examples: Histogram



Discrete Laplace Mech: Multi-Dimension

$\mathbf{X = (x_1, \ldots, x_n)}$ → Analyzer → $\mathbf{g(X) + DLap(\Delta(g)/\varepsilon)}$

Assumption: $\text{Range}(g) \subseteq \mathbb{Z}^d$

(ie, g is vector-valued with integer entries)

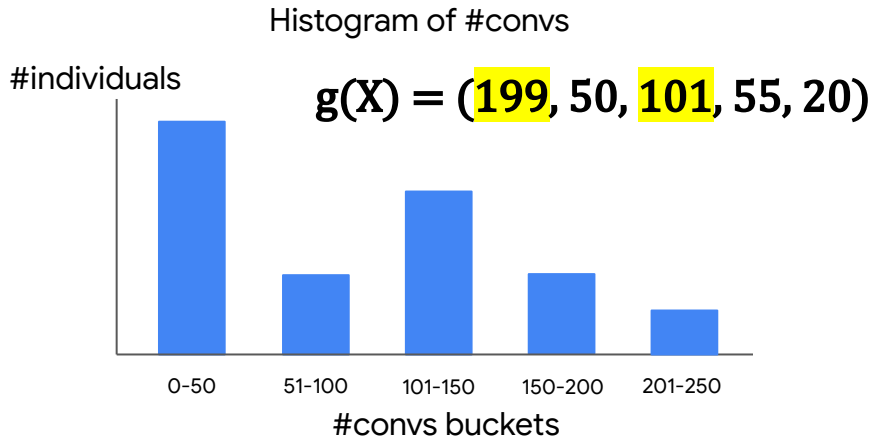
ℓ_p-Sensitivity

$$\Delta_p(g) = \max_{X \simeq X'} \|g(X) - g(X')\|_p$$

ℓ_p-Norm

$$\|v\|_p = (|v_1|^p + \cdots + |v_d|^p)^{1/p}$$

Examples: Histogram



Changing a row affects two buckets by ≤ 1 ⇒ $\Delta_1(g) \leq 2$

Discrete Laplace Mech: Multi-Dimension

$$X = (x_1, \ldots, x_n) \rightarrow \boxed{\text{Analyzer}} \rightarrow g(X) + \mathbf{DLap}(\Delta_1(g)/\varepsilon)^{\otimes d}$$

- Use l_1-sensitivity
- Each coordinate is independent

Assumption: $\text{Range}(g) \subseteq \mathbb{Z}^d$

(ie, g is vector-valued with integer entries)

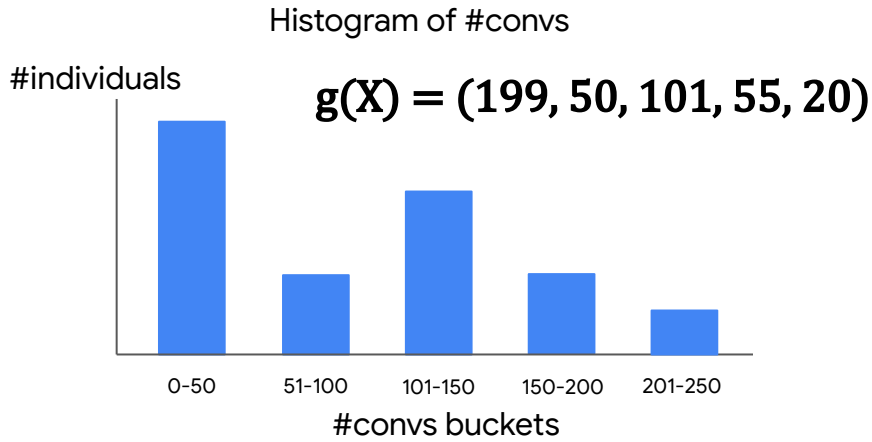
ℓ_p-Sensitivity

$$\Delta_p(g) = \max_{X \simeq X'} \|g(X) - g(X')\|_p$$

ℓ_p-Norm

$$\|v\|_p = (|v_1|^p + \cdots + |v_d|^p)^{1/p}$$

Examples: Histogram



Changing a row affects two buckets by ≤ 1 ⇒ $\Delta_1(g) \leq 2$

Discrete Laplace Mechanism

Theorem Assuming $\text{Range}(g) \subseteq \mathbb{Z}$, Discrete Laplace Mechanism is ε-DP.

Assumption: $\text{Range}(g) \subseteq \mathbb{Z}$

(ie, g is integer-valued)

Sensitivity

$\Delta(g) = \max_{X \simeq X'} |g(X) - g(X')|$

Larger sensitivity ⇒
More Noise Required

Discrete Laplace Distribution

For every integer i,

$$\Pr[i = DLap(b)] \propto e^{-|i|/b}$$

Discrete Laplace Mechanism

Theorem Assuming $\text{Range}(g) \subseteq \mathbb{Z}$, Discrete Laplace Mechanism is ε-DP.

Assumption: $\text{Range}(g) \subseteq \mathbb{Z}$

(ie, g is integer-valued)

Sensitivity

$\Delta(g) = \max_{X \simeq X'} |g(X) - g(X')|$

Larger sensitivity ⇒
More Noise Required

Discrete Laplace Distribution

For every integer i,

$\Pr[i = DLap(b)] \propto e^{-|i|/b}$

- Necessary!
- If $g(X) = 0.1, g(X') = 0.2$
  - Noise added is integer
  - Does not change the fractional part
  - Adversary can tell exactly which dataset it comes from

Laplace Mechanism

$X = (x_1, ..., x_n)$ → Analyzer → $g(X)$ + **Random Noise**

Assumption: $\text{Range}(g) \subseteq \mathbb{R}$

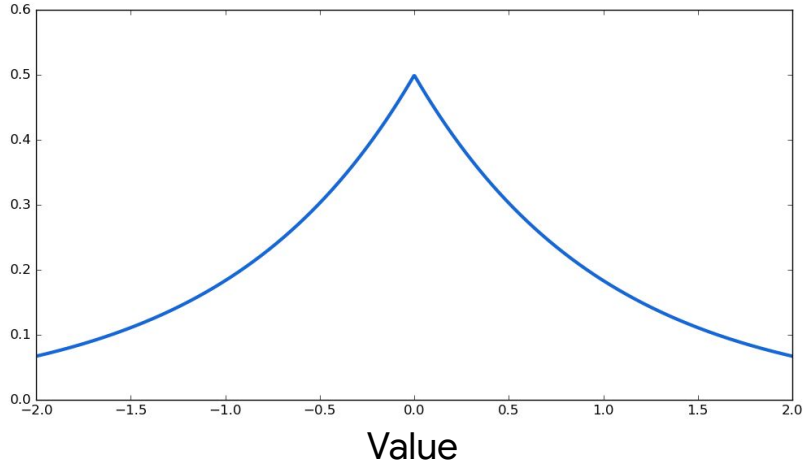
(ie, g is real-valued)

Laplace Distribution

For every real number z,

$$f_{Lap(b)}(z) \propto e^{-|z|/b}$$

Probability Density



Laplace Mechanism

$X = (x_1, ..., x_n)$ → Analyzer → $g(X) + Lap(\Delta(g)/\varepsilon)$

Theorem Assuming $\text{Range}(g) \subseteq \mathbb{R}$, Laplace Mechanism is ε-DP.

Assumption: $\text{Range}(g) \subseteq \mathbb{R}$

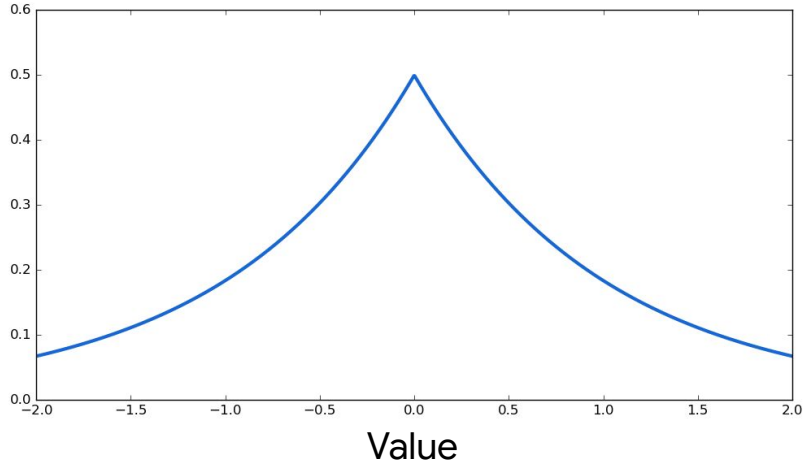
(ie, g is real-valued)

Laplace Distribution

For every real number z,

$$f_{Lap(b)}(z) \propto e^{-|z|/b}$$

Probability Density



Laplace Mechanism

$X = (x_1, ..., x_n)$ → Analyzer → $g(X) + \mathrm{Lap}(\Delta(g)/\varepsilon)$

Theorem Assuming $\mathrm{Range}(g) \subseteq \mathbb{R}$, Laplace Mechanism is ε-DP.

Theorem MSE = $2(\Delta(g)/\varepsilon)^2$.

Assumption: $\mathrm{Range}(g) \subseteq \mathbb{R}$

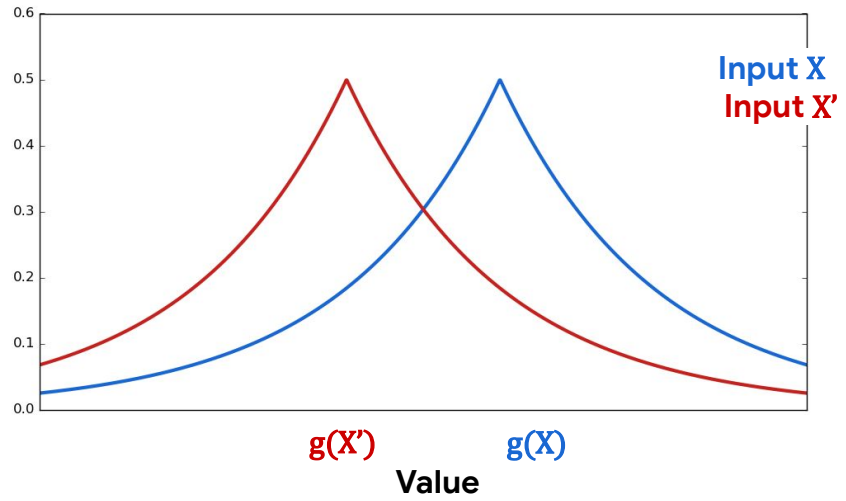
(ie, g is real-valued)

Laplace Distribution

For every real number z,

$$f_{\mathrm{Lap}(b)}(z) \propto e^{-|z|/b}$$

Probability Density



Laplace Mechanism: Multi-Dimension

$X = (x_1, ..., x_n)$ → Analyzer → $g(X) + Lap(\Delta_1(g)/\varepsilon)^{\otimes d}$

- Use l_1-sensitivity
- Each coordinate is independent

Theorem Assuming $\text{Range}(g) \subseteq \mathbb{R}^d$, Laplace Mechanism is ε-DP.

Assumption: $\text{Range}(g) \subseteq \mathbb{R}^d$

(ie, g is vector-valued with real entries)

Laplace Distribution

For every real number z,

$$f_{Lap(b)}(z) \propto e^{-|z|/b}$$

Gaussian Mechanism

$$X = (x_1, \ldots, x_n) \rightarrow \boxed{\text{Analyzer}} \rightarrow g(X) + \mathcal{N}(0, \sigma^2)$$

Assumption: $\text{Range}(g) \subseteq \mathbb{R}$

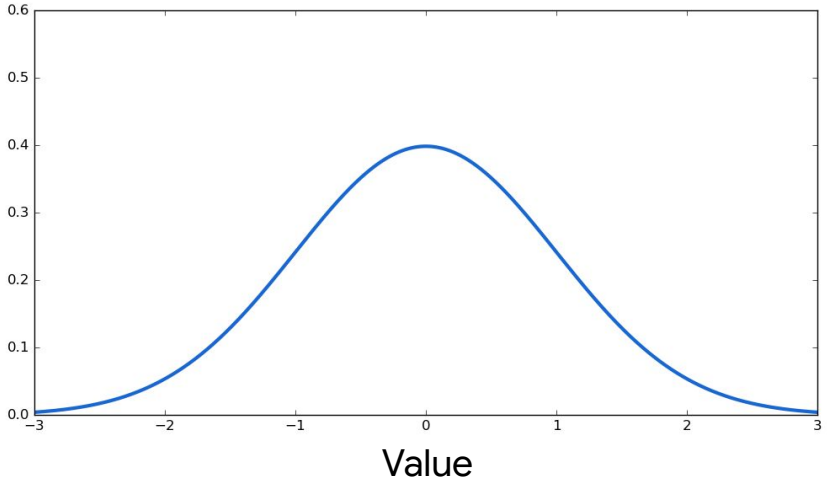
(ie, g is real-valued)

Gaussian Distribution

For every real number z,

$$\text{PDF}_{\mathcal{N}(0,\sigma^2)}(z) \propto e^{-(z/\sigma)^2}$$

Probability Density



Gaussian Mechanism

$$\sigma = \frac{2\sqrt{2\ln(2/\delta)}}{\epsilon} \cdot \Delta(g)$$

$$X = (x_1, \ldots, x_n) \rightarrow \boxed{\text{Analyzer}} \rightarrow g(X) + \mathcal{N}(0, \sigma^2)$$

Theorem Assuming $\text{Range}(f) \subseteq \mathbb{R}$ and $\varepsilon \leq 1$, Gaussian Mechanism is (ε, δ)-DP.

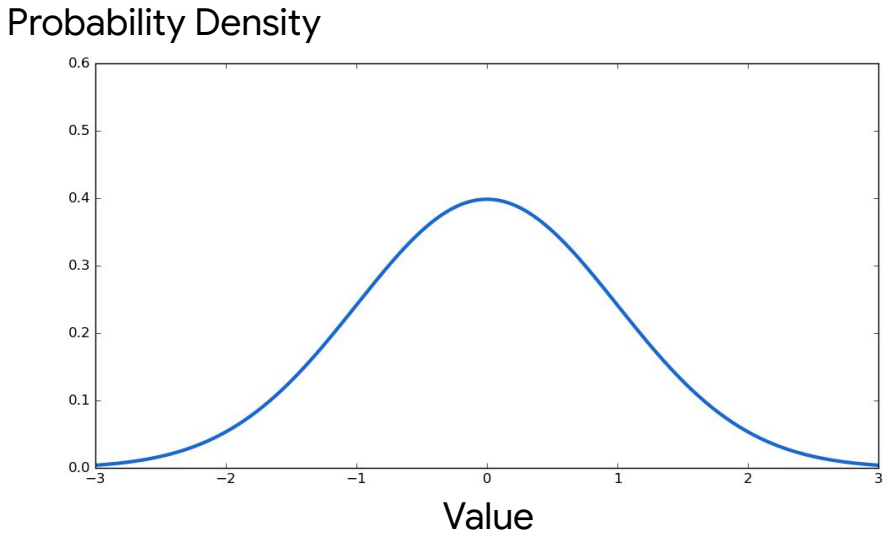
Assumption: $\text{Range}(g) \subseteq \mathbb{R}$

(ie, g is real-valued)

Gaussian Distribution

For every real number z,

$$\text{PDF}_{\mathcal{N}(0,\sigma^2)}(z) \propto e^{-(z/\sigma)^2}$$



Gaussian Mechanism

$$\sigma = \frac{2\sqrt{2\ln(2/\delta)}}{\epsilon} \cdot \Delta(g)$$

$$X = (x_1, \ldots, x_n) \rightarrow \boxed{\text{Analyzer}} \rightarrow g(X) + \mathcal{N}(0, \sigma^2)$$

Theorem Assuming $\text{Range}(f) \subseteq \mathbb{R}$ and $\varepsilon \leq 1$, Gaussian Mechanism is (ε, δ)-DP.

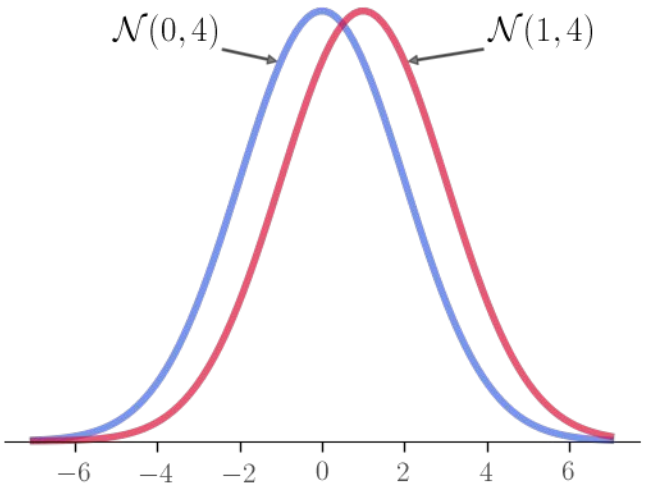
Assumption: $\text{Range}(g) \subseteq \mathbb{R}$

(ie, g is real-valued)

Gaussian Distribution

For every real number z,

$$\text{PDF}_{\mathcal{N}(0,\sigma^2)}(z) \propto e^{-(z/\sigma)^2}$$



Gaussian Mechanism

$$\sigma = \frac{2\sqrt{2\ln(2/\delta)}}{\epsilon} \cdot \Delta_2(g)$$

$$X = (x_1, \ldots, x_n) \rightarrow \boxed{\text{Analyzer}} \rightarrow g(X) + \mathcal{N}(0, \sigma^2)^{\otimes d}$$

- Use l_2-sensitivity
- Each coordinate is independent

Theorem Assuming $\text{Range}(f) \subseteq \mathbb{R}^d$ and $\varepsilon \leq 1$, Gaussian Mechanism is (ε, δ)-DP.

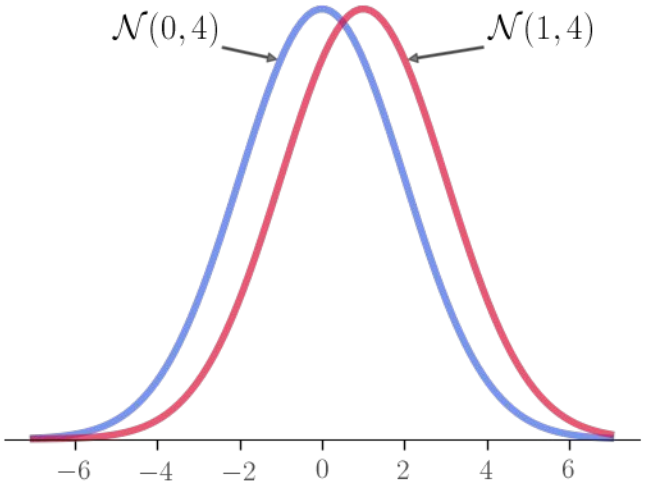
Assumption: $\text{Range}(g) \subseteq \mathbb{R}^d$

(ie, g is vector-valued with real entries)

Gaussian Distribution

For every real number z,

$$\text{PDF}_{\mathcal{N}(0,\sigma^2)}(z) \propto e^{-(z/\sigma)^2}$$



Gaussian Mechanism

$$\sigma = \frac{2\sqrt{2\ln(2/\delta)}}{\epsilon} \cdot \Delta_2(g)$$

$$X = (x_1, \ldots, x_n) \rightarrow \boxed{\text{Analyzer}} \rightarrow g(X) + \mathcal{N}(0, \sigma^2)^{\otimes d}$$

- Use l_2-sensitivity
- Each coordinate is independent

Theorem Assuming $\text{Range}(f) \subseteq \mathbb{R}^d$ and $\varepsilon \leq 1$, Gaussian Mechanism is (ε, δ)-DP.

Assumption: $\text{Range}(g) \subseteq \mathbb{R}^d$

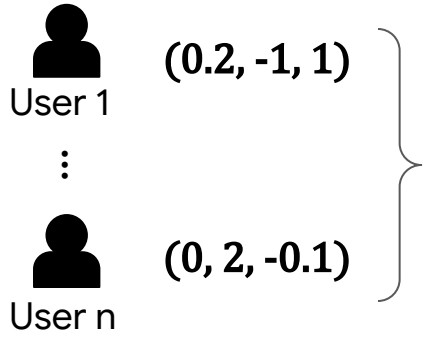
(ie, g is vector-valued with real entries)

Gaussian Distribution

For every real number z,

$$\text{PDF}_{\mathcal{N}(0,\sigma^2)}(z) \propto e^{-(z/\sigma)^2}$$

Examples: Vector summation

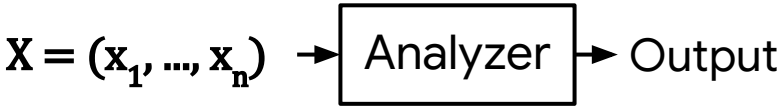


Assumption: Each vector has l_2-norm $\leq C$ ⇒ $\Delta_2(g) \leq 2C$

Threat Models

Central vs Local DP

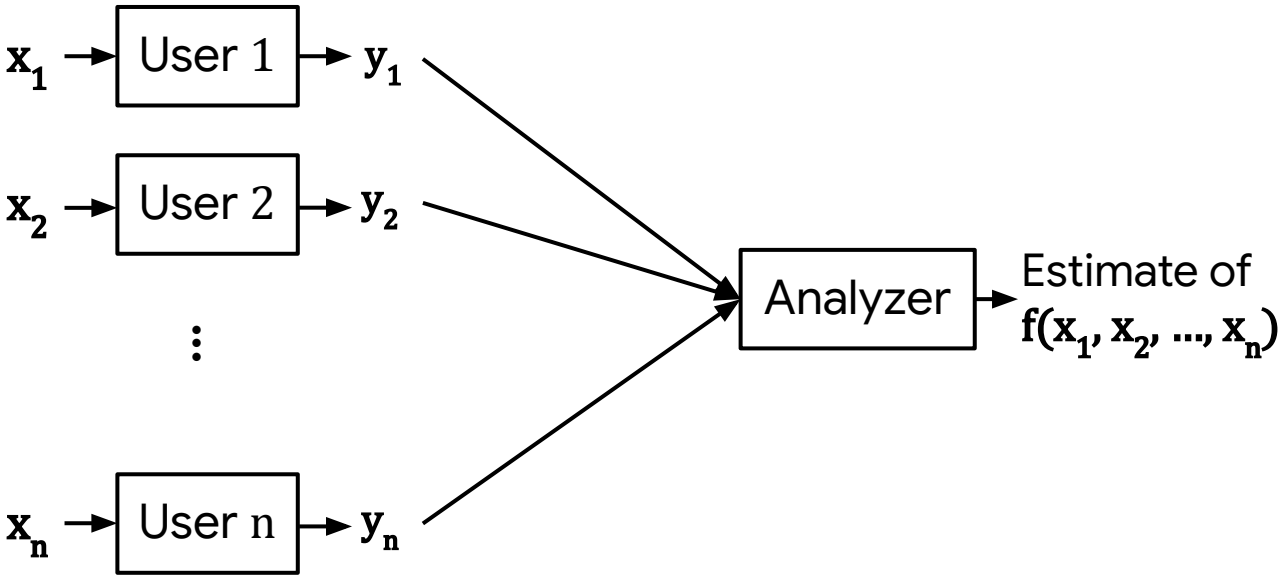
Model studied so far



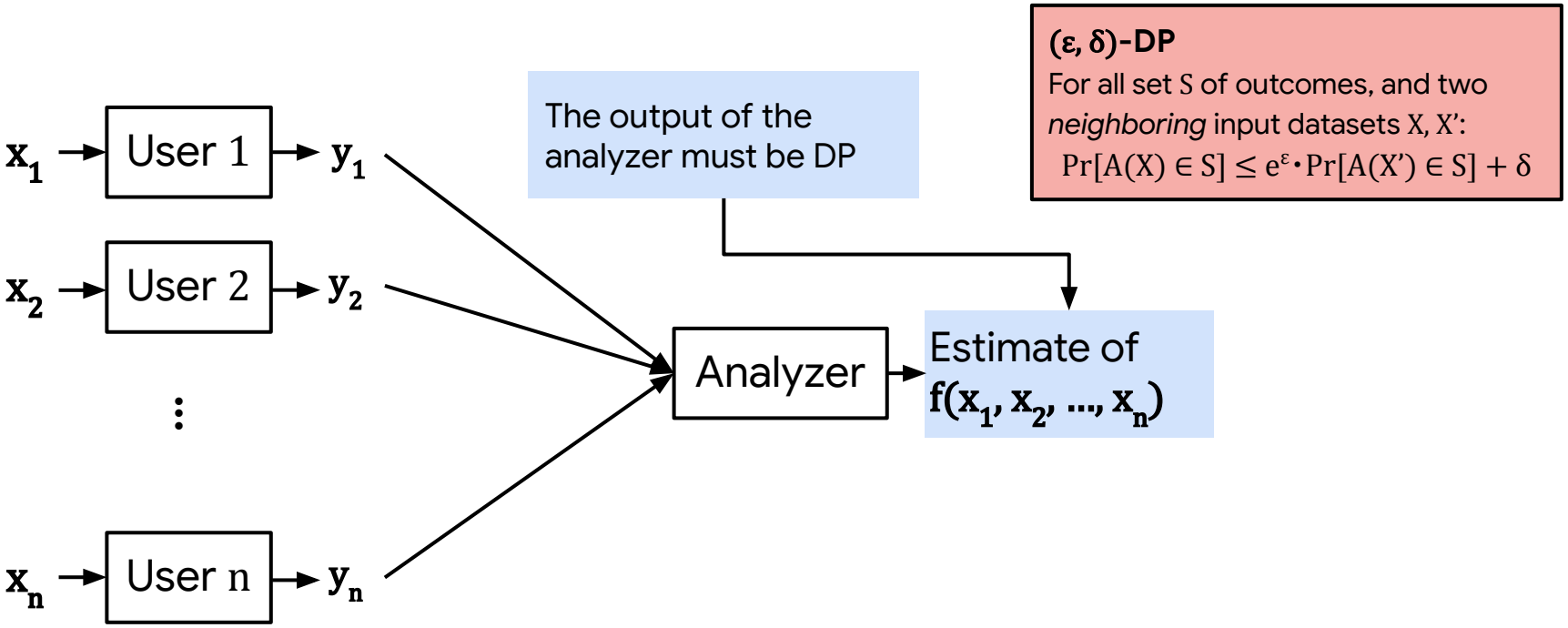
“Central DP”

- Analyzer gets to see raw data
- Undesirable if:
  - There is no trusted central authority
  - The analysis is done in a distributed manner

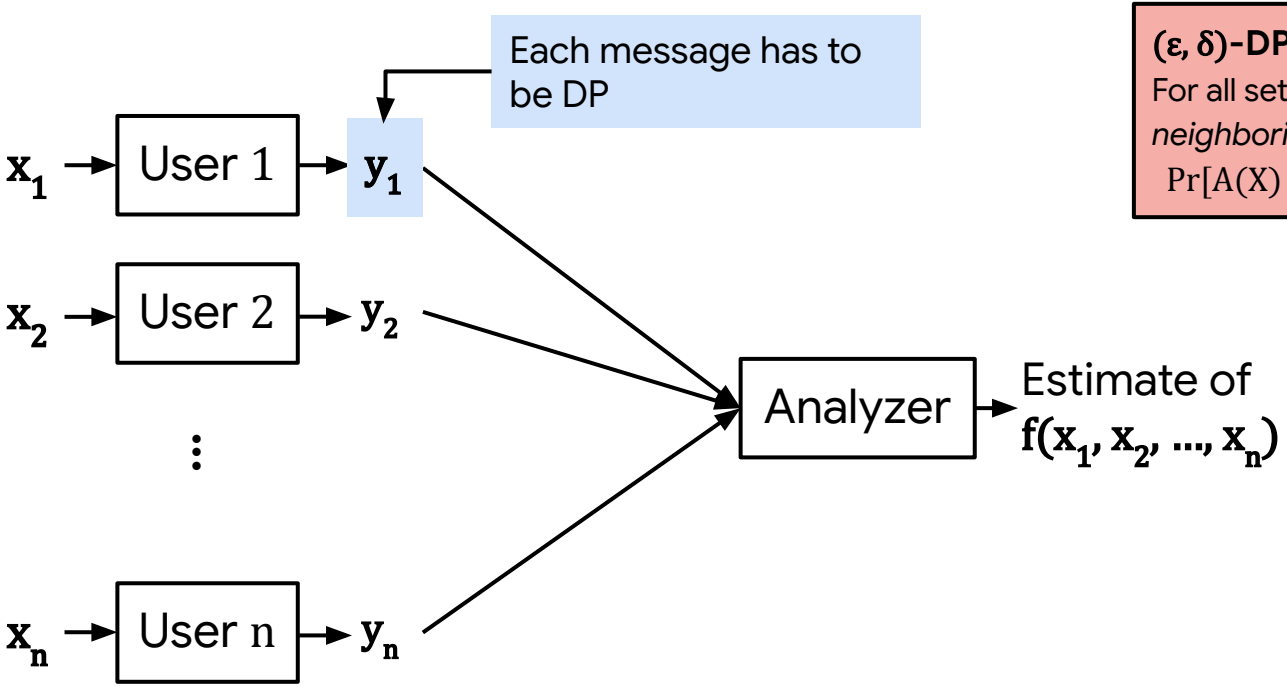
Distributed Analytics



Central DP Model



Local DP Model *[Kasiviswanathan et al.]*

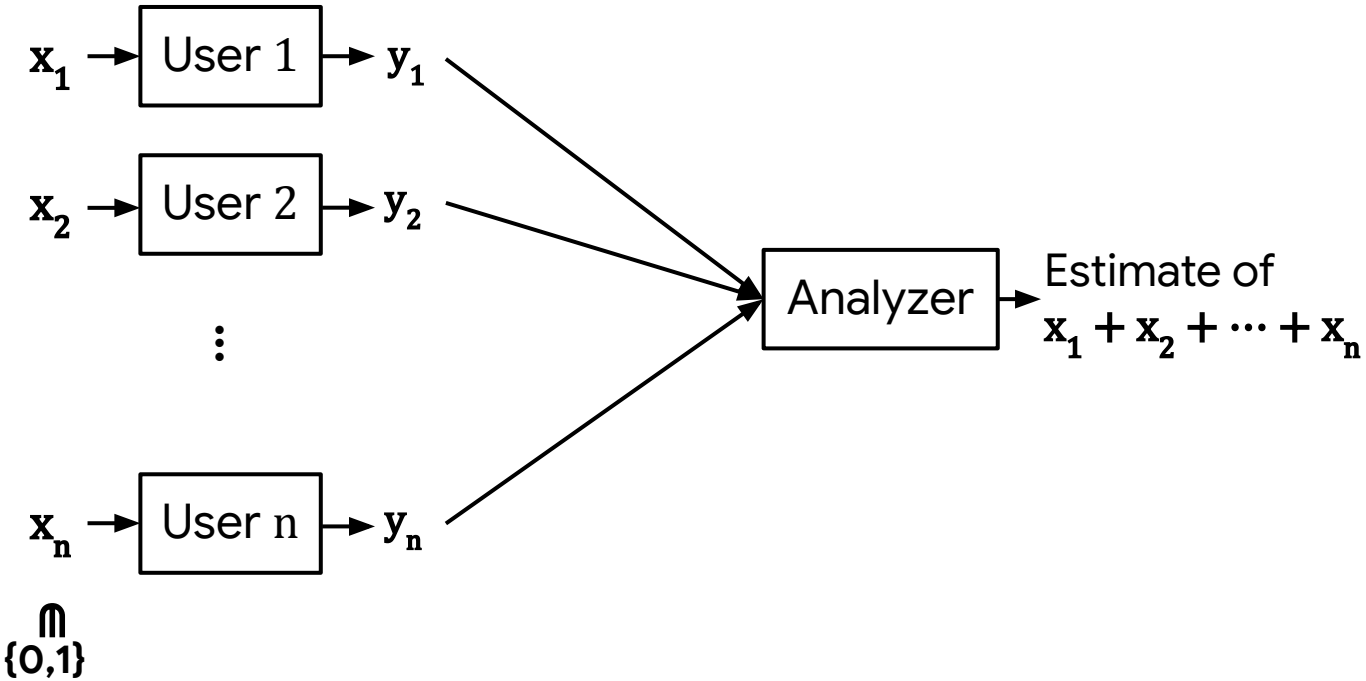


(ε, δ)-DP

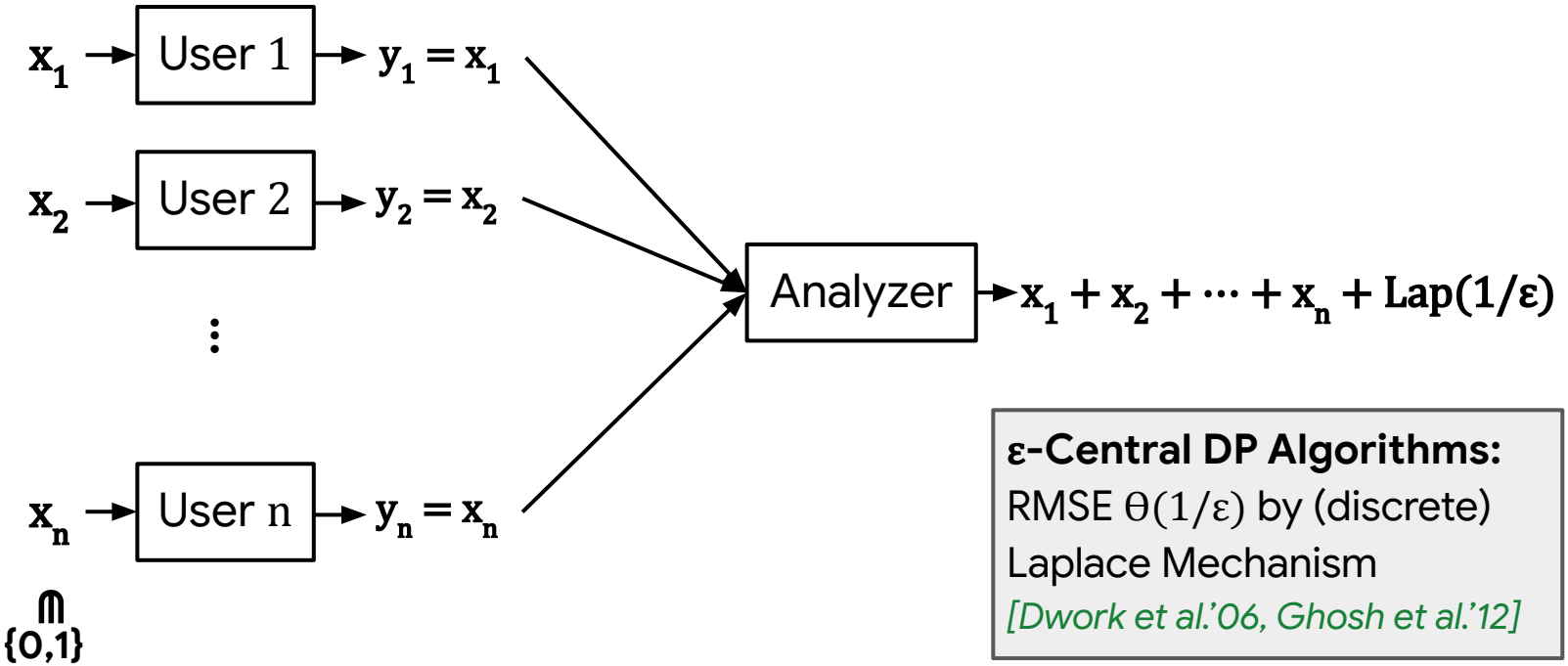
For all set S of outcomes, and two *neighboring* input datasets X, X':

$$\Pr[A(X) \in S] \leq e^{\varepsilon} \cdot \Pr[A(X') \in S] + \delta$$

Distributed Analytics: Counting

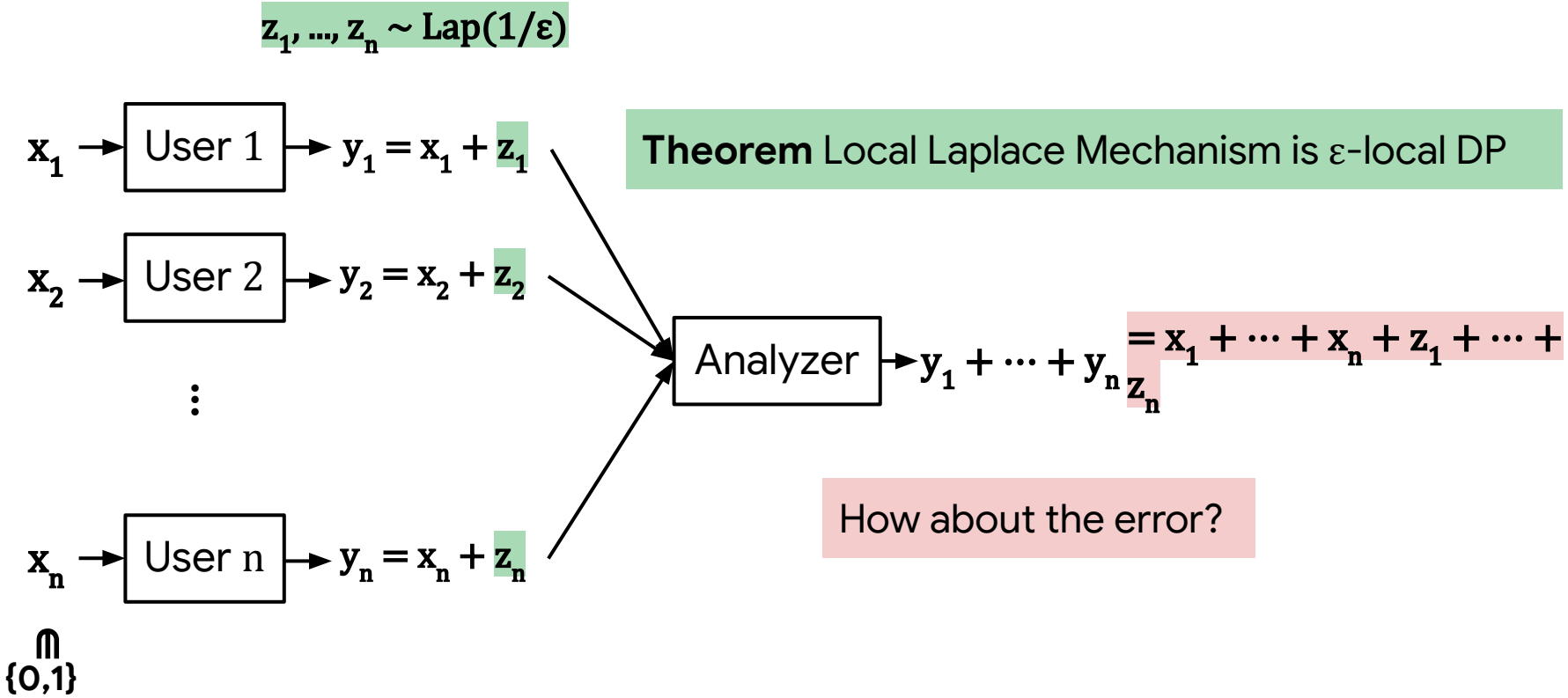


Counting in *Central* DP: Laplace Mechanism



ε-Central DP Algorithms:
RMSE $\theta(1/\varepsilon)$ by (discrete) Laplace Mechanism
[Dwork et al.'06, Ghosh et al.'12]

Counting in *Local* DP: Laplace Mechanism



Local Laplace Mechanism: Utility

Theorem Estimator from Local Laplace Mechanism has MSE $2n/\varepsilon^2$

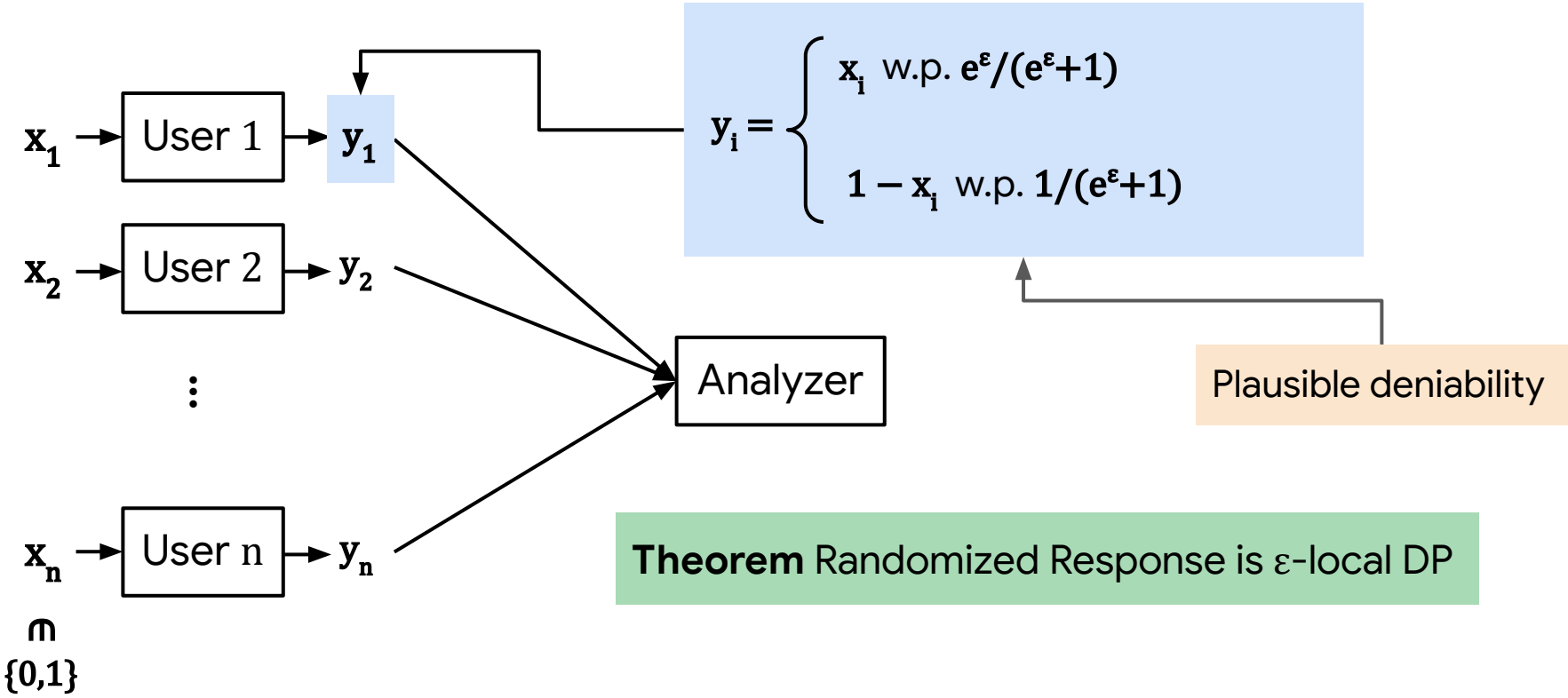
$\text{RMSE} = \sqrt{(2n)}/\varepsilon$

Proof

$$
\begin{aligned}
\text{MSE} &= \mathbf{E}[(\textstyle\sum_{i\in[n]} y_i - \sum_{i\in[n]} x_i)^2] \\
&= \mathbf{E}[(\textstyle\sum_{i\in[n]} z_i)^2] \\
&= \textstyle\sum_{i\in[n]} \mathbf{E}[z_i^2] \\
&= \textstyle\sum_{i\in[n]} \text{Var}(z_i) \\
&= n\ (2/\varepsilon^2)
\end{aligned}
$$

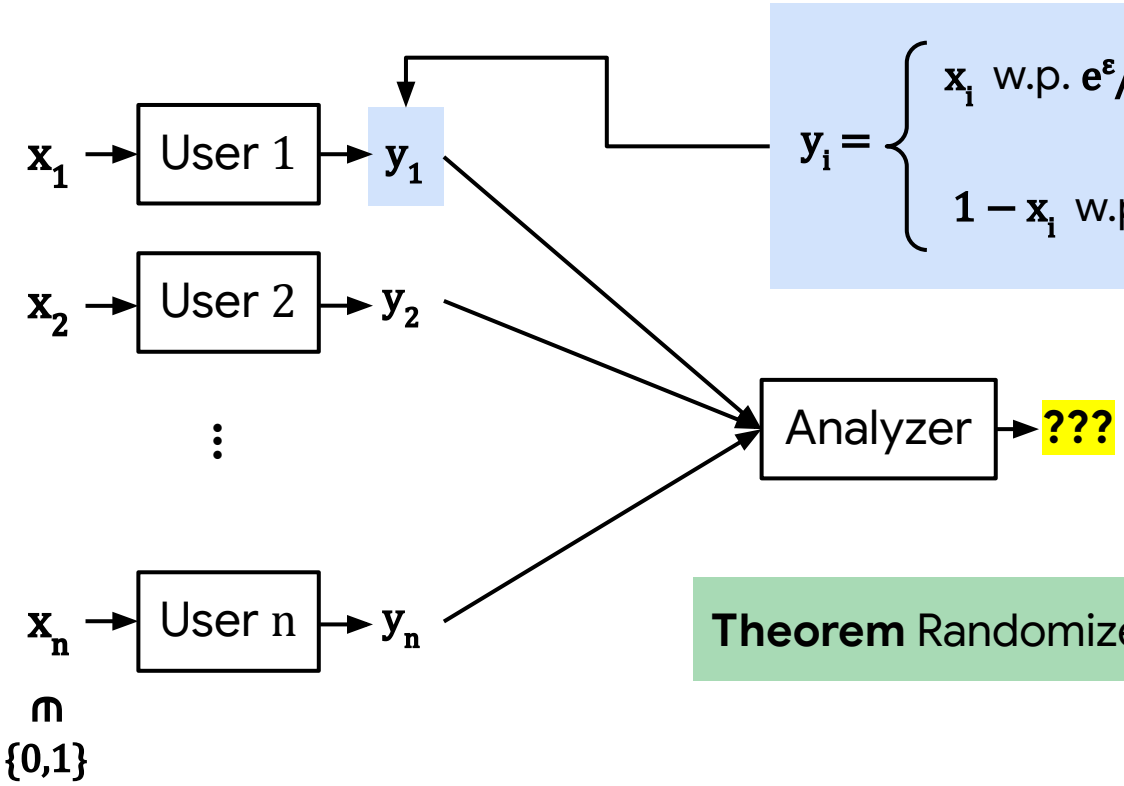
QED

Randomized Response (RR) *[Warner]*



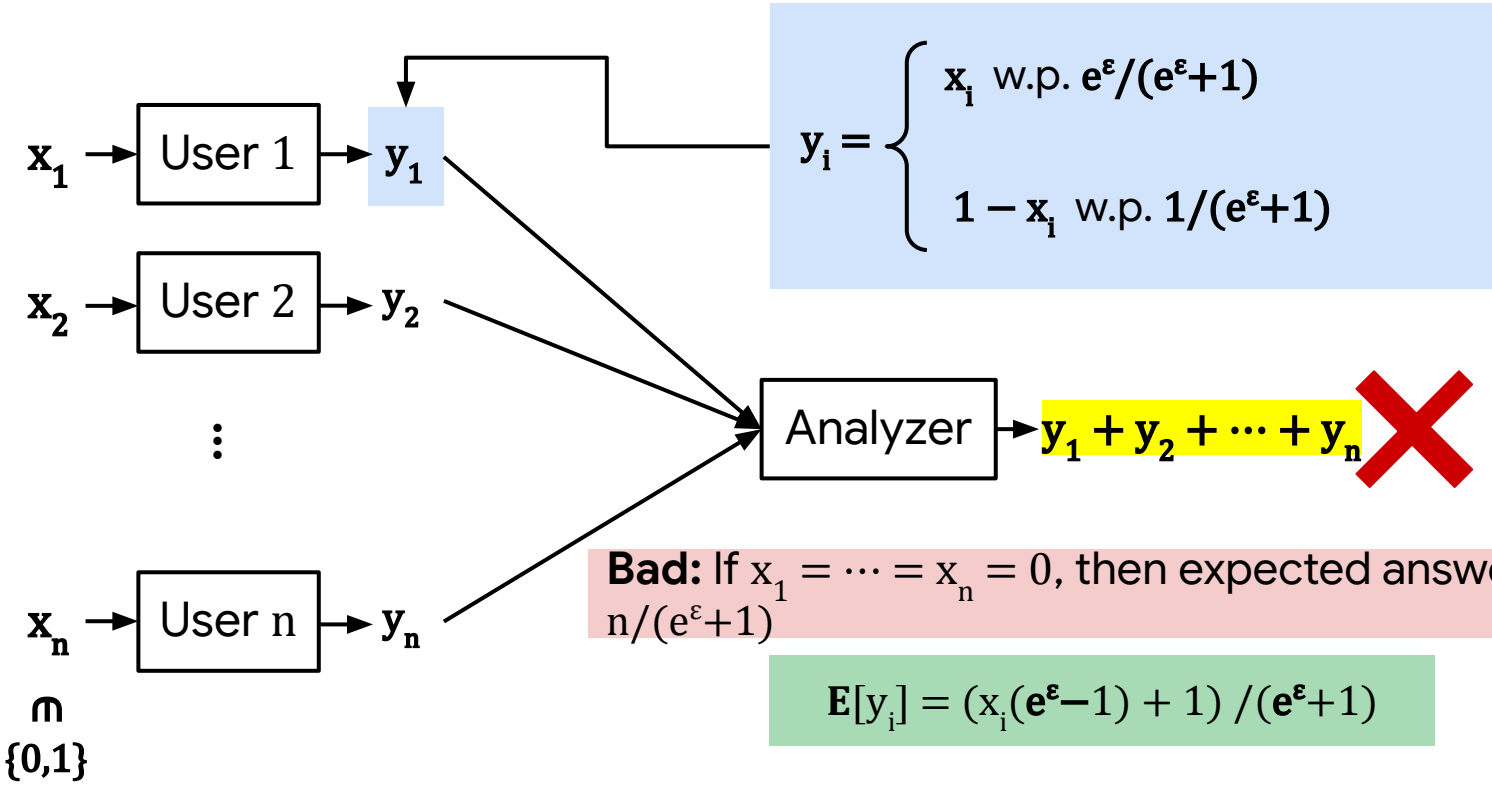
Theorem Randomized Response is ε-local DP

Randomized Response (RR): Estimator



Theorem Randomized Response is ε-local DP

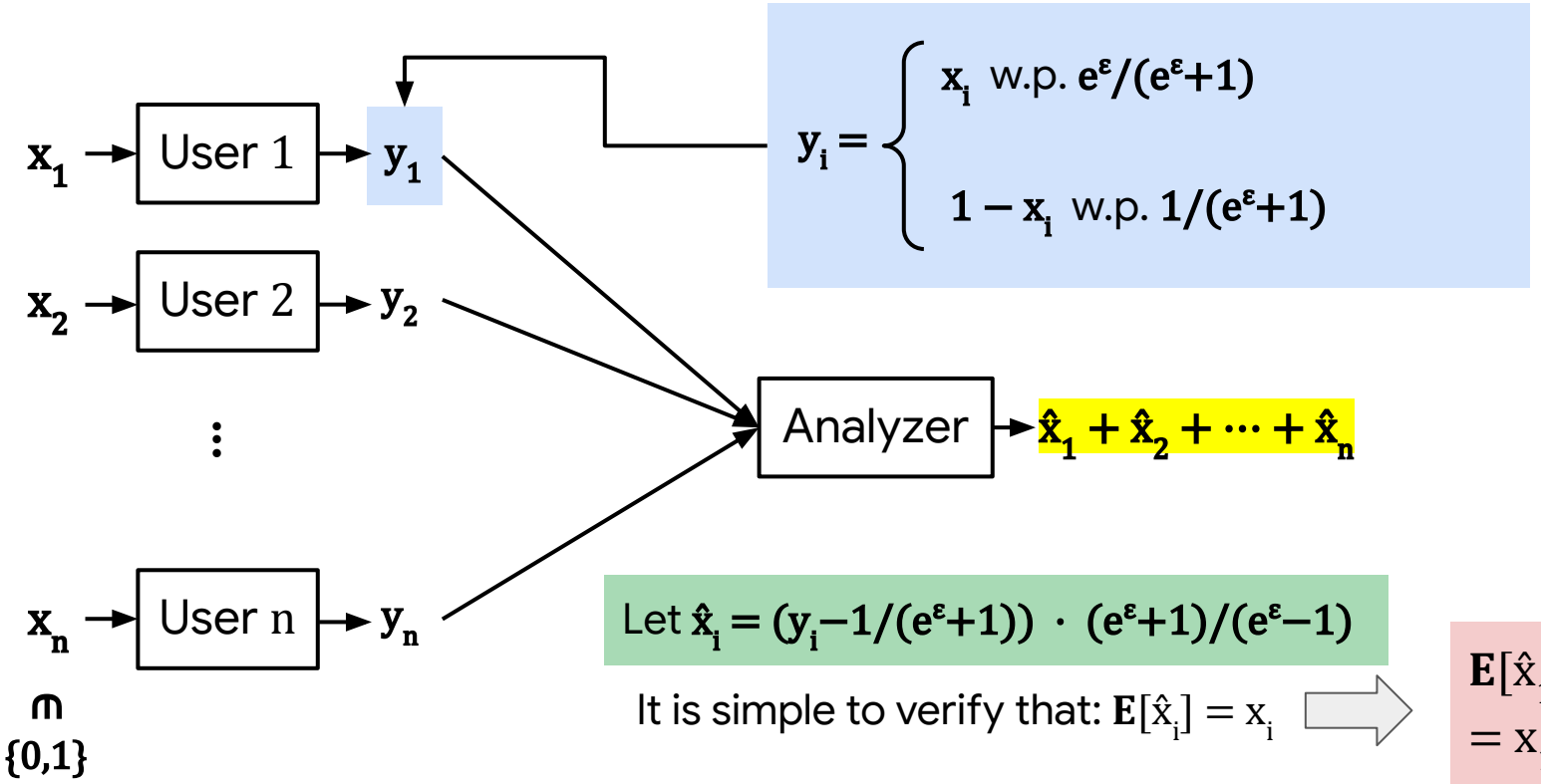
Randomized Response (RR): Estimator



Bad: If $x_1 = \cdots = x_n = 0$, then expected answer $= n/(e^{\varepsilon}+1)$

$$\mathbf{E}[y_i] = (x_i(e^{\varepsilon}-1) + 1)\ /(e^{\varepsilon}+1)$$

Randomized Response (RR): Estimator



Randomized Response: Utility

Theorem Estimator from RR has MSE $n\, e^{\varepsilon} / (e^{2\varepsilon} - 1)^2$ $\quad \text{RMSE} = O(\sqrt{n}/\varepsilon)$

Proof It is simple to verify that:

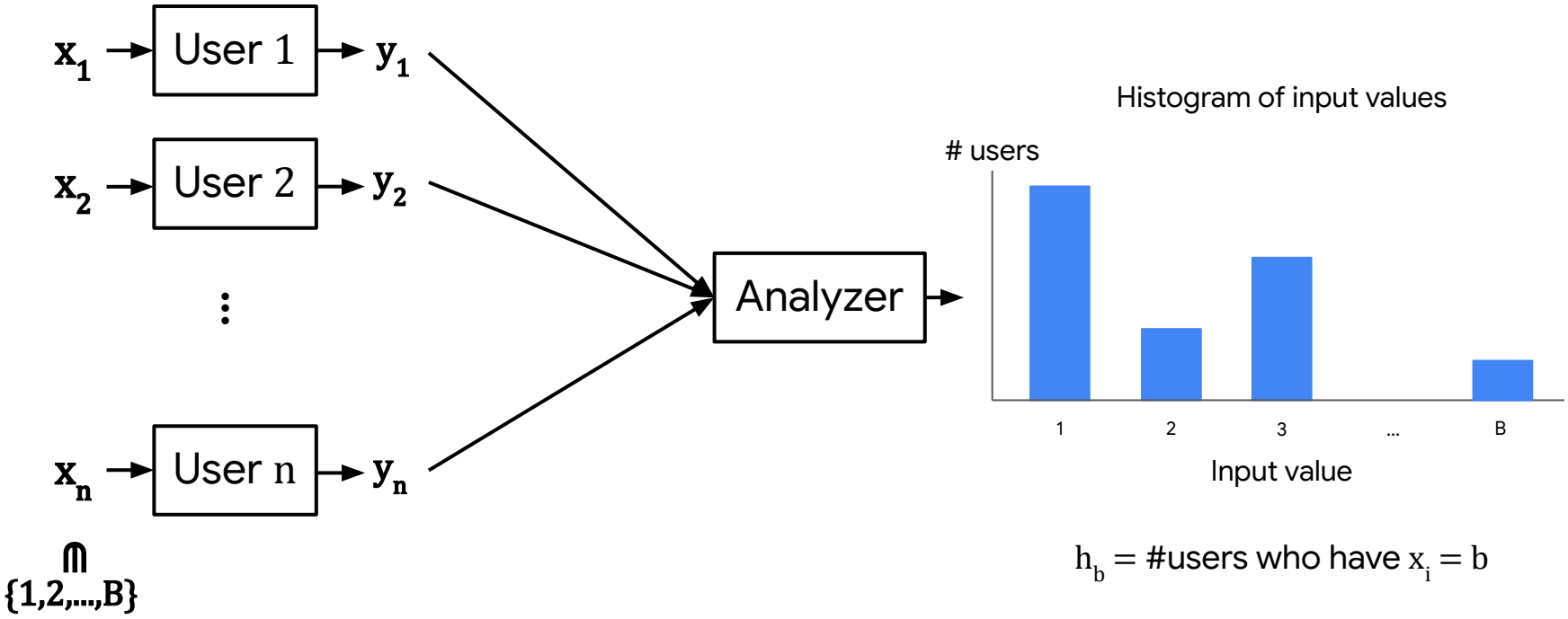
- $\mathbf{E}[\hat{x}_i] = x_i$
- $\text{Var}[\hat{x}_i] = e^{\varepsilon} / (e^{2\varepsilon} - 1)^2$

Thus,

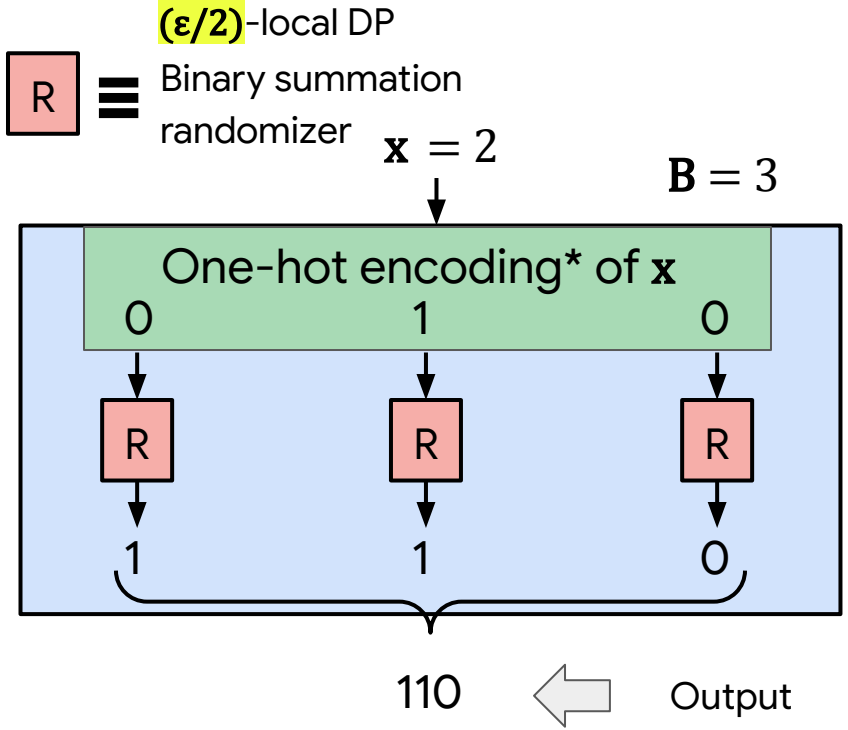
$$\begin{aligned} \text{MSE} &= \mathbf{E}[(\textstyle\sum_{i\in[n]} \hat{x}_i - \sum_{i\in[n]} x_i)^2] \\ &= \textstyle\sum_{i\in[n]} \mathbf{E}[(\hat{x}_i - x_i)^2] \\ &= \textstyle\sum_{i\in[n]} \text{Var}(\hat{x}_i) \\ &= n\, e^{\varepsilon} / (e^{2\varepsilon} - 1)^2 \end{aligned}$$

QED

Recall: Histogram Problem



RAPPOR Algorithm *[Erlingsson et al.'14]*



*B-length bit string of zeros, except one at x

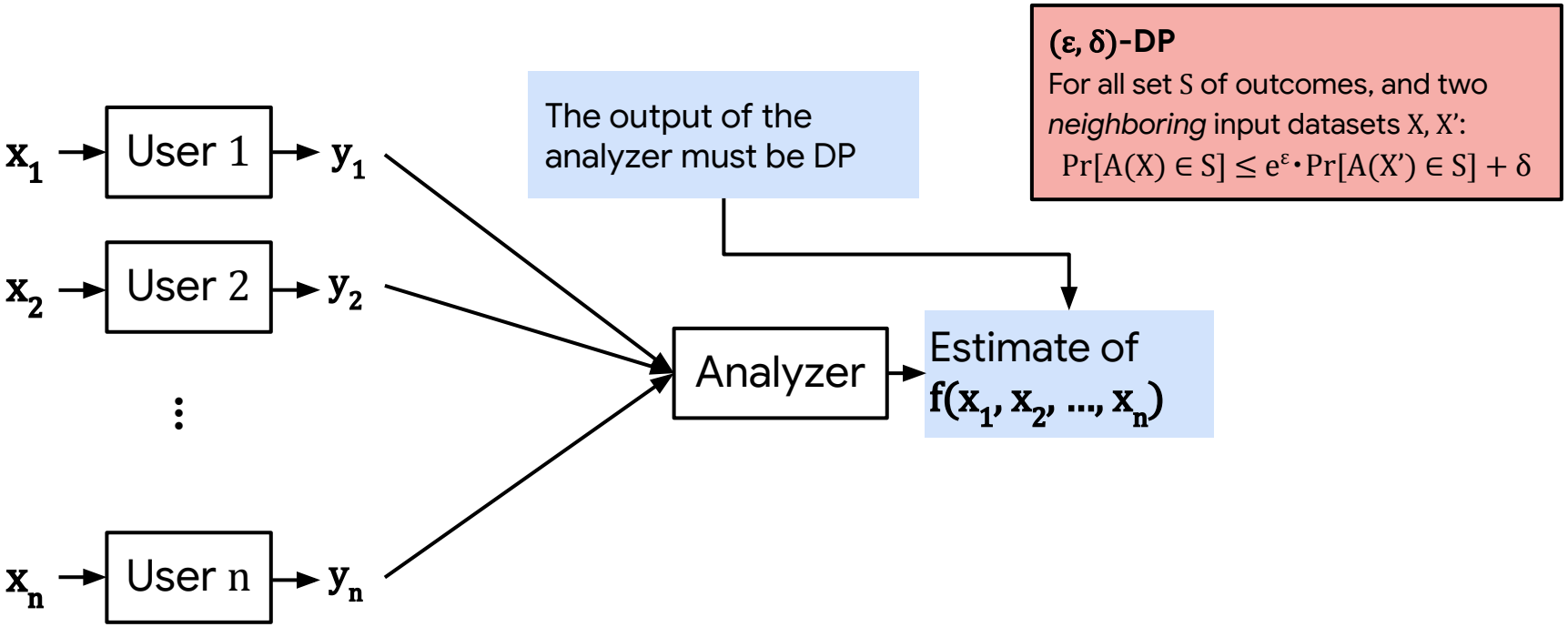
Theorem RAPPOR is ε-DP.

Utility: Exactly the same as RR with $\varepsilon/2$-DP

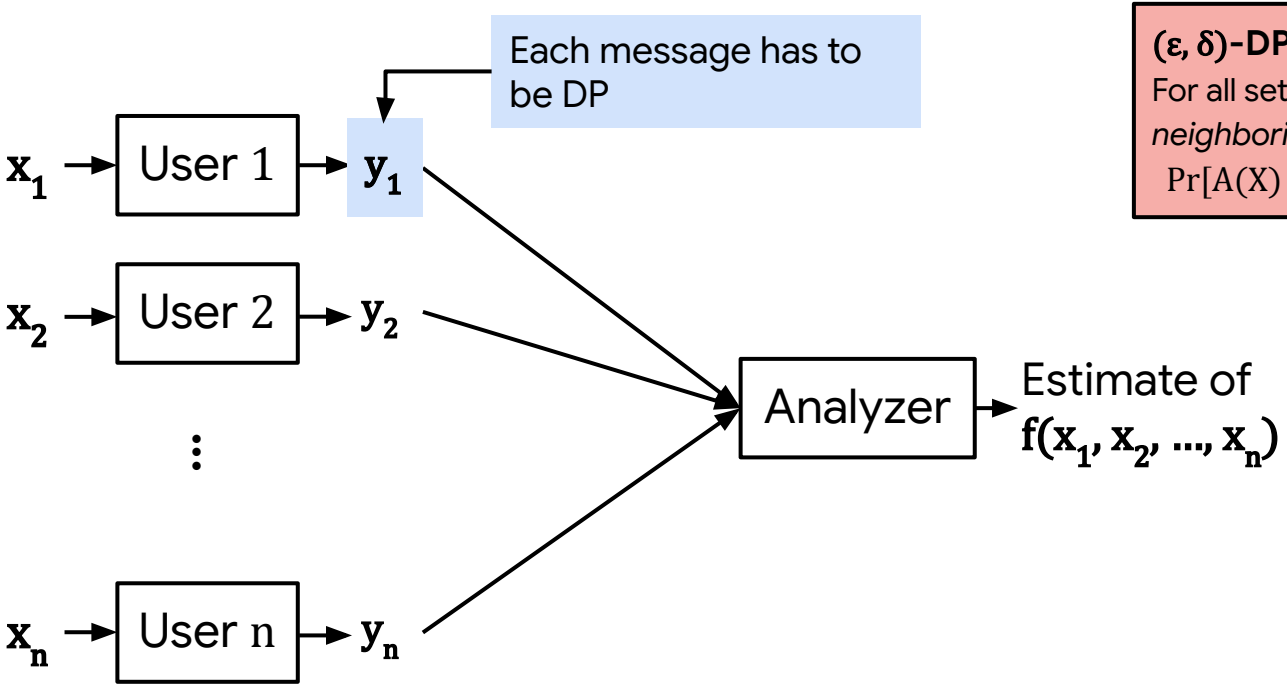
Theorem MSE of each entry of histogram is $n\, e^{\varepsilon/2} / (e^{\varepsilon} - 1)^2$.

Shuffle DP, SMPC, TEEs

Recall: Central DP Model



Recall: Local DP Model

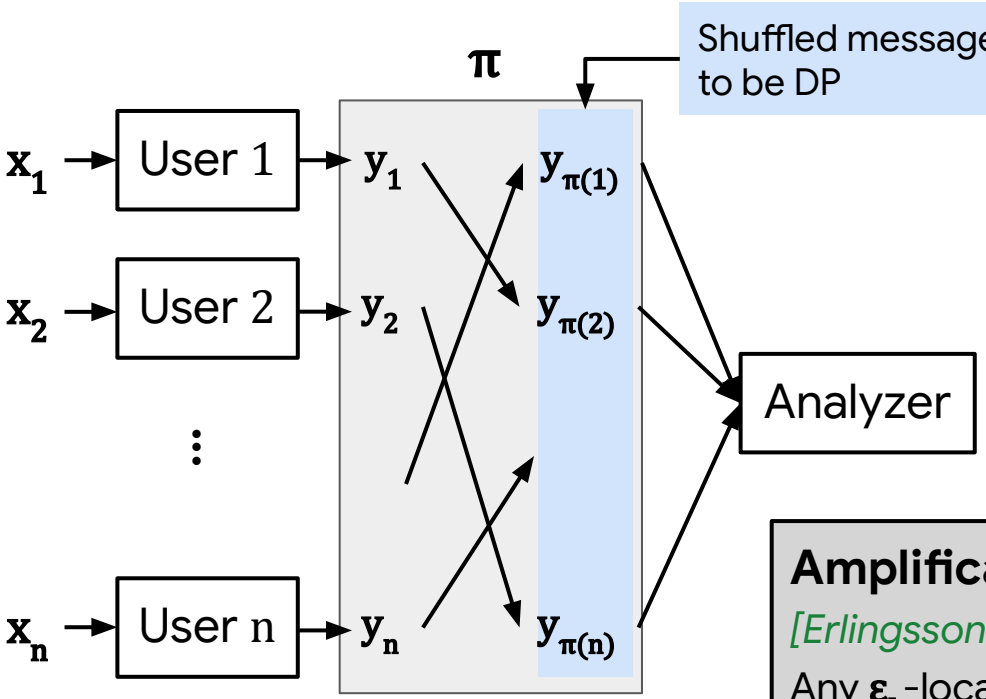


(ε, δ)-DP

For all set S of outcomes, and two *neighboring* input datasets X, X':

$$\Pr[A(X) \in S] \leq e^{\varepsilon} \cdot \Pr[A(X') \in S] + \delta$$

Shuffle DP Model *[Bittau et al., Erlingsson et al.]*



Shuffled messages have to be DP

≡

Multiset of messages have to be DP

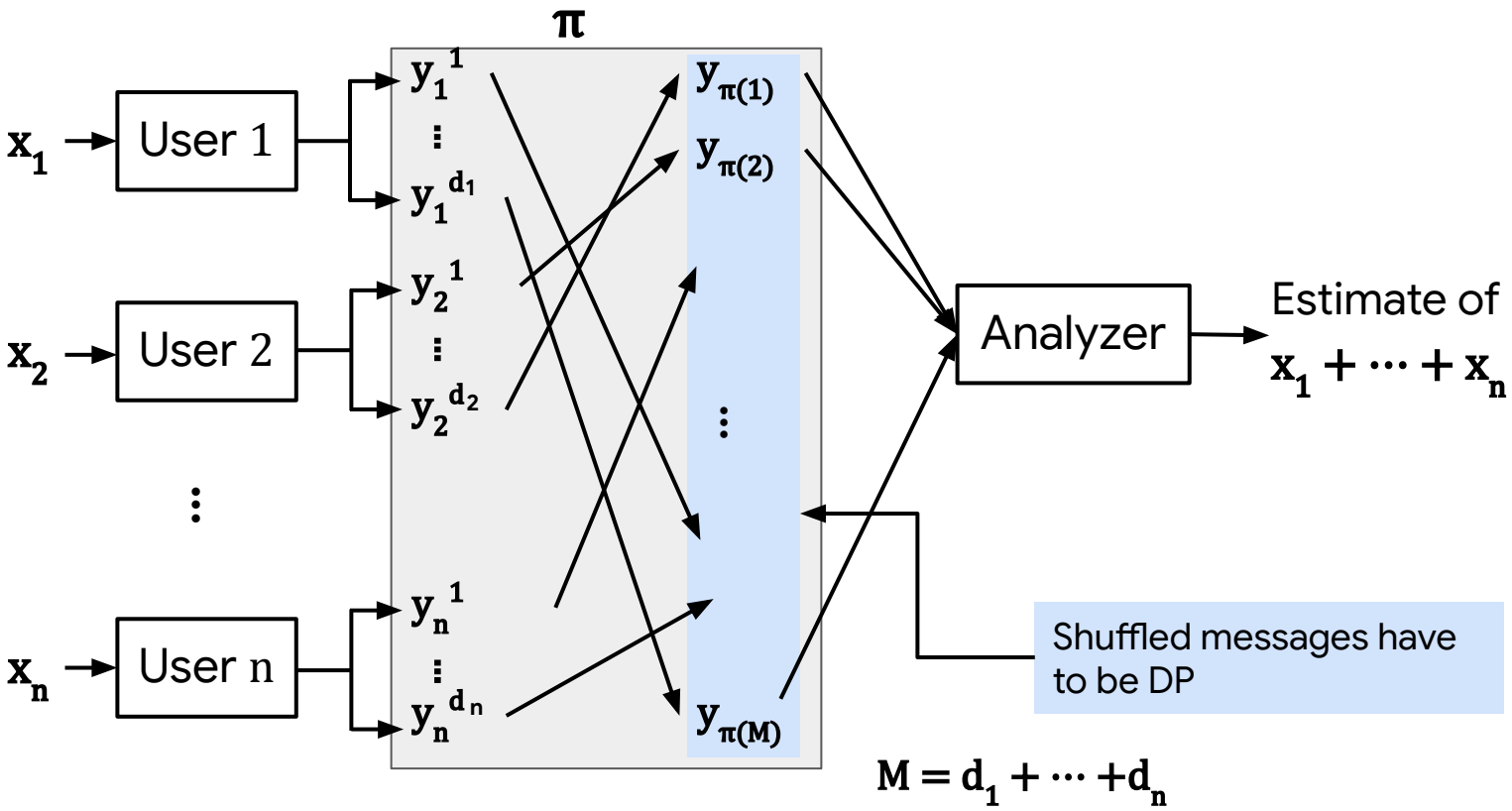
- **Anonymous Model** *[Ishai et al.]*
- **Encode-Shuffle-Analyze Architecture** *[Bittau et al.]*
- **Shuffled/Shuffle Model** *[Cheu et al., Erlingsson et al.]*

Amplification by Shuffling (Informal)

[Erlingsson et al., Balle et al., Feldman et al.]

Any ε_L-local DP algorithm is (ε, δ)-shuffle DP for $\varepsilon \ll \varepsilon_L$ and for reasonable value of δ.

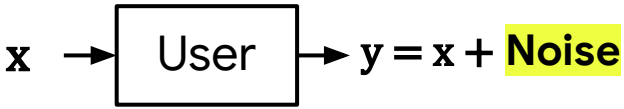
Shuffle DP Model: Multi-Message Setting



Summation in Shuffle DP Model

Single-Message

Randomizer:



Analyzer:

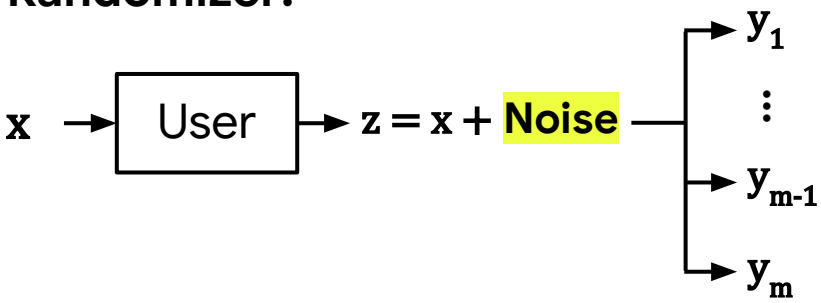
Just sum all messages up!

Benefit over Local DP:
Amplification by Shuffling

Multi-Message:

Split-and-Mix Protocol *[Balle et al.'19]*

Randomizer:



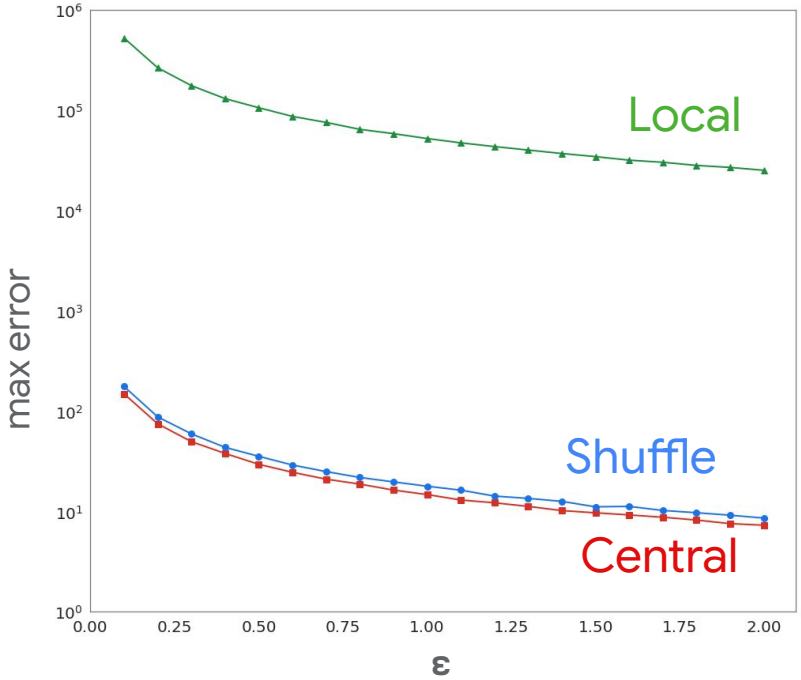
$y_1, \ldots, y_m$ randomly selected so that $y_1 + \cdots + y_m = z$

As good accuracy as central DP!

Summation in Shuffle DP Model

Experiment (IPUMS 1940 City Dataset)

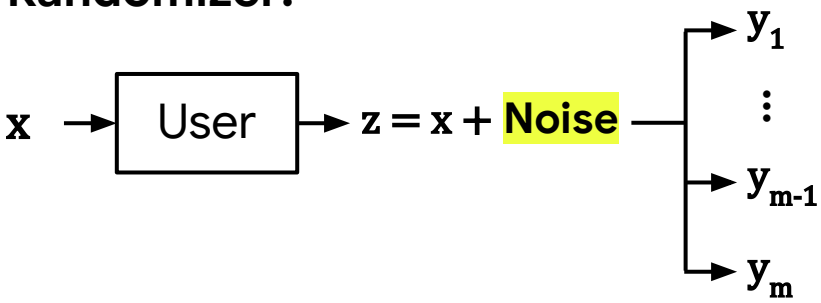
Parameters: n ≈ 60M, B = 915, □ = 2 * 10^{-9}



Multi-Message:

Split-and-Mix Protocol *[Balle et al.'19]*

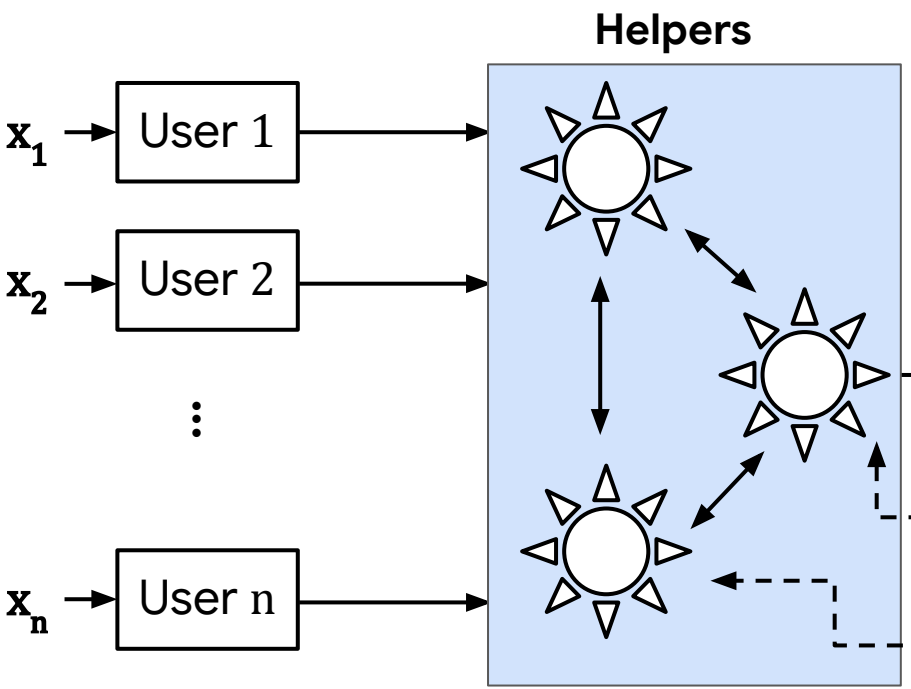
Randomizer:



$y_1, \ldots, y_m$ randomly selected so that $y_1 + \cdots + y_m = z$

As good accuracy as central DP!

Secure Multi-Party Computation (SMPC)



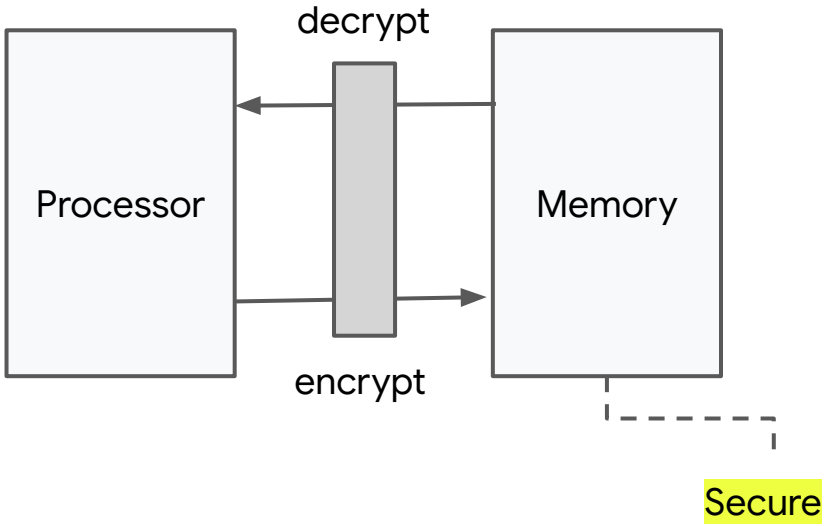
Can compute any function using garbled circuits

Analyst learns output but helpers learn nothing

Honest-but-curious
Malicious
...

Can be used with central DP protocols

Trusted Execution Environments (TEEs)



Higher performance
Remote attestation of code and data
Offered by Cloud providers

DP Ad Analytics

Reach & Frequency

Reach

- # unique users who view the ads

k-Frequency

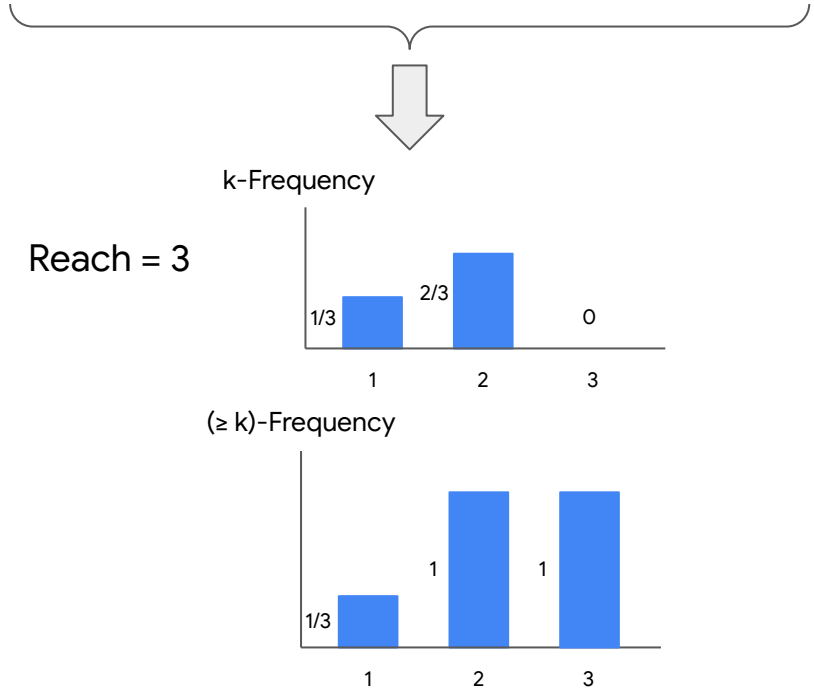
- Fraction of users that view the ads *exactly* k times among those who see ads at least once

(≥ k)-Frequency

- Fraction of users that view the ads *at least* k times among those who see ads at least once

User 1: See ads once, **User 2:** Doesn't see ads

User 3: See ads twice, **User 4:** See ads twice

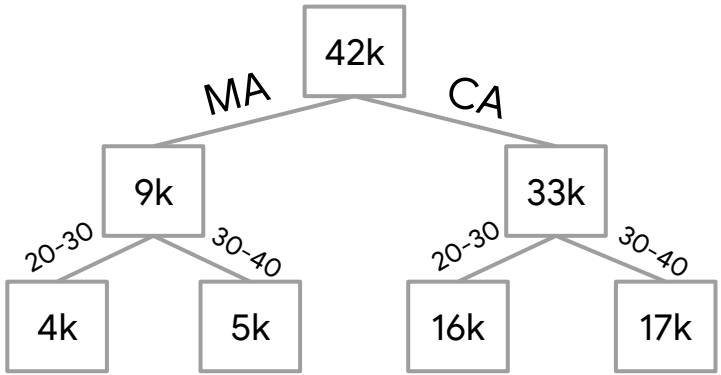


Slicing Queries

Example queries

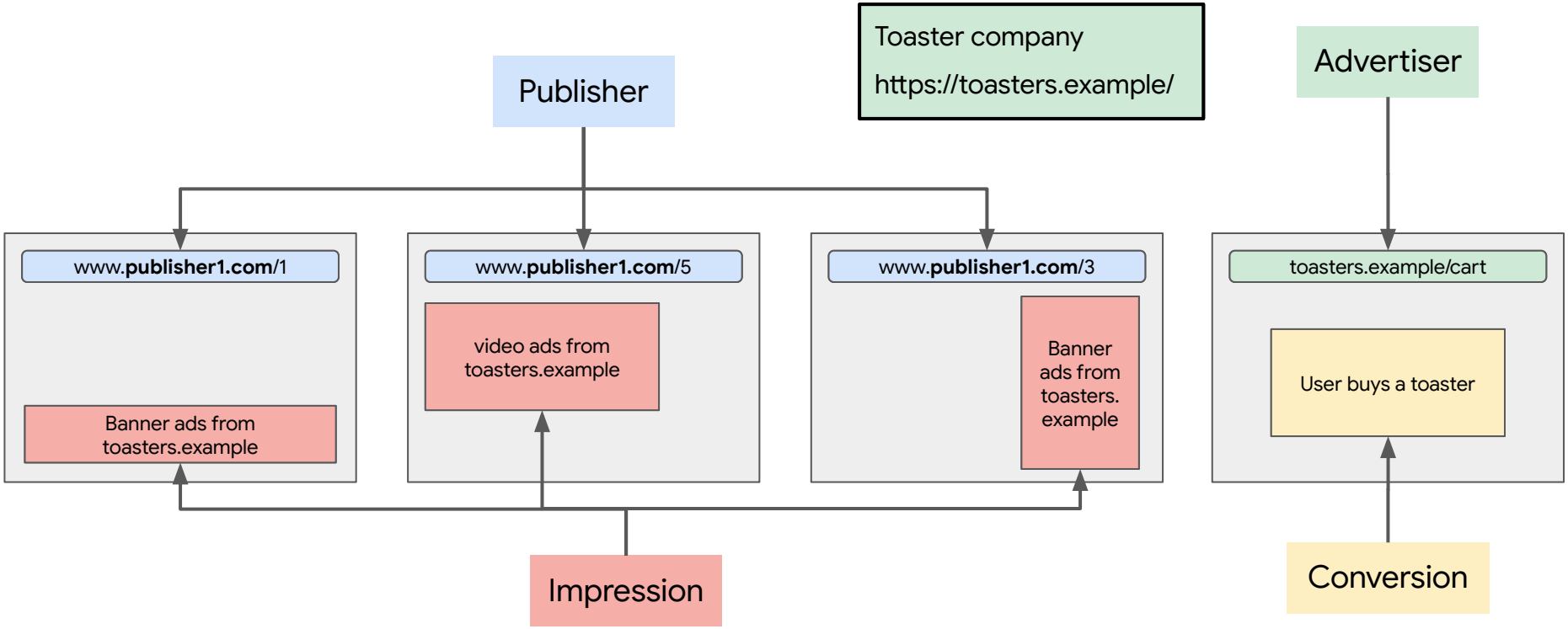
- Reach for entire ads campaign
- Reach for people in MA
- Reach for people in CA
- ...
- Reach for people in MA aged 20-30
- Reach for people in MA aged 30-40
- ...

Hierarchical Queries



...

Online Advertising: Example



Ads Attribution

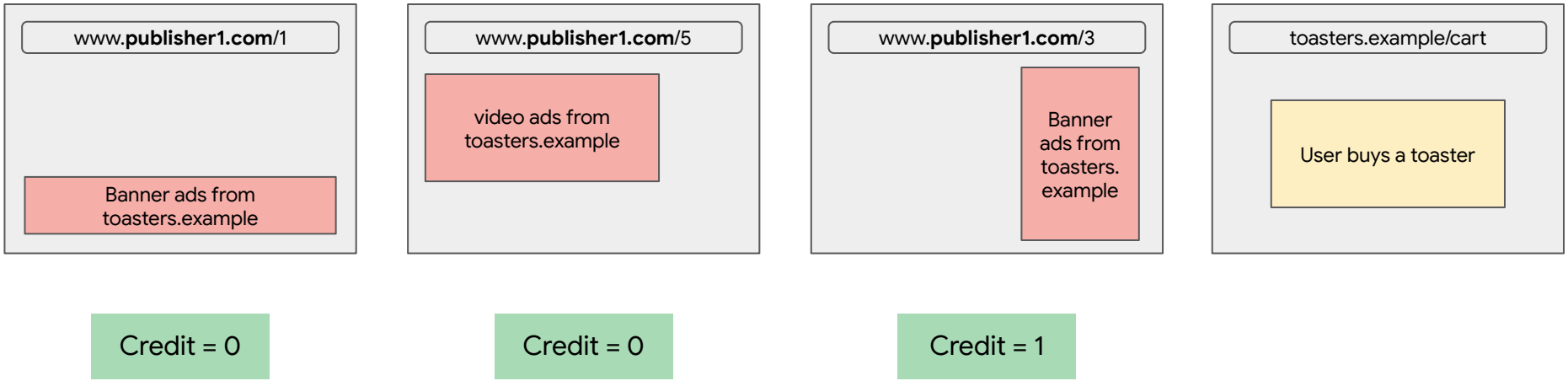
“Allocate the credit to the different impressions leading to a conversion”

Which impressions are more important for this conversion?

Ads Attribution

"Allocate the credit to the different impressions leading to a conversion"

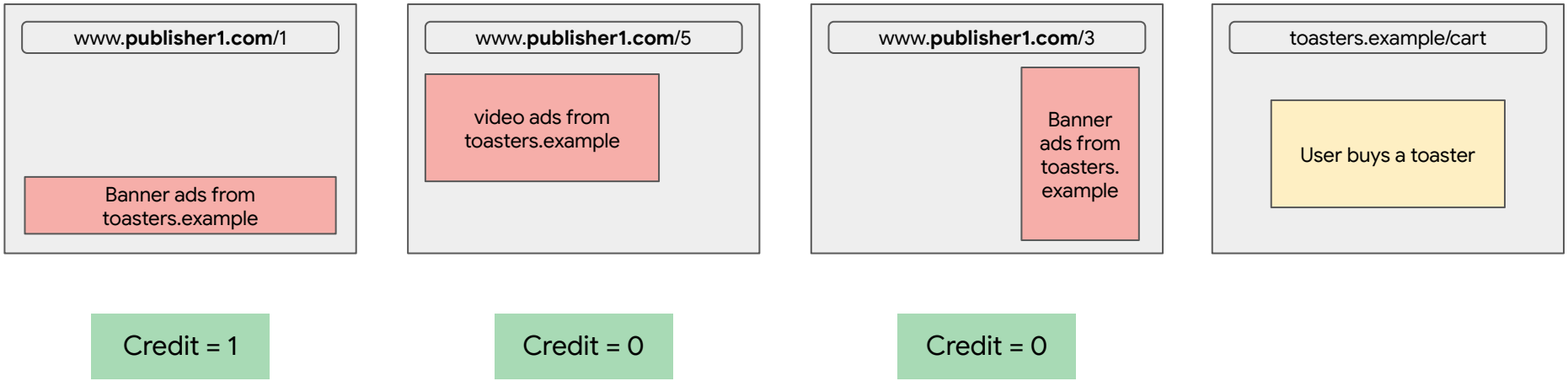
Last Touch Attribution (LTA): Assigns all credits to *most recent* impression



Ads Attribution

"Allocate the credit to the different impressions leading to a conversion"

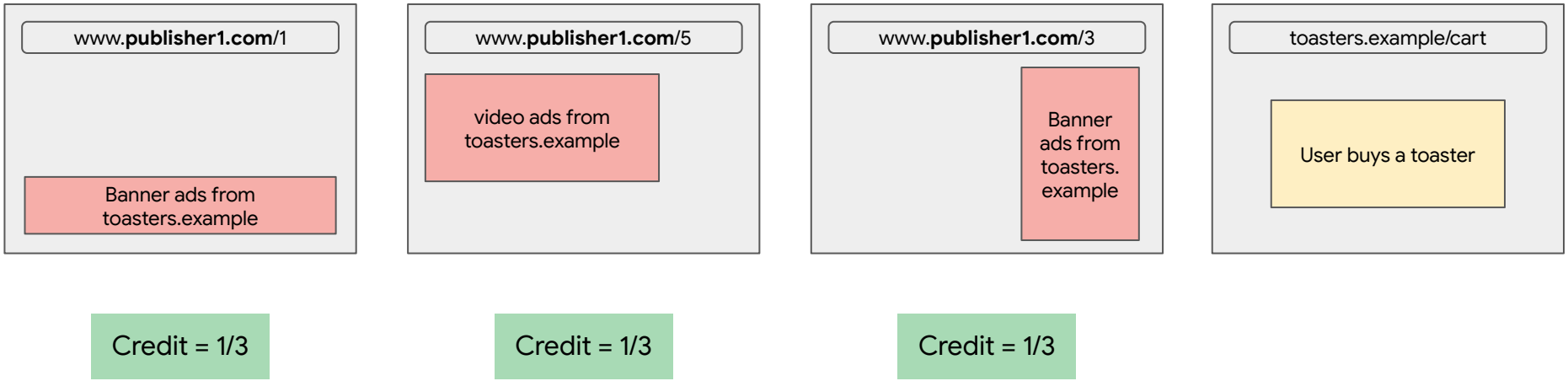
First Touch Attribution (LTA): Assigns all credits to *first* impression



Ads Attribution

"Allocate the credit to the different impressions leading to a conversion"

Linear (aka Uniform): Assigns credits *equally* among all impressions

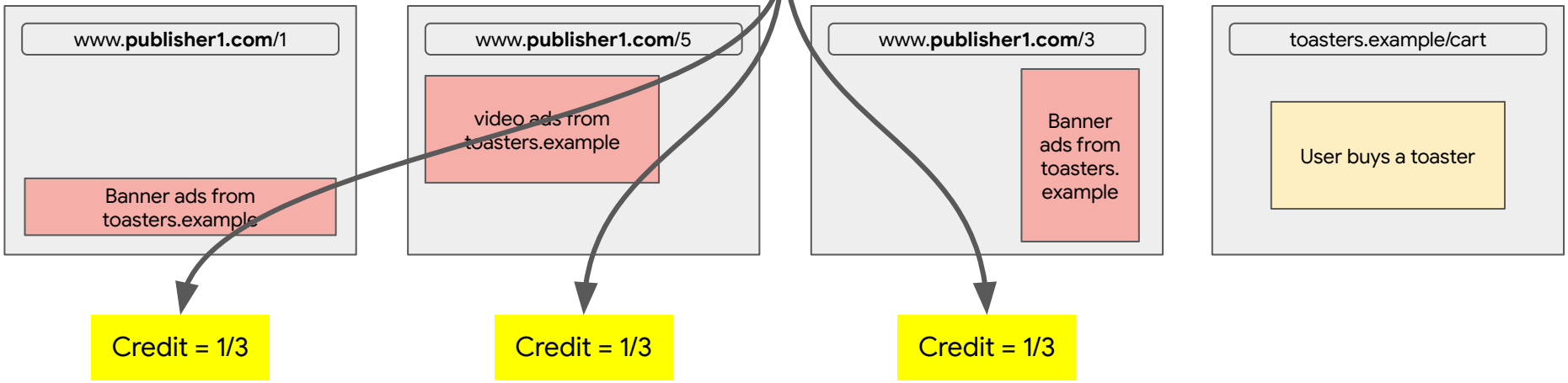


Ads Attribution

"Allocate the credit to the different impressions leading to a conversion"

Linear (aka Unif

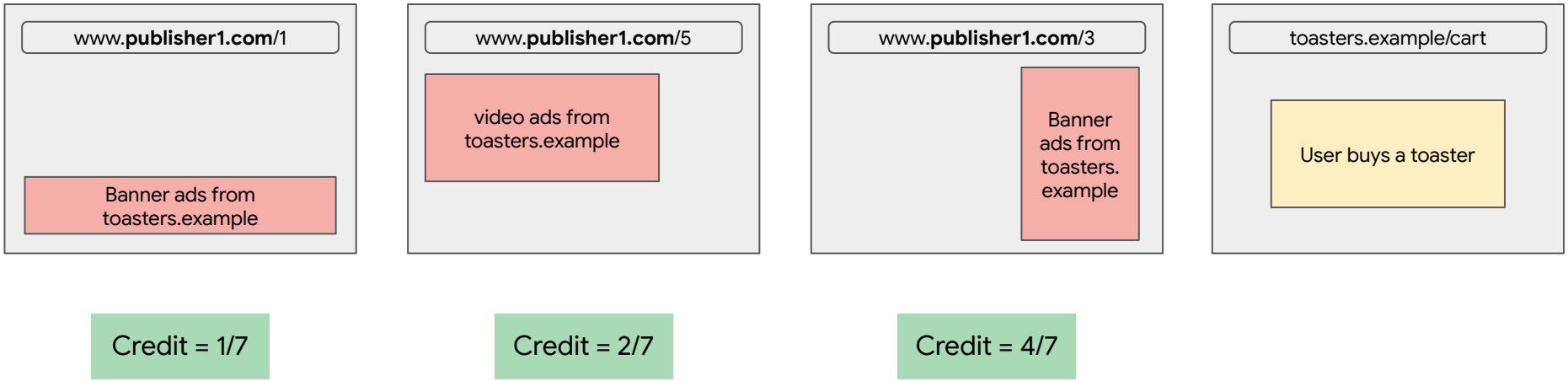
Multi-touch attribution: more than one impression gets credits



Ads Attribution

"Allocate the credit to the different impressions leading to a conversion"

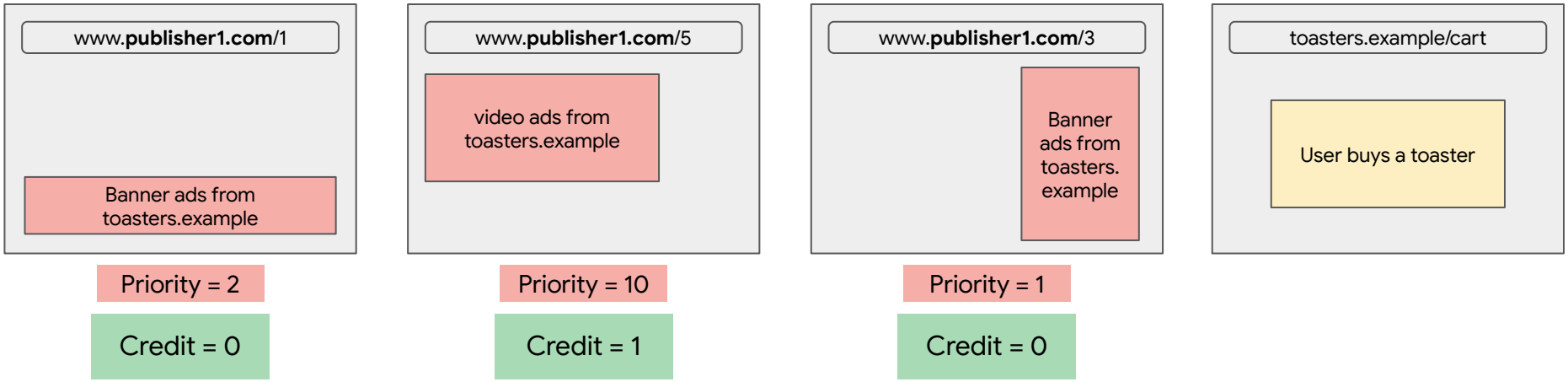
Exponential Time Decay: Assigns credits that *decreases exponentially* for older impressions



Ads Attribution

"Allocate the credit to the different impressions leading to a conversion"

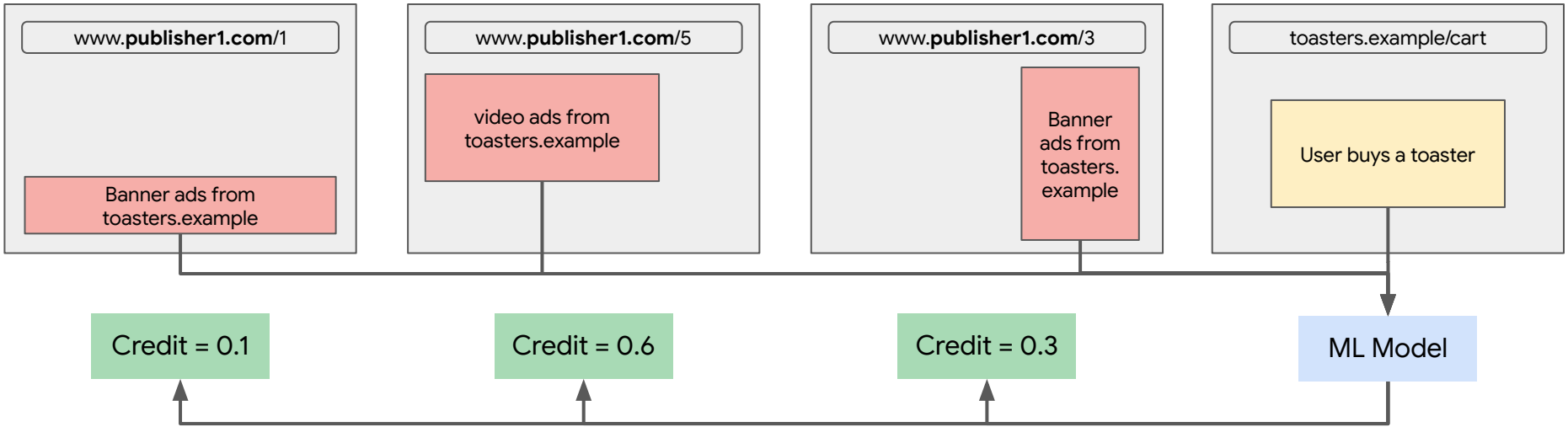
Prioritization: Each impression has *priority*; assign all credits to *highest-priority* impression



Ads Attribution

"Allocate the credit to the different impressions leading to a conversion"

Data-driven attribution (DDA): Train an *ML model* and use the model to assign credits

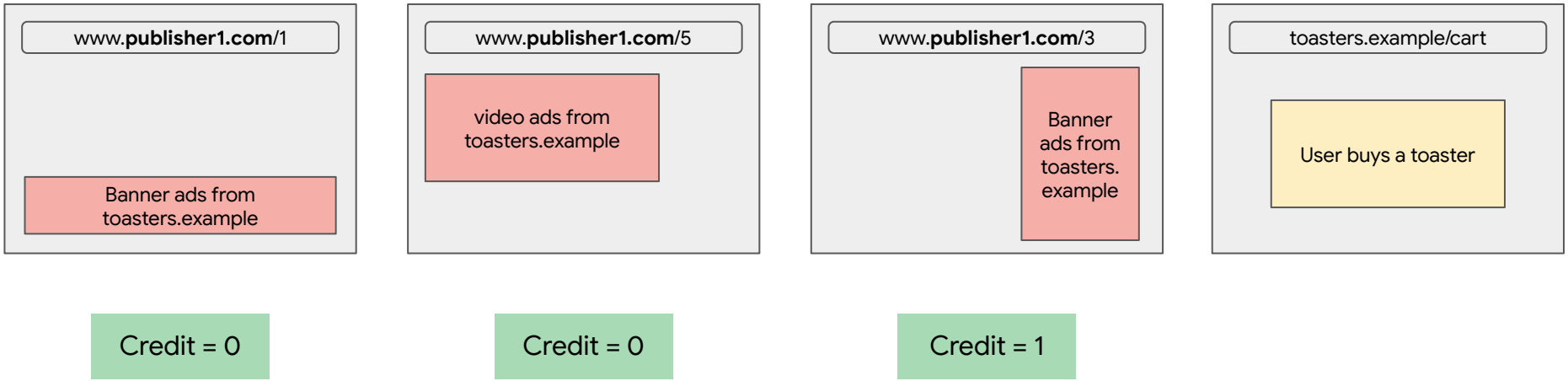


Ads Attribution

For simplicity, focus on *single-touch* attribution, i.e. when one impression gets all the credits

"Allocate the credit to the different impressions leading to a conversion"

Last Touch Attribution (LTA): Assigns all credits to *most recent* impression

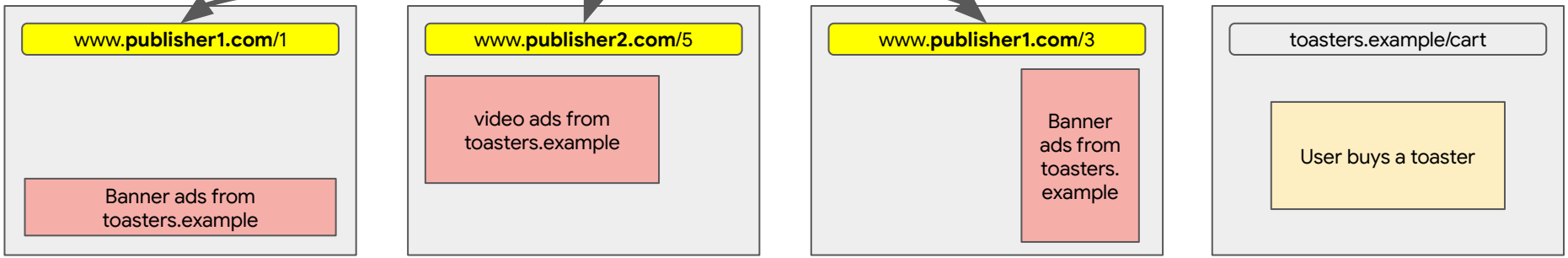


Ads Attribution

"Allocate the credit to the different impressions leading to a conversion"

Multiple publishers can be involved

Last Touch Attribution (LTA): Assigns all credits to *most recent* impression



Credit = 0

Credit = 0

Credit = 1

Ads Attribution Queries

Aggregated Statistics

- Conversion rate
  - % views or clicks that leads to conversion
- Average conversion value
- Can be sliced / hierarchical similar to reach / frequency queries

Event-Level Statistics

- Each impression:
  - Whether it leads to conversion
  - Total conversion value

Example queries

- Conversion value for entire ad campaign
- Conversion value for people in MA
- Conversion value for people in CA
- ...
- Conversion value for people in MA aged 20-30
- Conversion value for people in MA aged 30-40
- ...

Ads Attribution Queries

Aggregated Statistics

- Conversion rate
  - % views or clicks that leads to conversion
- Average conversion value
- Can be sliced / hierarchical similar to reach / frequency queries

Event-Level Statistics

- Each impression:
  - Whether it leads to conversion
  - Total conversion value

Used to:

- Business decision
  - Increase / decrease ads budget
- Build models
  - Predict conversion rate
  - Predict conversion value
  - ⇒ Used to determine bidding price in automatic ads auctions

Privacy & Measurements

Concerns around leakage of information about individual users through

- Released **statistics**
- Trained **ML models**
- **Intermediate tools**, e.g. 3PC, used for above purposes

Privacy & Measurements

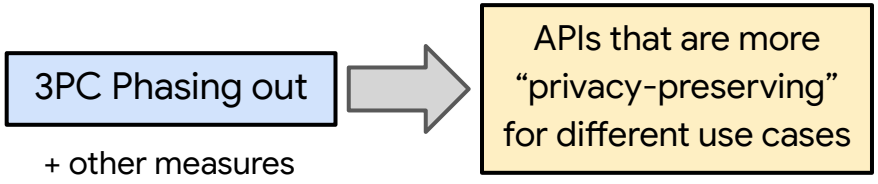
Concerns around leakage of information about individual users through

- Released **statistics**
- Trained **ML models**
- **Intermediate tools**, e.g. 3PC, used for above purposes

Ad measurement goals are ***aligned*** with differential privacy

- Interested in patterns that are large enough to impact business decisions, not individual users
- Prefer to be robust to outliers

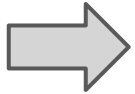
Anti-Tracking Efforts



Anti-Tracking Efforts

3PC Phasing out

+ other measures



APIs that are more "privacy-preserving" for different use cases

For Measurements

- **Without DP:**
  - Private click measurement (PCM)
- **With DP:**
  - Attribution Reporting API (ARA)
    - Event-level Report
    - Summary Report
  - MaskedLARK
  - Interoperable Private Attribution (IPA)
  - Private Ads Measurement (PAM)

Our focus

+ other use cases

For list of other proposals, see W3C github repo.

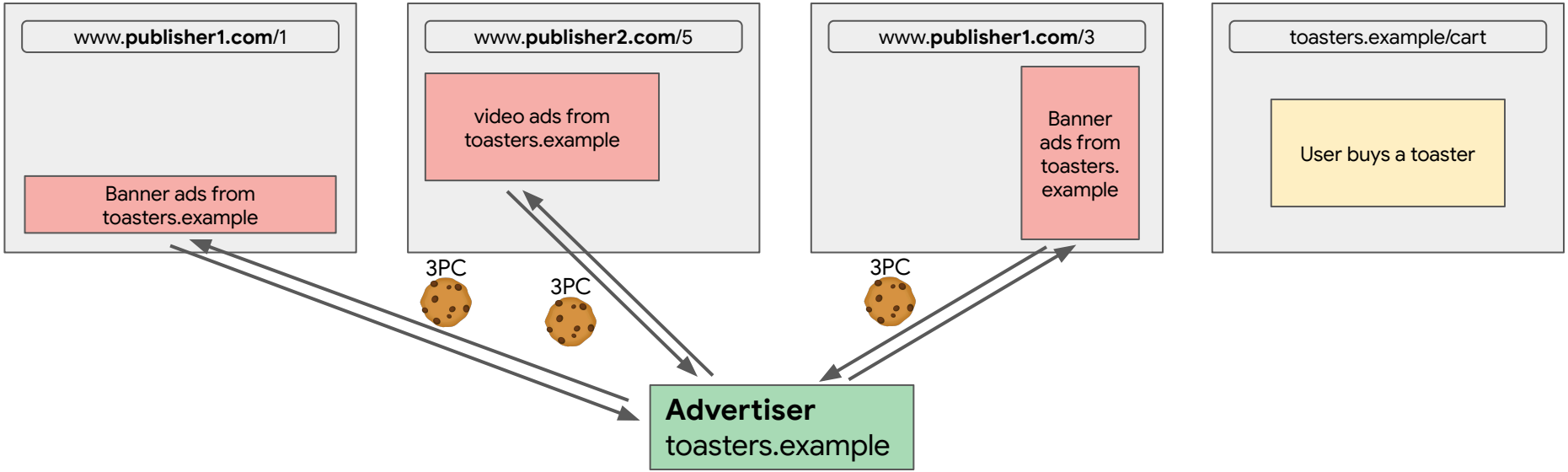
Third-Party Cookies (3PC) & Measurement

3PC ⇒ helps determine which impressions / conversions are from the same user

- ⇒ Reach / frequency measurements
- ⇒ Ads Attribution measurements

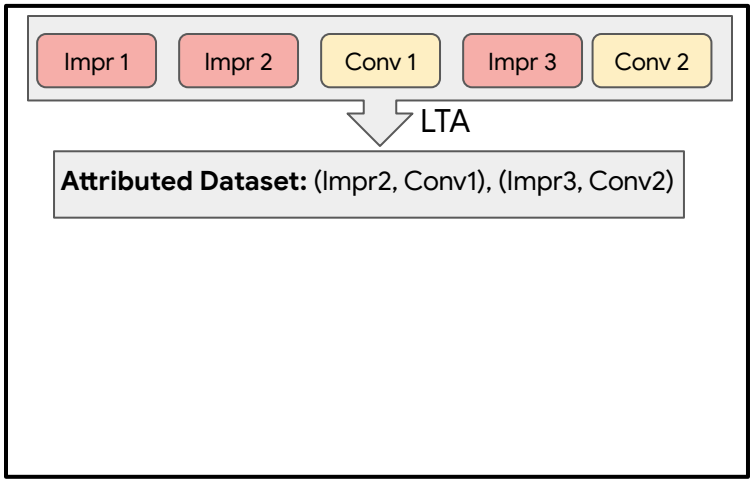
Privacy Concern

- Allows bad actors to track users across sites
  - I.e. know that the user visits publisher1 then publisher2 then … etc.



High-Level Overview of APIs: Aggregate

Device 1



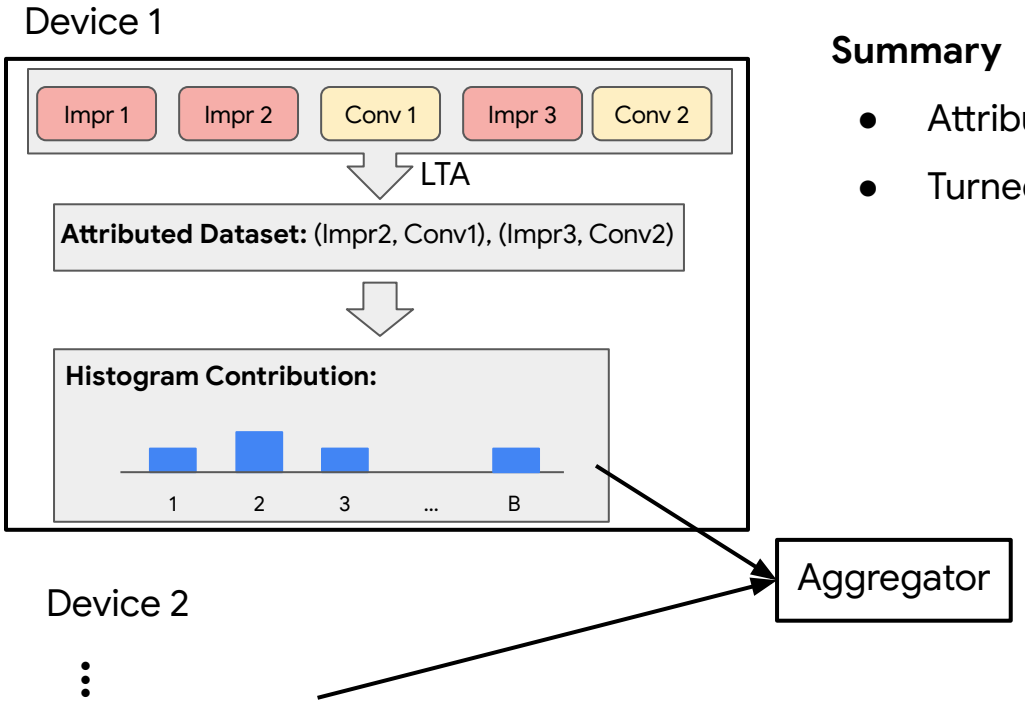
Device 2

⋮

Summary

- Attributions happen on device / browser

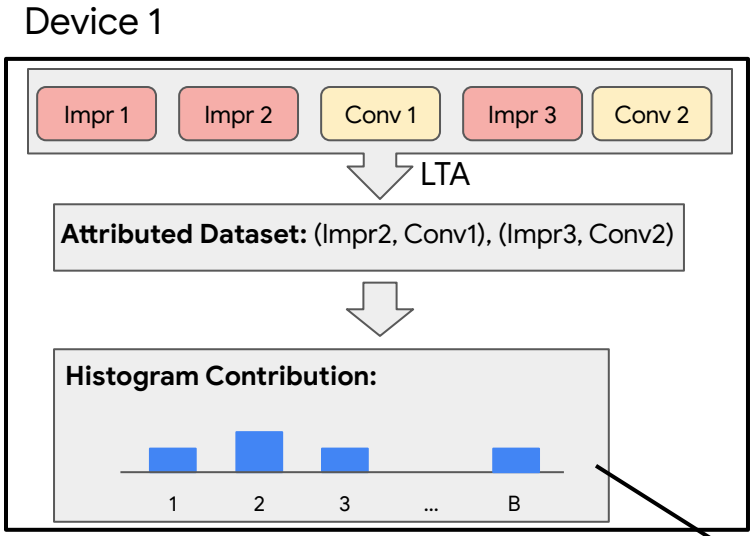
High-Level Overview of APIs: Aggregate



Summary

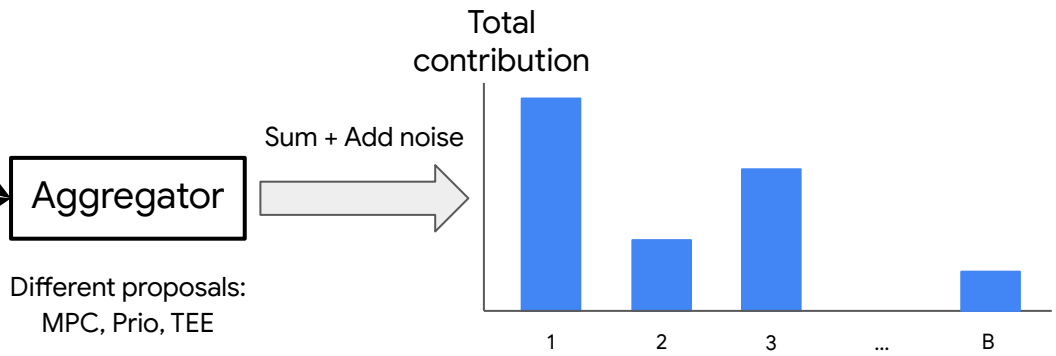
- Attributions happen on device / browser
- Turned into histogram contributions with capping

High-Level Overview of APIs: Aggregate

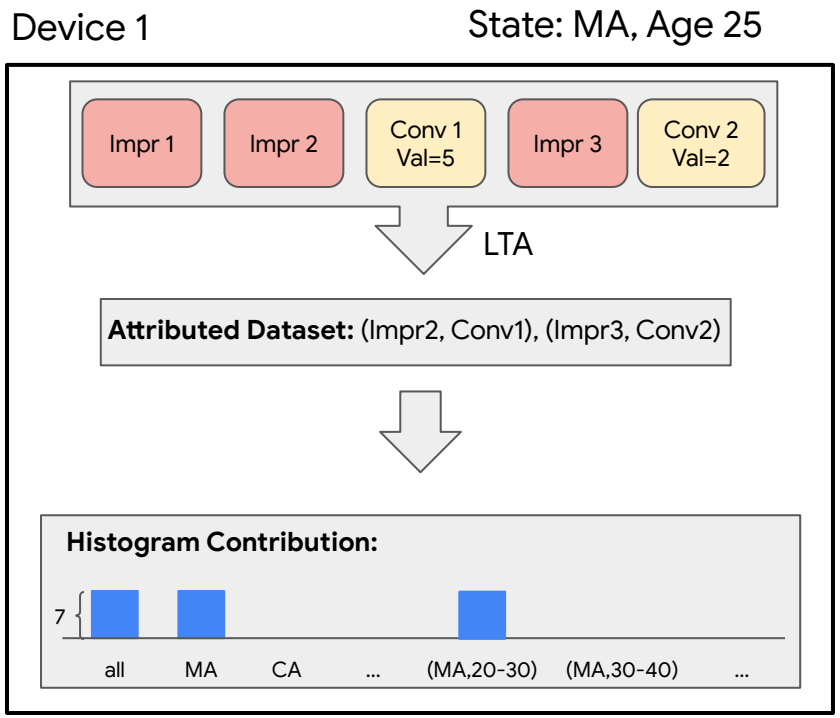


Summary

- Attributions happen on device / browser
- Turned into histogram contributions with capping
- Aggregator sums these contributions + add noise



Example Usage of Aggregate APIs

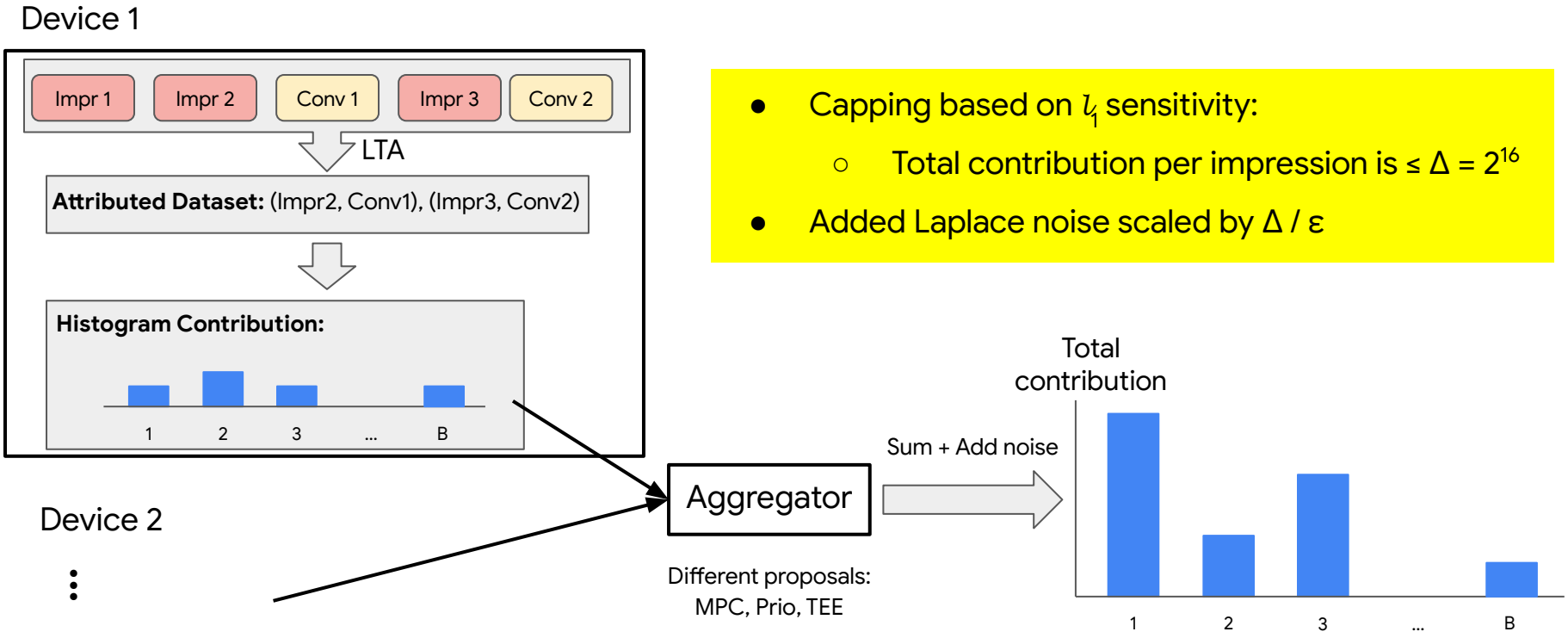


Example queries

- Conversion value for entire ad campaign
- Conversion value for people in MA
- Conversion value for people in CA
- ...
- Conversion value for people in MA aged 20-30
- Conversion value for people in MA aged 30-40
- ...

One bucket for each query

ARA Summary Reports



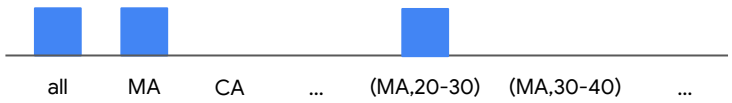
- Capping based on l_1 sensitivity:
  - Total contribution per impression is ≤ $\Delta = 2^{16}$
- Added Laplace noise scaled by Δ / ε

Optimizing Utility of ARA Summary Reports

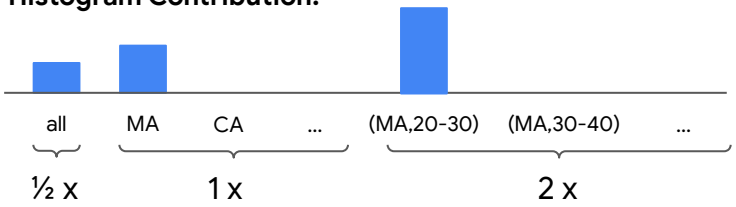
Budgeting via Rescaling

- Multiply bucket **i** contribution by a factor of $\mathbf{a_i}$
- Rescale back after aggregation
- ⇔ assigning "privacy budget" $\mathbf{a_i}$ to query **i**

Histogram Contribution:

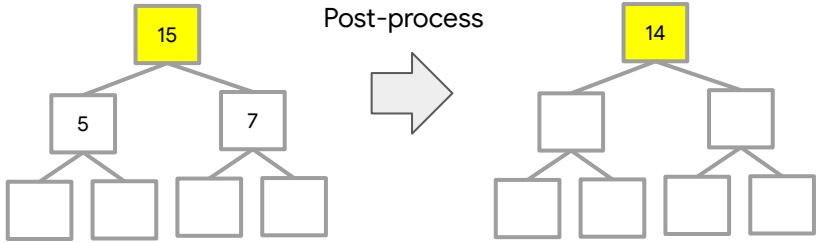


Histogram Contribution:



Post-processing

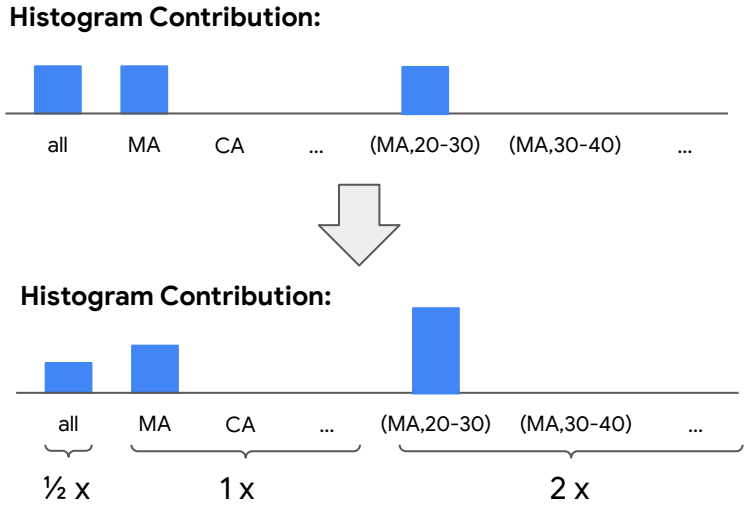
- Given the noisy measurements, can post-process to reduce error further *[Dawson et al.'23]*
- Exploit linear constraints–parent = sum of children
  - E.g. for top query, can take linear comb. between itself and sum of its children



Optimizing Utility of ARA Summary Reports

Budgeting via Rescaling

- Multiply bucket **i** contribution by a factor of $\mathbf{a_i}$
- Rescale back after aggregation
- ⇔ assigning "privacy budget" $\mathbf{a_i}$ to query **i**



Post-processing

- Given the noisy measurements, can post-process to reduce error further *[Dawson et al.'23]*
- Exploit linear constraints–parent = sum of children
  - E.g. for top query, can take linear comb. between itself and sum of its children
- Extends the methods of *[Hay et al., VLDB'10, Cormode et al., ICDE'12]* for regular trees
- Is a special case of the matrix mechanism *[Li et al., VLDB '15, Nikolov et al., STOC'13]*

Aggregate APIs: Challenges

- **Objective**
  - Aligning objective with downstream tasks
    - E.g. relative error
  - Study in DP literature has focused on additive errors
- **Optimization**
  - Require some historical data to perform parameter optimization
- **Time Delay**
  - May have to wait for a long time before getting report
- **Extending APIs to support more complex algorithms:**
  - Exponential mechanism
    - DP Synthetic data generation
  - Sparse Vector Technique
  - Propose-test-release

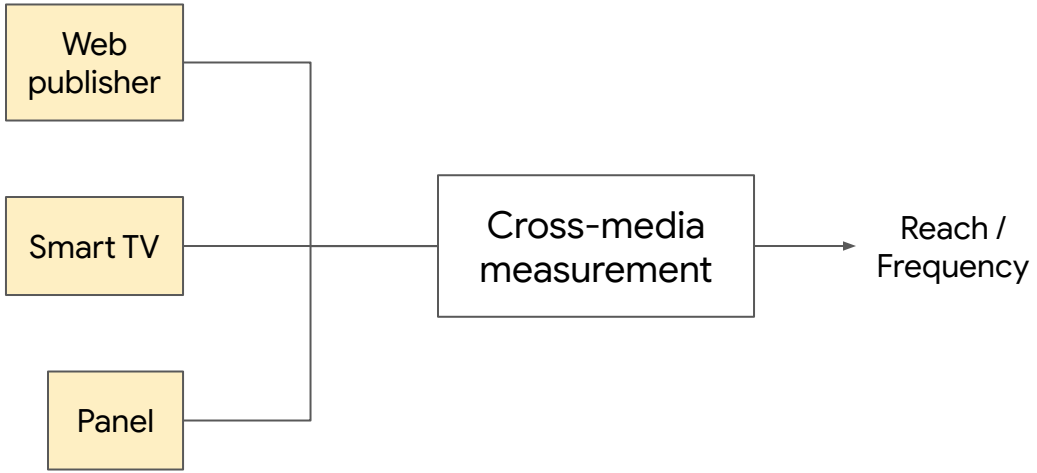
Cross-Media Measurement

Setting

- **p** publishers
- Publisher **i** has a multiset $\mathbf{S_i}$ of users it reaches

Goal

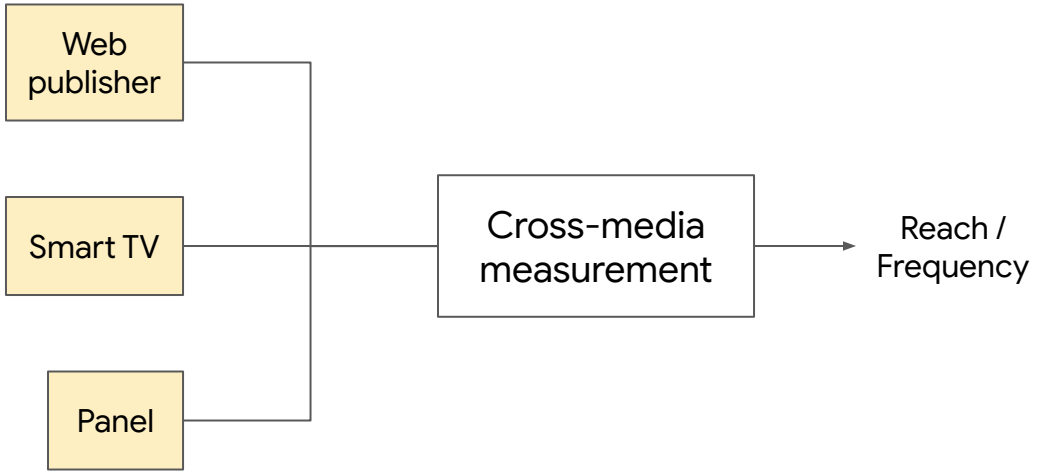
- **Reach**: number of unique users in the union of $S_1, \ldots, S_p$
- **k-Frequency**: the fraction of users that are reached k times across all publishers.



Project Halo

Background & Goals

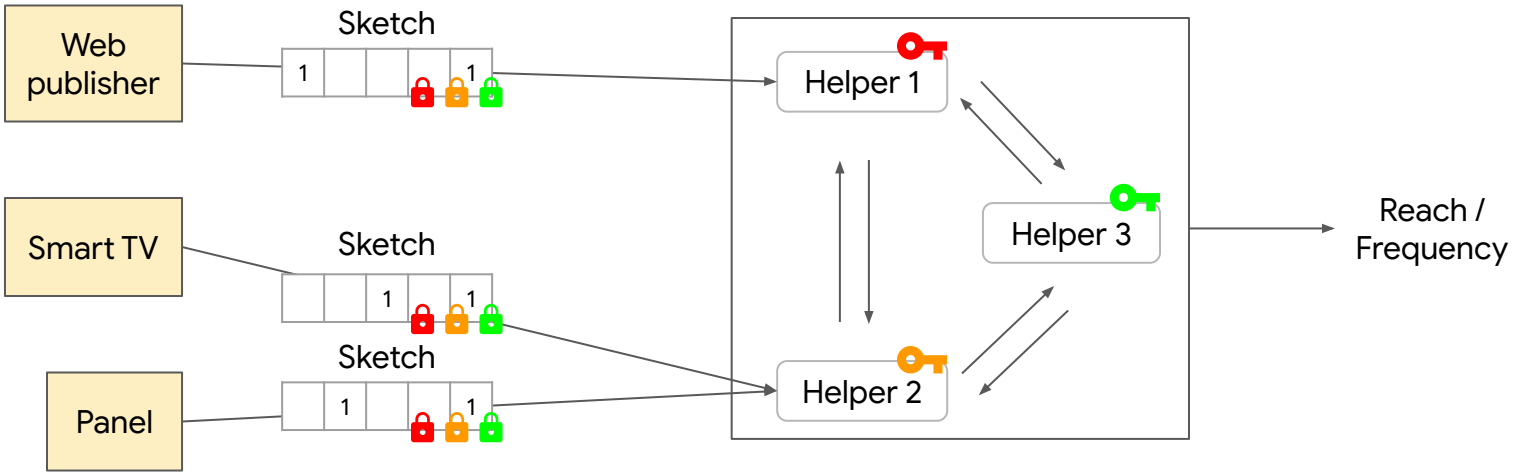
- Led by *World Federation of Advertisers (WFA)*
- Allow advertisers to measure *cross-platform* ads campaign performances
  - Respecting user's privacy
- Scope is quite large
  - Reach / Frequency measurements
  - Deduplication of users
  - Cross-publisher data metrics, APIs
  - Planning ...
- Open-source (see here)
- More details here



Overview of Solution

Privacy Guarantee
View of all-but-one helpers (& output) is computationally user-level DP.

- Uses sketching to reduce communication complexity
- Helpers merge the sketch, add noise and estimate reach / frequency



Full protocol published and presented at PoPETs 2022

Private Ads Modeling

Ad Modeling for Bidding

Bidding

- **pCVR:** predicting if an impression will result in an attributed conversion
- **pConvs:** predicting # of conversions attributed to impression
- **pValue:** predicting $ conversion value attributed to impression
- **pCTR:** predicting if an ad will be clicked
- Prediction signals used as inputs to bidding in ad auctions
- Models are huge; e.g., billions of parameters
- Data is sparse and class-imbalanced

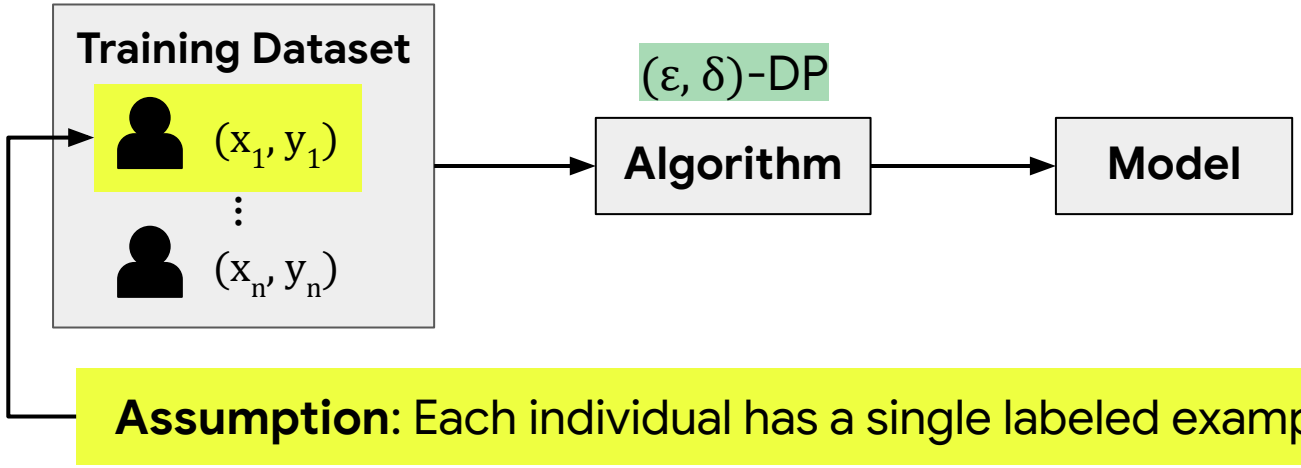
One Commonly Used Metric:

- AUC (Area under the Curve):
  - Plot *True Positive Rate* against *False Positive Rate*, & compute area under it.
  - AUC = probability that the classifier ranks a randomly chosen positive example higher than a randomly chosen negative one.
  - Random guessing: AUC = 0.5
  - Perfect prediction utility: AUC = 1
- AUC-Loss = 1 - AUC
- Relative Change in AUC-Loss (in %) = (AUC-Loss - $\text{AUC-Loss}_{\text{baseline}}$) / $\text{AUC-Loss}_{\text{baseline}}$ * 100
  - baseline is without DP

Full DP

Training ML Model

Ensure that the model is privacy-safe to be used in downstream tasks



Substitution-DP

$X \asymp^s X'$ ⟺ X is X' with single training example changed

DP-SGD

Gradient Descent

Training data X

Labeled Samples

$(x_1, y_1), \ldots, (x_n, y_n)$

Loss function: $\ell(\hat{y}, y) \in \mathbb{R}$

Empirical loss

$$\mathcal{L}_w(X) := \Sigma_{i\in[n]} \ \ell(f_w(x_i), y_i) \ / \ n$$

Training Objective

Find w that minimizes $\mathcal{L}_w(X)$

Gradient

$$\nabla_w \mathcal{L}(X) = [d\mathcal{L}(X) / dw_1, \ldots, d\mathcal{L}(X) / dw_d]$$

$$\nabla_w \mathcal{L}(X) = \Sigma_{i\in[n]} \nabla_w \ell(f_w(x_i), y_i) \ / \ n$$

η_t: learning rate

Gradient Descent (GD)

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$:

$$w_t \leftarrow w_{t-1} - \eta_t \nabla_w \mathcal{L}(X)$$

Return w_T

Gradient Descent

Gradient Descent (GD)

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$:

$$w_t \leftarrow w_{t-1} - \eta_t \left(\Sigma_{i\in[n]} \nabla_w \ell(f_w(x_i), y_i)\right) / n$$

Return w_T

Gaussian Mechanism

$$\sigma = \frac{2\sqrt{2\ln(2/\delta)}}{\epsilon} \cdot \Delta_2(g)$$

$$X = (x_1, \ldots, x_n) \rightarrow \boxed{\text{Analyzer}} \rightarrow g(X) + \mathcal{N}(0, \sigma^2)^{\otimes d}$$

- Use l_2-sensitivity
- Each coordinate is independent

Theorem Assuming $\text{Range}(f) \subseteq \mathbb{R}^d$ and $\varepsilon \leq 1$, Gaussian Mechanism is (ε, δ)-DP.

Assumption: $\text{Range}(g) \subseteq \mathbb{R}^d$

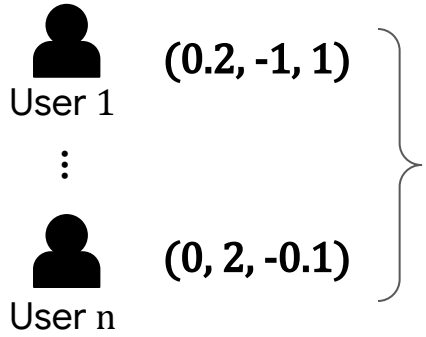
(ie, g is vector-valued with real entries)

Gaussian Distribution

For every real number z,

$$\text{PDF}_{\mathcal{N}(0,\sigma^2)}(z) \propto e^{-(z/\sigma)^2}$$

Examples: Vector summation



Assumption: Each vector has l_2-norm $\leq C$ ⇒ $\Delta_2(g) \leq 2C$

Gradient Descent

Additional parameters
- σ: noise standard deviation

Add Gaussian noise to average gradients!

Gradient Descent (GD)

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$

$$w_t \leftarrow w_{t-1} - \eta_t \left(\mathcal{N}(0, \sigma^2 \cdot I) + \Sigma_{i \in [n]} \nabla_w \ell(f_w(x_i), y_i)\right) / n$$

Return w_T

Not DP: each $\nabla_w \ell(f_w(x_i), y_i)$ can be arbitrarily large!

Clipping Trick

Additional parameters
- σ: noise standard deviation
- C: clipping norm

DP-GD

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$:

For $i = 1,\ldots, n$:

$v_i = \nabla_w \ell(f_w(x_i), y_i)$

$\tilde{v}_i = v_i \cdot \min(1, C / \|v_i\|_2)$

$w_t \leftarrow w_{t-1} - \eta_t \left(\mathcal{N}(0, \sigma^2 \cdot I) + \Sigma_{i \in [n]} \tilde{v}_i\right) / n$

Return w_T

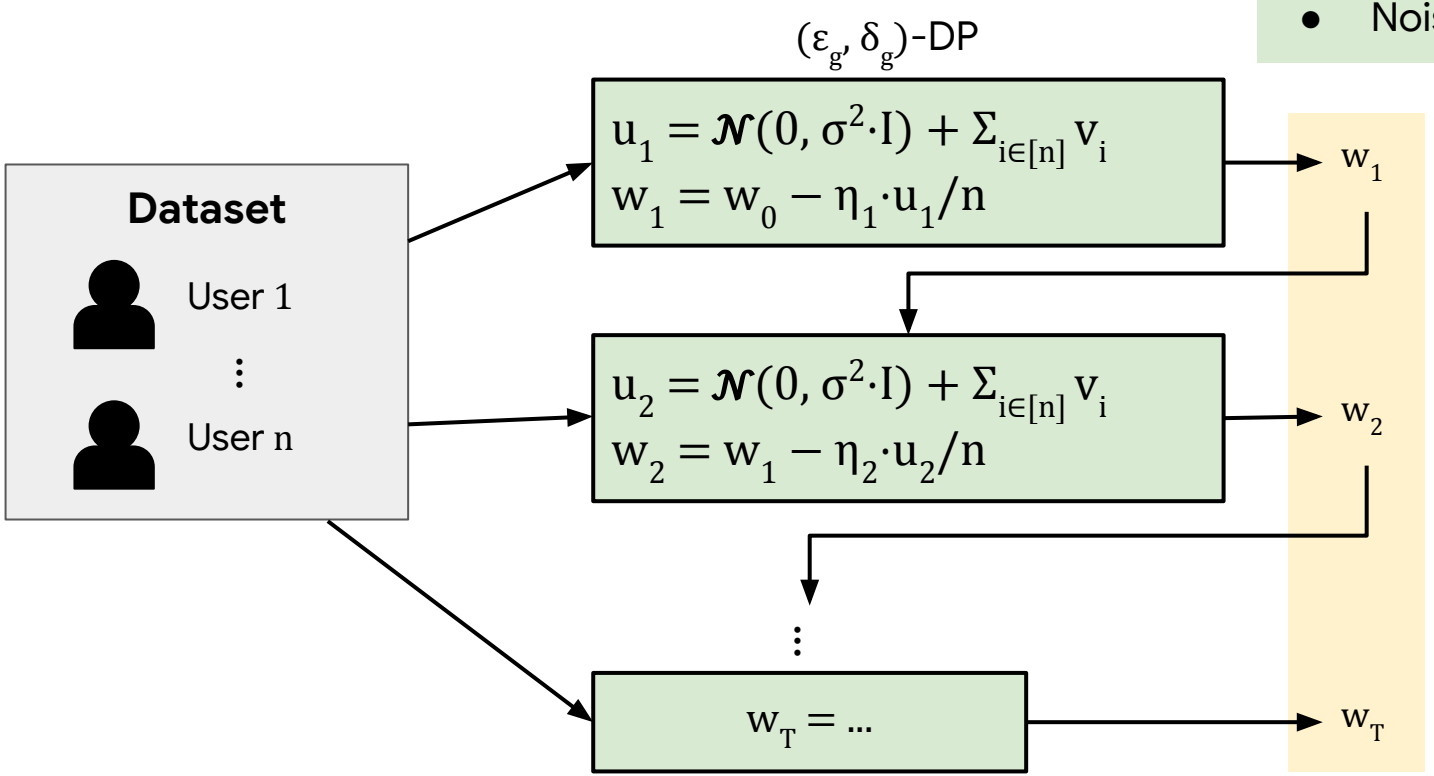
Enforce: gradient norm bound at most C

Not DP: each $\nabla_w \ell(f_w(x_i), y_i))$ can be arbitrarily large!

"If $\nabla_w \ell(f_w(x_i), y_i))$ is too large, rescale it to be smaller"
(bias-variance tradeoff)

DP-GD: Privacy Analysis

Gaussian mechanism:
- ℓ_2-sensitivity $\leq 2C$
- Noise std σ



T-fold basic / advanced composition

Entire Alg is (ε, δ)-DP

Actually can release all of $w_1, \ldots, w_T$!

DP-GD: Privacy Analysis

DP-GD

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$:

For $i = 1, \ldots, n$:

$$v_i = \nabla_w \ell(f_w(x_i), y_i)$$

$$\tilde{v}_i = v_i \cdot \min(1, C / \|v_i\|_2)$$

$$w_t \leftarrow w_{t-1} - \eta_t \left(\mathcal{N}(0, \sigma^2 \cdot I) + \Sigma_{i \in [n]} \tilde{v}_i\right) / n$$

Return w_T

Each iteration

Gaussian mechanism:

- ℓ_2-sensitivity $\leq 2C$
- Noise std σ
- $\Rightarrow (\varepsilon_g, \delta_g)$-DP

T-fold basic / advanced composition

Entire Algorithm: (ε, δ)-DP

Gradient Descent & Friends

Gradient Descent (GD)

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$:

$$w_t \leftarrow w_{t-1} - \eta_t \nabla_w \mathcal{L}(X)$$

Return w_T

Stochastic GD (SGD)

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$:

$i \leftarrow$ random example index

$$w_t \leftarrow w_{t-1} - \eta_t \nabla_w \ell(f_w(x_i), y_i)$$

Return w_T

Mini-batch SGD

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$:

$S \leftarrow$ random index set of size B

$$w_t \leftarrow w_{t-1} - \eta_t \Sigma_{i \in S} \nabla_w \ell(f_w(x_i), y_i) / B$$

Return w_T

Generalizes both GD, SGD

DP-SGD *[Abadi et al'16]*

Modifications to achieve DP

- Add Gaussian noise to average gradients!

Mini-batch SGD

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$

$S \leftarrow$ random index set of size B

$w_t \leftarrow w_{t-1} - \eta_t \Sigma_{i \in S} \nabla_w \ell(f_w(x_i), y_i) / B$

Return w_T

DP-SGD *[Abadi et al'16]*

Modifications to achieve DP

- Add Gaussian noise to average gradients!

Additional parameters

- σ: noise standard deviation

Mini-batch SGD

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$

$S \leftarrow$ random index set of size B

$$w_t \leftarrow w_{t-1} - \eta_t \left(\mathcal{N}(0, \sigma^2 \cdot I) + \Sigma_{i \in S} \nabla_w \ell(f_w(x_i), y_i)\right) / B$$

Return w_T

Issue: gradient can be very large!

DP-SGD *[Abadi et al'16]*

Modifications to achieve DP

- Add Gaussian noise to average gradients!
- Clip each gradient to bound its norm

Additional parameters

- σ: noise standard deviation
- C: clipping norm bound

Mini-batch SGD

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$

  $S \leftarrow$ random index set of size B

  For $i \in S$:

    $v_i = \nabla_w \ell(f_w(x_i), y_i)$

    $\tilde{v}_i = v_i \cdot \min(1, C / \|v_i\|_2)$

  $w_t \leftarrow w_{t-1} - \eta_t \left(\mathcal{N}(0, \sigma^2 \cdot I) + \Sigma_{i \in S} \tilde{v}_i\right) / B$

Return w_T

Enforce: gradient norm bound at most C

DP-SGD *[Abadi et al'16]*

Mini-batch DP-SGD

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$:

$S \leftarrow$ random index set of size B

For $i \in S$:

$$v_i = \nabla_w \boldsymbol{\ell}(f_w(x_i), y_i)$$

$$\tilde{v}_i = v_i \cdot \min(1, C / \|v_i\|_2)$$

$$w_t \leftarrow w_{t-1} - \eta_t \left(\boldsymbol{\mathcal{N}}(0, \sigma^2 \cdot I) + \Sigma_{i \in S} \tilde{v}_i\right) / B$$

Return w_T

DP-SGD: Privacy Analysis

Mini-batch DP-SGD

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$:

$S \leftarrow$ random index set of size B

For $i \in S$:

$$v_i = \nabla_w \ell(f_w(x_i), y_i)$$

$$\tilde{v}_i = v_i \cdot \min(1, C / \|v_i\|_2)$$

$$w_t \leftarrow w_{t-1} - \eta_t \left(\mathcal{N}(0, \sigma^2 \cdot I) + \Sigma_{i \in S} \tilde{v}_i\right) / B$$

Return w_T

Each iteration

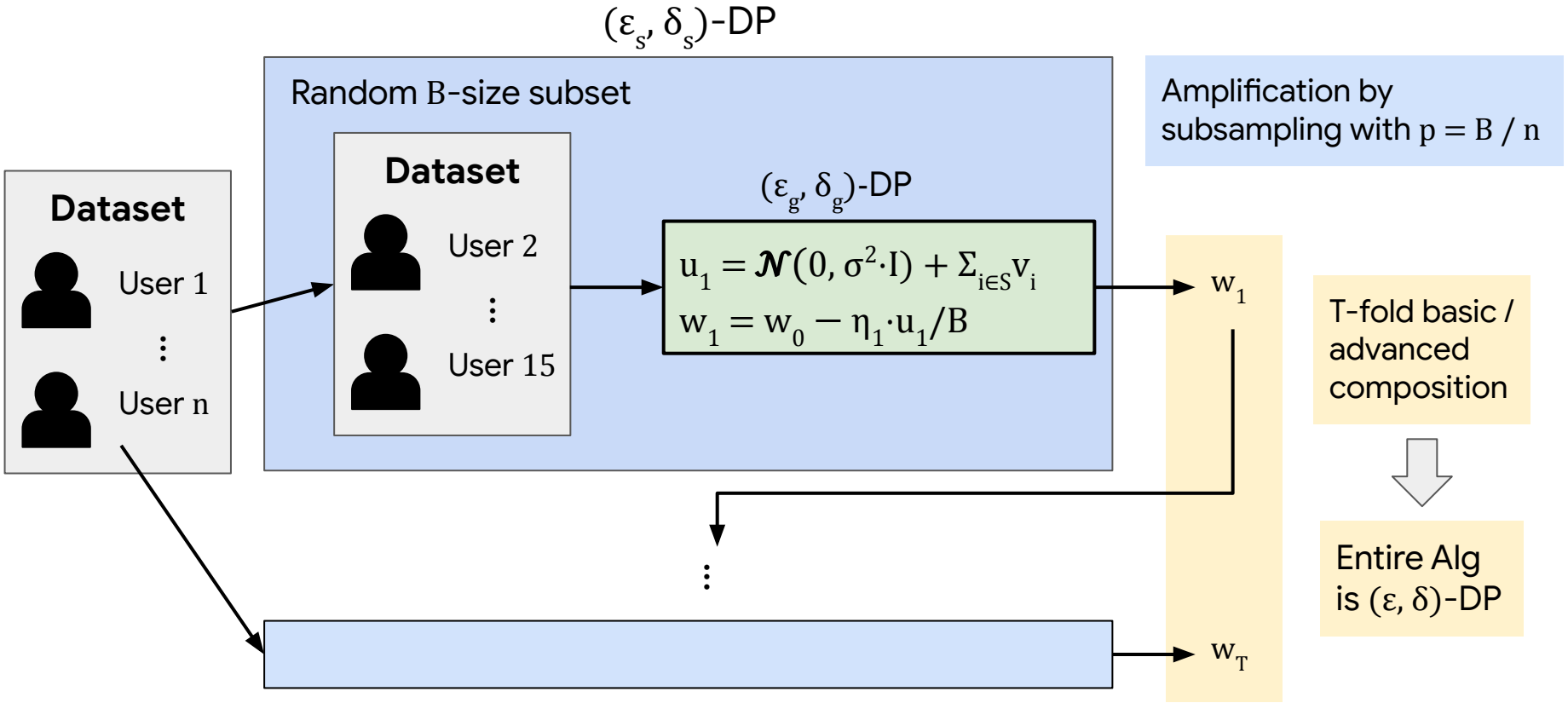
Gaussian mechanism:

- ℓ_2-sensitivity $\leq 2C$
- Noise std σ
- $\Rightarrow (\varepsilon_g, \delta_g)$-DP

T-fold basic / advanced composition

Entire Algorithm: (ε, δ)-DP

DP-SGD: Privacy Analysis



DP-SGD: Privacy Analysis

Mini-batch SGD

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$

$S \leftarrow$ random index set of size B

For $i \in S$:

$$v_i = \nabla_w \boldsymbol{\ell}(f_w(x_i), y_i)$$

$$\tilde{v}_i = v_i \cdot \min(1, C / \|v_i\|_2)$$

$$w_t \leftarrow w_{t-1} - \eta_t \left(\mathcal{N}(0, \sigma^2 \cdot I) + \Sigma_{i \in S} \tilde{v}_i\right) / B$$

Return w_T

Each iteration

Gaussian mechanism:

- $\boldsymbol{\ell}_2$-sensitivity $\leq 2C$
- Noise std σ
- $\Rightarrow (\varepsilon_g, \delta_g)$-DP

T-fold basic / advanced composition

Entire Algorithm: (ε, δ)-DP

DP-SGD: Privacy Analysis

Mini-batch SGD

$w_0 \leftarrow$ initial parameter

For $t = 1,\ldots,T$

$S \leftarrow$ random index set of size B

For $i \in S$:

$$g_i = \nabla_w \ell(f_w(x_i), y_i)$$

$$g_i = g_i \cdot \min(1, C / \|g_i\|_2)$$

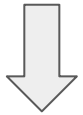
$$w_t \leftarrow w_{t-1} - \eta_t \left(\mathcal{N}(0, \sigma^2 \cdot I) + \Sigma_{i \in S} g_i\right) / B$$

Return w_T

Each iteration

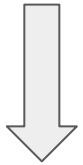
Gaussian mechanism:

- ℓ_2-sensitivity $\leq 2C$
- Noise std σ
- $\Rightarrow (\varepsilon_g, \delta_g)$-DP



Amplification by subsampling with $p = B / n$

Each iteration: $(\varepsilon_s, \delta_s)$-DP



T-fold basic / advanced composition

Entire Algorithm: (ε, δ)-DP

Merits of DP-SGD

- Generic recipe
- Broadly applicable

Possible improvements

- Gradient sparsity *[Ghazi et al.'23]*
- Privacy accounting

- Might incur (large) utility drop
- Increased training time

TensorFlow Privacy

Opacus (PyTorch)

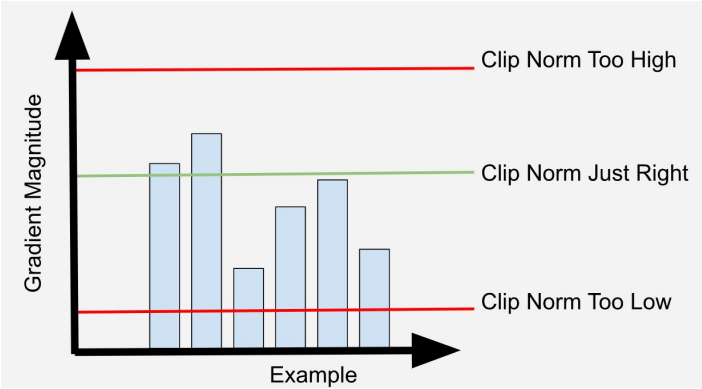
Ads Modeling *[Denison et al,'23]*

- Noise only added once per batch
  - Bigger batches ⇒ Less noise per example
- Large batches often take more epochs to converge

Ads Modeling *[Denison et al,'23]*

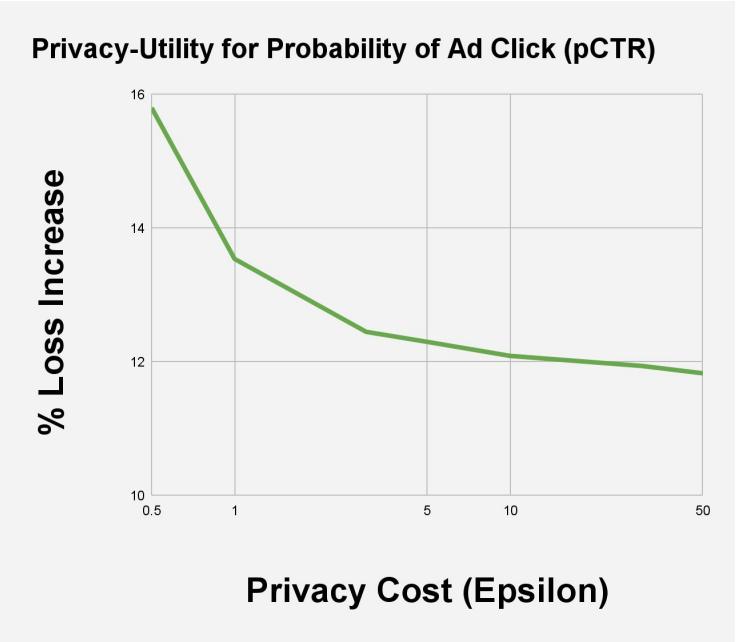
Clipping is a bias-variance tradeoff

- Noise is scaled with clip norm
- Clipping gradients loses signal
- Tune clip norm using fixed batch size



Ads Modeling *[Denison et al,'23]*

- Competitive Loss with DP
  - +12.1% Loss @ ε = 10
  - +13.5% Loss @ ε = 1
  - +15.8% Loss @ ε = 0.5
- Compute needs increased by 20%

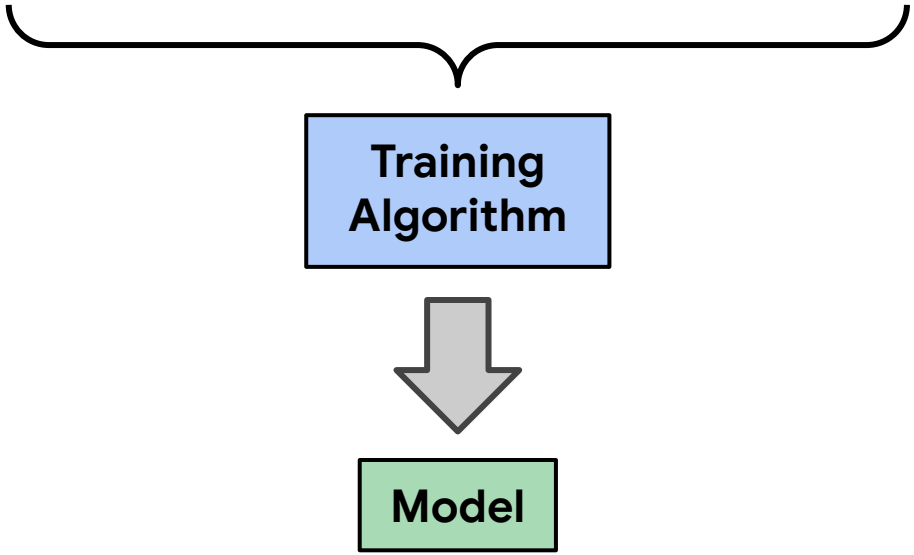


Label DP

Learning with DP

unknown data

(x_1, y_1) (x_2, y_2) ... (x_n, y_n)



$\mathbf{X} \asymp^s \mathbf{X}'$ $\Leftrightarrow$ $\mathbf{X}$ is $\mathbf{X}'$ with single input labeled example changed

(ε, δ)-DP *[Dwork et al.'06]*

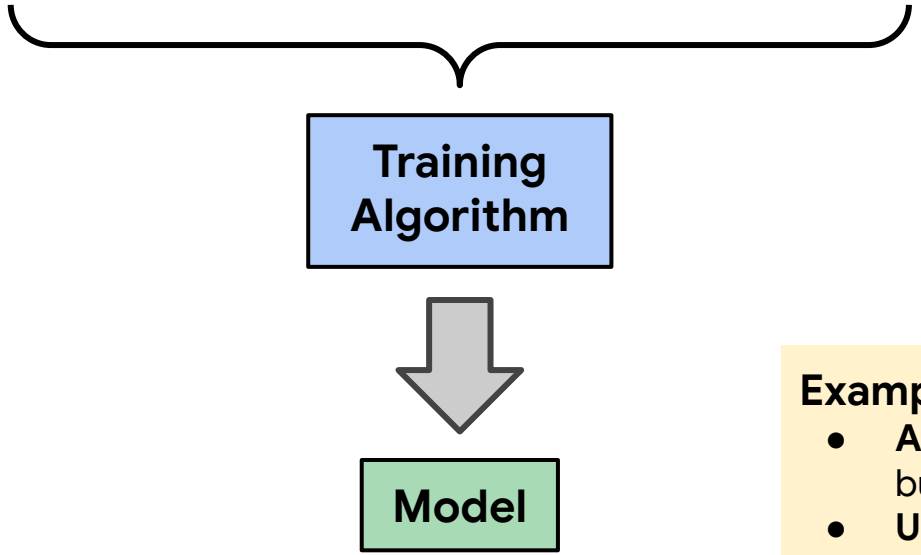
For every $\mathbf{X} \asymp^s \mathbf{X}'$ and every set S of models,

$$\Pr[A(\mathbf{X}) \in S] \leq e^{\varepsilon} \cdot \Pr[A(\mathbf{X}') \in S] + \delta$$

Learning with *Label* DP *[Chaudhuri-Hsu '11]*

unknown data

(x_1, y_1) (x_2, y_2) ... (x_n, y_n)



$\mathbf{X} \asymp^s \mathbf{X'} \Leftrightarrow \mathbf{X}$ is $\mathbf{X'}$ with label changed

(ε, δ)-DP *[Dwork et al.'06]*
For every $\mathbf{X} \asymp^s \mathbf{X'}$ and every set S of models,

$$\Pr[A(\mathbf{X}) \in S] \leq e^{\varepsilon} \cdot \Pr[A(\mathbf{X'}) \in S] + \delta$$

Examples:

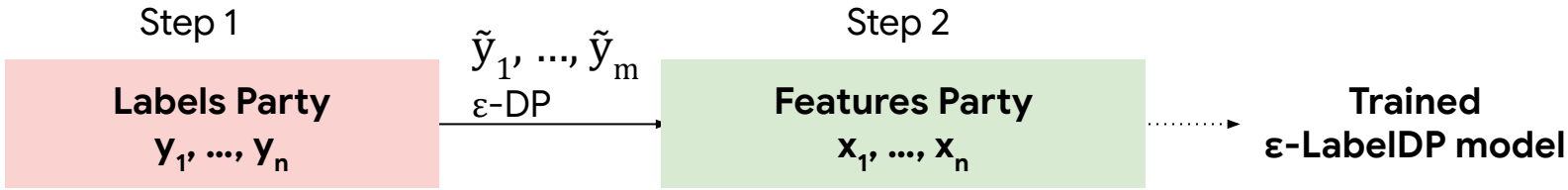
- **Ads:** Adtech already knows the impression, but conversion is unknown
- **User surveys:** basic demographic info is known but, eg, income is unknown

Label DP Algorithms

- RR *[Warner'65]*
- RR-with-Prior *[Ghazi et al. '21]*
- RR-on-Bins *[Ghazi et al. '23]*
- Unbiased Mechanism *[Badanidiyuru et al.'23]*
- Logistic Regression *[Meta'21]*
- PATE-FM *[Malek et al. '21]*
- ALIBI *[Malek et al. '21]*
- Clustering *[Esfandiari et al. '22]*

Our focus

Feature-Oblivious Label DP



- Generate DP (noisy) labels $\tilde{y}_1, \ldots, \tilde{y}_m$ using M
- Run standard training to minimize loss using noisy labels:

$$\Sigma_{i \in [n]} \ell(f_w(x_i), \tilde{y}_i))$$

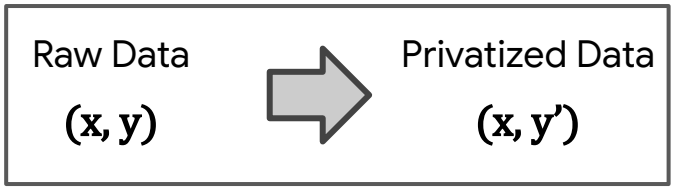
- Laplace Mechanism on each y_i
- Randomized Response on each y_i

Choose ε-DP mechanism M to minimize $\mathbf{E}_{y, \tilde{y} \sim M(y)} L(\tilde{y}, y)$

- Privately learn a prior over labels y, to construct M
- For classification, $L(\tilde{y}, y) = \mathbf{1}[\tilde{y} \neq y]$
- For regression, eg, $L(\tilde{y}, y) = (\tilde{y} - y)^2$

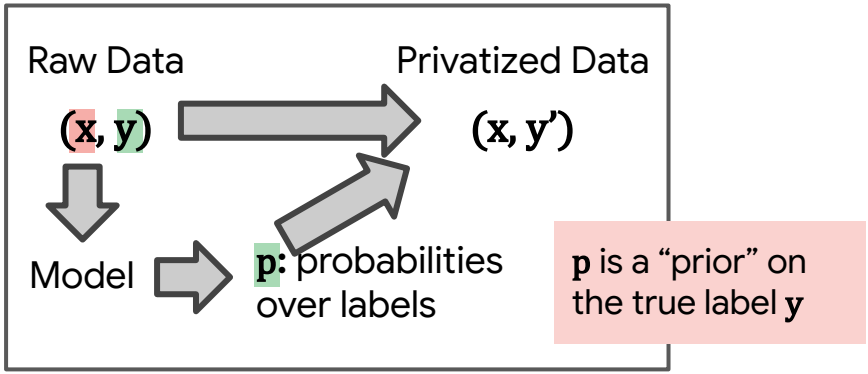
RR-with-Prior (Classification) *[Ghazi et al'21]*

Randomized Response (RR) [*Warner'65*]



$$\Pr[y' = a] = \begin{cases} e^{\varepsilon} / (e^{\varepsilon} + K - 1) & \text{if } a = y \\ 1 / (e^{\varepsilon} + K - 1) & \text{if } a \neq y \end{cases}$$

Labels $\in \{1, \ldots, K\}$



- Run RR only on the top **k** labels (eg, **k ≤ 3**)
- Can help reduce the noise, for same **ε**
- This is optimal for the objective of maximizing **Pr[y' = y]**, assuming **p** is the "true prior" and **k** is chosen appropriately

Optimality of RR-with-Prior

Choose ε-DP mechanism M to minimize $\mathbf{E}_{y\sim p,\ \tilde{y}\sim M(y)}\ \mathbf{1}[\tilde{y} \neq y]$

M can be written as a K x K matrix
$M(\tilde{y}, y) = \Pr[M \text{ outputs } \tilde{y} \text{ on input } y]$

Study the structure of the solutions to this LP to conclude optimality

maximize $\Sigma_{y\in[K]}\ p_y \cdot M(y, y)$ subject to

(i) $\Sigma_{\tilde{y}\in[K]}\ M(\tilde{y}, y) = 1,\ \forall y \in [K]$

Probability distribution

(ii) $M(\tilde{y}, y) \geq 0$

(iii) $M(\tilde{y}, y) \leq e^{\varepsilon} \cdot M(\tilde{y}, y'),\ \forall \tilde{y}, y, y' \in [K]$

DP condition

RR-on-Bins (Regression) *[Ghazi et al'23]*

Partition the range of possible y values into intervals ("bins"), and assign a representative value to each bin

- M(y) := ε-RR over Φ(y), where Φ(y) = representative value for bin containing y
- M computed efficiently using dynamic programming in time $O(KD^2)$, where
  - D = #possible y values (discretize if values are continuous)
  - K = upper bound on number of bins (≤ D)
- M(·) on distribution p is optimal for minimizing $\mathbf{E}_{y\sim p,\, \tilde{y}\sim M(y)}\, L(\tilde{y}, y)$
  - Characterize the optimal solutions to an underlying LP
- Led to flexible reports on Privacy Sandbox Chrome/Android APIs

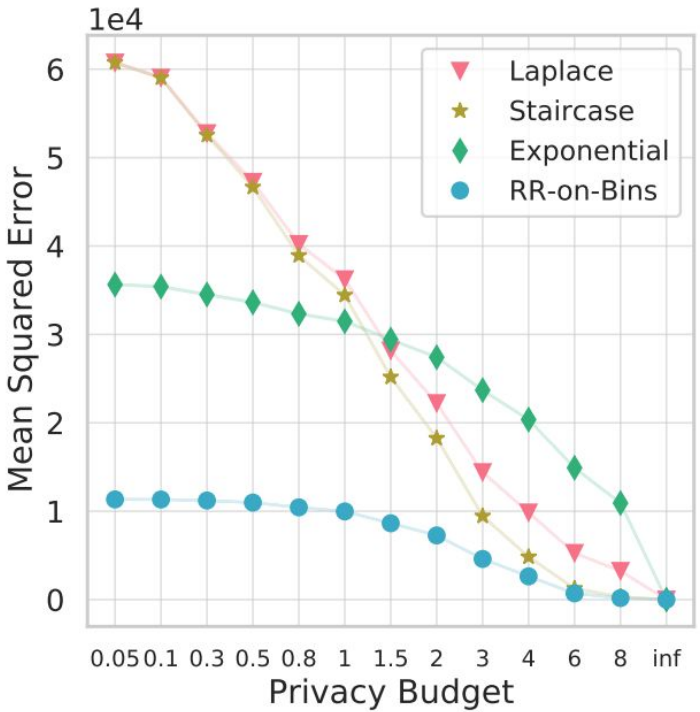
Criteo Conversion Log Dataset

- Criteo Sponsored Search Conversion Log Dataset:

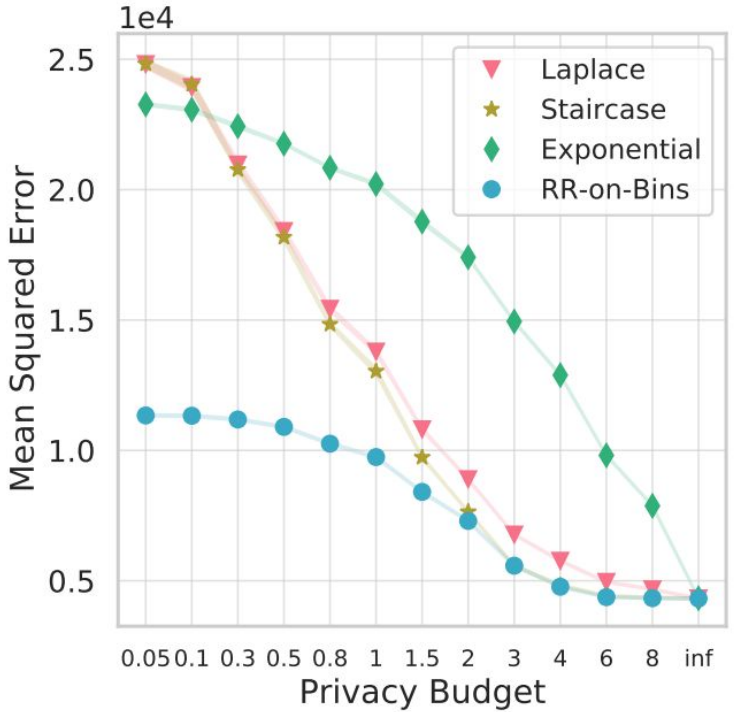
  90 days of Criteo live traffic data, with ~15M examples.

  ailab.criteo.com/criteo-sponsored-search-conversion-log-dataset/
- Goal: Predict conversion value (in €) (clipped to €400 for simplicity)
- Regression task / dataset

Results on Criteo Conversion Log Dataset



(a) Mechanism



(b) Generalization

Predict conversion value

Bias-Variance Trade-offs

Challenge: Sometimes RR-on-Bins do not give best results after training

- E.g. sometimes less bias ⇒ better accuracy *[Badanidiyuru et al. ’23]*
- Problem can still be formulated as a linear program
- Isn’t known to have an explicit solution

Unbiased Mechanism

Regression *[Ghazi et al. '23]*

$$\min_{M} \mathop{\mathbb{E}}_{\substack{y\sim P \\ y'\sim M(y)}} \left[(y'-y)^2\right]$$

subject to: M is ε-DP.

Linear program
(for fixed inputs Y, outputs Y')

Classification *[Ghazi et al. '21]*

$$\min_{M} \mathop{\Pr}_{\substack{y\sim P \\ y'\sim M(y)}} \left[y' \neq y\right]$$

subject to: M is ε-DP.

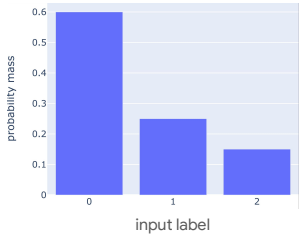
Regression *[Badanidiyuru et al.'23]*

$$\min_{M} \mathop{\mathbb{E}}_{\substack{y\sim P \\ y'\sim M(y)}} \left[(y'-y)^2\right]$$

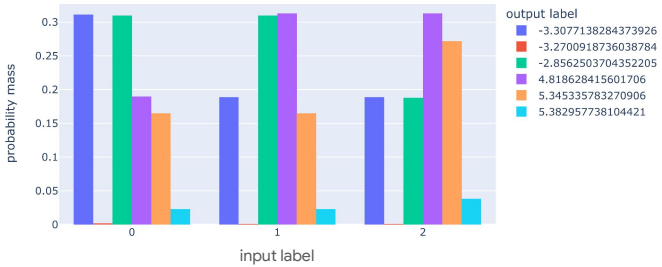
subject to: M is ε-DP

and $\forall y : \mathop{\mathbb{E}}_{y'\sim M(y)} [y'] = y$

Prior P



Optimal Unbiased Mechanism M for ε = 0.5



Motivation for Unbiased Noisy Labels

Regression *[Ghazi et al. '23]*

$$\min_{M} \mathop{\mathbb{E}}_{\substack{y\sim P \\ y'\sim M(y)}} \left[(y'-y)^2\right]$$

subject to: M is ε-DP.

Regression *[Badanidiyuru et al.'23]*

$$\min_{M} \mathop{\mathbb{E}}_{\substack{y\sim P \\ y'\sim M(y)}} \left[(y'-y)^2\right]$$

subject to: M is ε-DP

and $\forall y : \mathop{\mathbb{E}}_{y'\sim M(y)} [y'] = y$

Minimizing variance, while having zero bias.

- Zero bias preserves the *Bayes Optimal Predictor*

Theorem. The following are equivalent:

- **∀ D :** Predictor minimizing loss w.r.t. **noisy labels** = Predictor minimizing loss w.r.t. **true labels**
- Mechanism is unbiased, that is, $\forall y : \mathop{\mathbb{E}}_{y'\sim M(y)} [y'] = y$

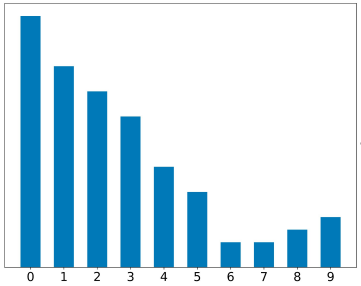
- Zero bias provides unbiased stochastic gradients

$$\mathop{\mathbb{E}}_{y'\sim M(y)} \nabla_\theta \ell(f_\theta(x), y') = \nabla_\theta \ell(f_\theta(x), y)$$

Since gradient is affine in the label: $\nabla_\theta \ell(f_\theta(x), y) = f_\theta(x) \cdot \nabla_\theta f_\theta(x) - y \cdot \nabla_\theta f_\theta(x)$

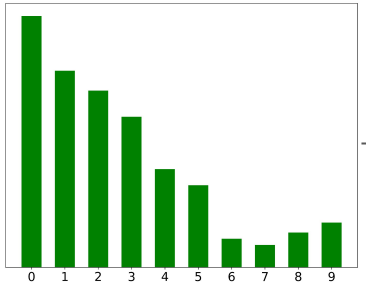
Final mechanism: Using privately estimated prior

Prior
(normalized histogram of labels in training set)



ε_{prior}-DP

Private Prior
(adding Laplace noise to histogram)



$$\min_{M} \mathop{\mathbb{E}}_{\substack{y\sim P \\ y'\sim M(y)}} [(y'-y)^2]$$

subject to: M is ε_{label}-DP

and $\forall y : \mathop{\mathbb{E}}_{y'\sim M(y)} [y'] = y$

M
(Optimal Unbiased Randomizer)

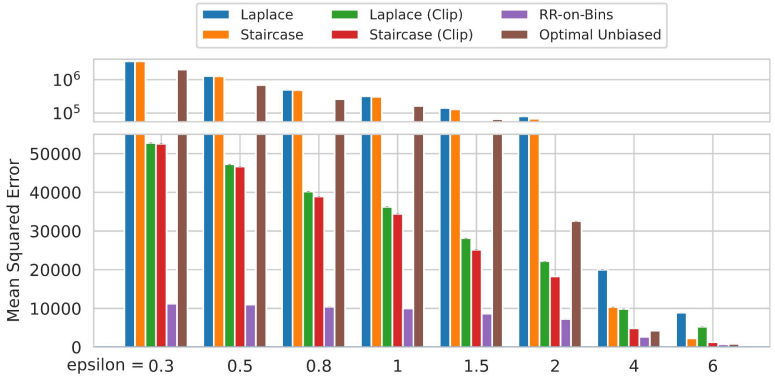
Privacy budget split: $\varepsilon = \varepsilon_{prior} + \varepsilon_{label}$

Choose output set Y' as fine enough grid with heuristic endpoints

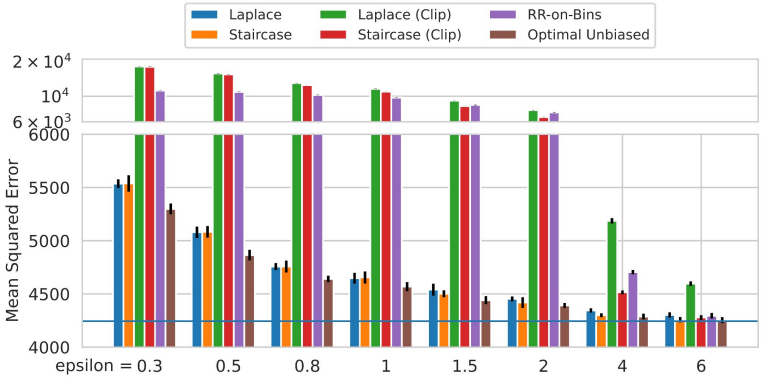
Apply ε_{label}-DP randomizer M to every label in training set.

Evaluation on Criteo Conversion Log Dataset

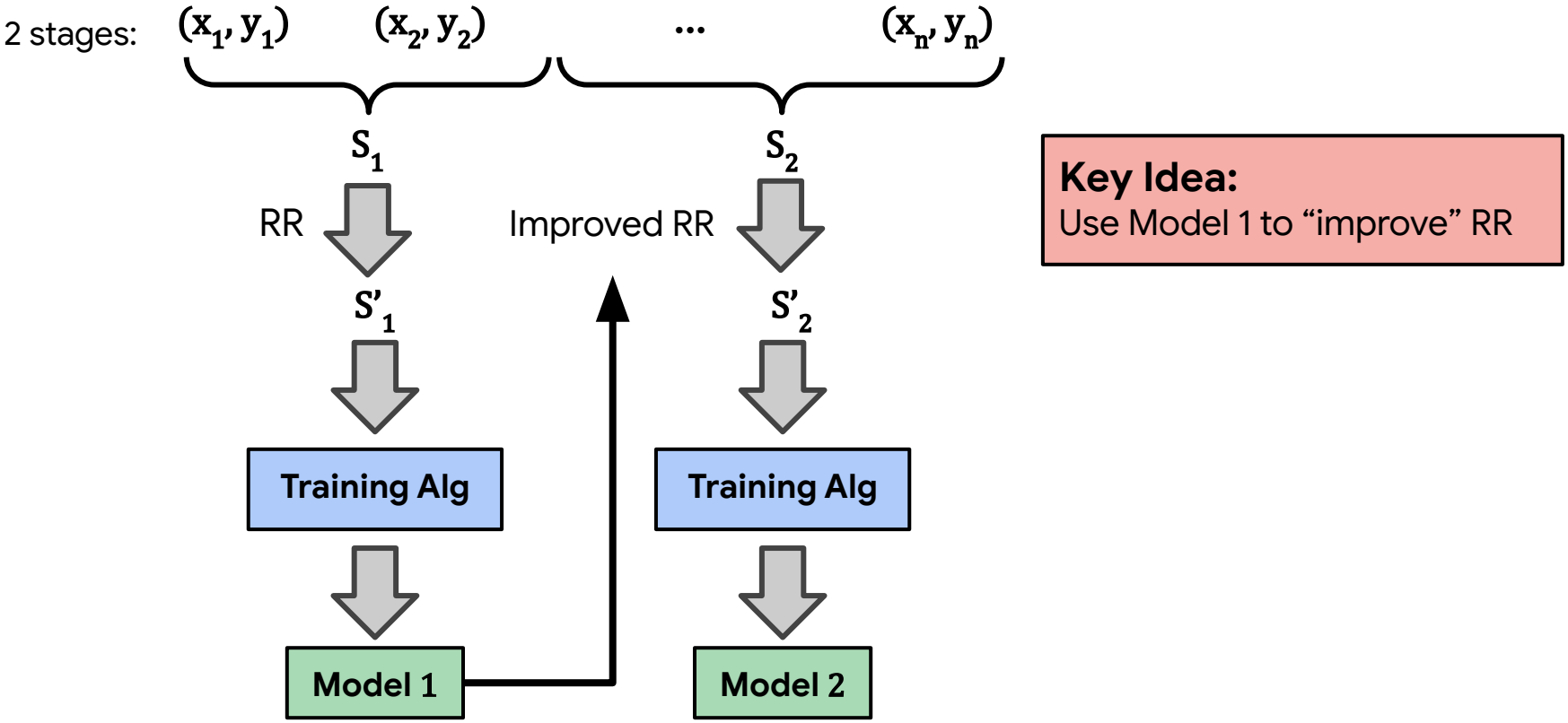
Noisy label loss on train data: $\frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} (y_i - y_i')^2$



Prediction loss on test data: $\frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (f_w(x_i) - y_i)^2$

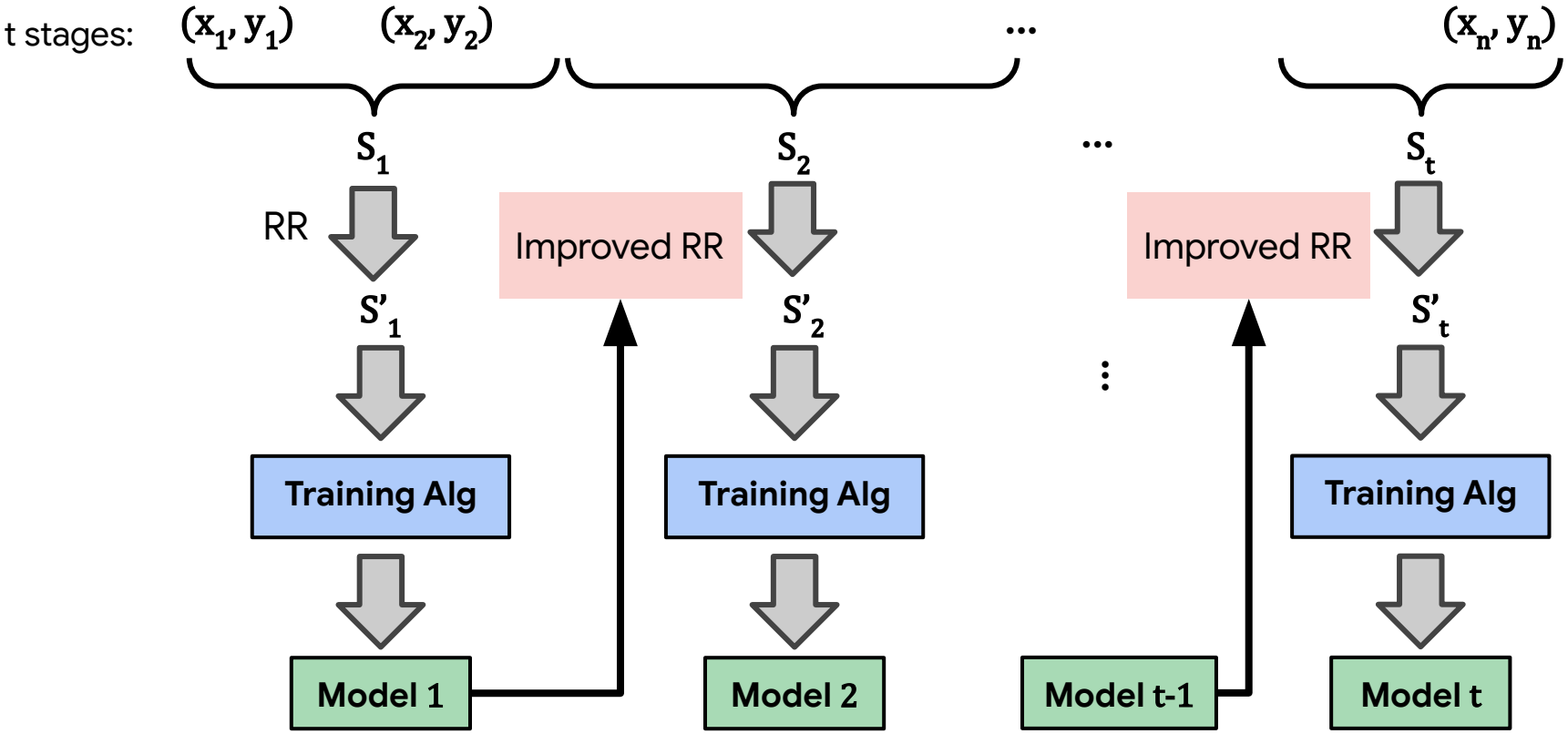


Multi-Stage Training

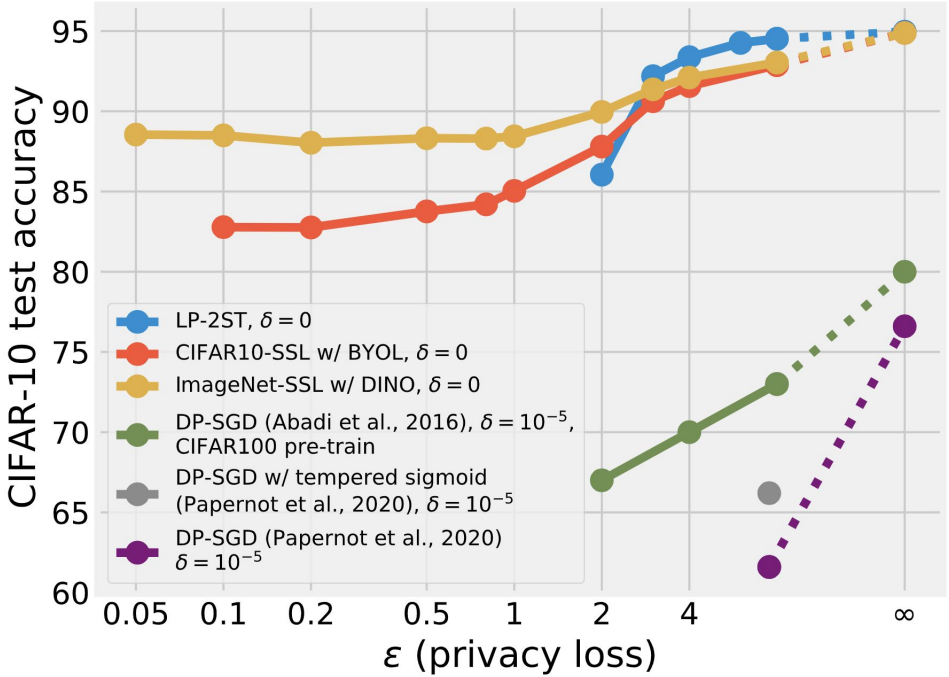


Key Idea:
Use Model 1 to "improve" RR

Multi-Stage Training



Results on CIFAR-10 (Classification)



Constructing priors

- Self-supervised training (SSL): BYOL, DINO
- Clustering and DP histograms
- Multi-stage training with data splitting
- Multi-stage training with privacy budget splitting

Significantly narrowed the performance gap between private and non-private models

Logistic Regression

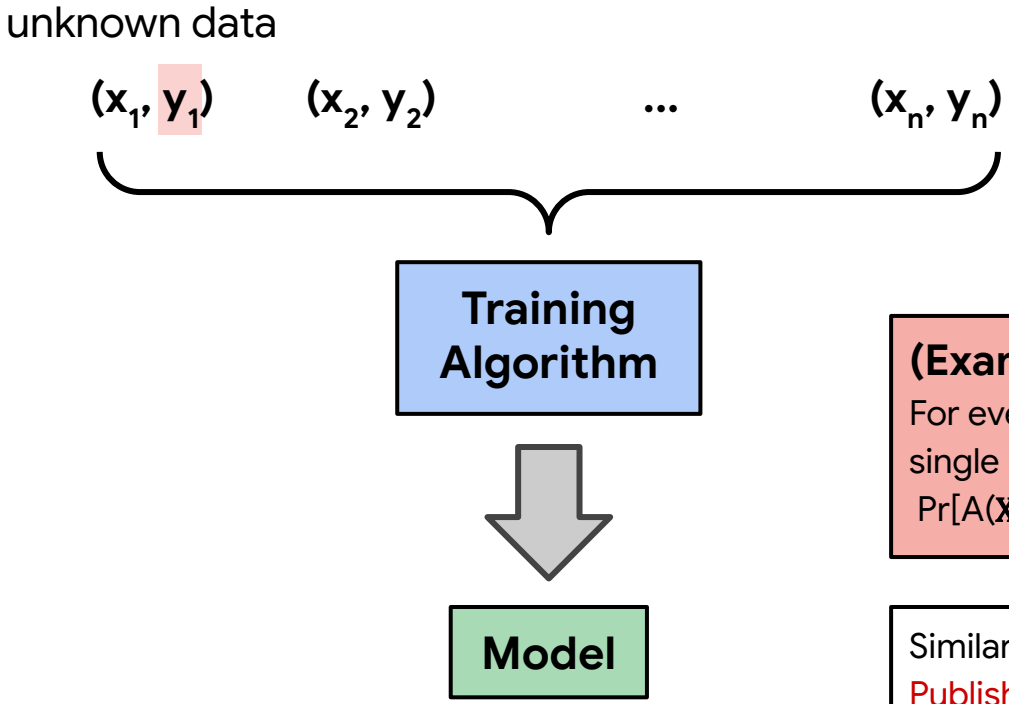
- For logistic loss, gradient depends on labels Y only through additive term proportional to $X^T Y$.
- For given features X, quantity $X^T Y$ is a linear function of labels Y that can be estimated using Discrete Laplace Mechanism (e.g., via summary reports in Privacy Sandbox ARA)
  - Would satisfy label DP in central model
- Similar idea used in Meta's winning solution to the Criteo AdKDD 2021 competition https://medium.com/criteo-engineering/results-from-the-criteo-adkdd-2021-challenge-50abc9fa3a6

Label DP Logistic Regression

Model type	Privacy ε	Optimizer, learning rate, batch size	AUC
Private logistic regression with noise	3	yogi, 0.01, 1024	0.74277
Logistic regression without noise	NA	yogi, 0.01, 1024	0.77057
Logistic baseline	NA	yogi, 0.01, 1024	0.7646
RR + logistic baseline	3	yogi, 0.01, 1024	0.75745
RR + MLP	3	yogi, 0.01, 1024	0.80212
MLP	NA	yogi, 0.01, 1024	0.805

- Results on Criteo pCTR dataset
- For same value of ε, randomized response substantially outperforms label DP logistic regression

Training with (User) Label Differential Privacy



(Example-level) ε-Label-DP

For every datasets $\mathbf{X} \simeq^s \mathbf{X}'$ that differ by a single input **label** and every output model o,

$\Pr[A(\mathbf{X}) = o] \leq e^{\varepsilon} \cdot \Pr[A(\mathbf{X}') = o]$

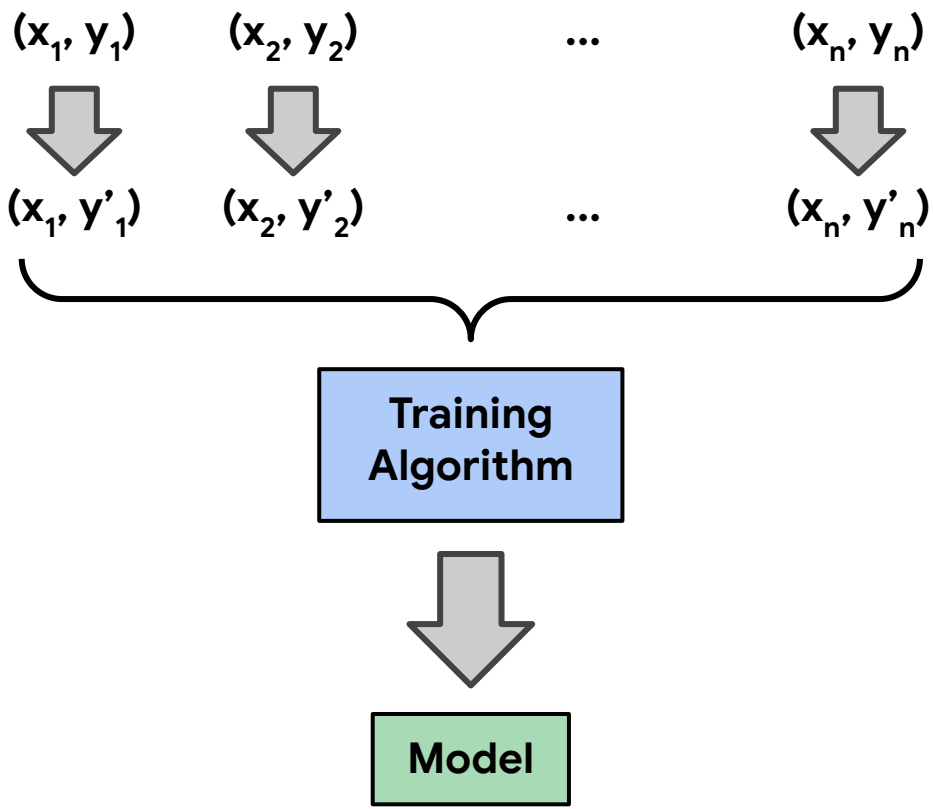
Similarly, for Impression x Time, User x Publisher x Time, User x Advertiser x Time, and User x Time privacy units

Binary Randomized Response

RR_p *[Warner'65]*

Given y_i in {0, 1}:
$y'_i = y_i$ with probability **1 - p**
Otherwise, y'_i is a random label in {0, 1}.

RR_p is ε-DP for $p = 2/(e^{\varepsilon}+1)$.



If y'_i is obtained by applying RR_p independently to y_i, then output model is ε-Label-DP.

Handling Multiple Impressions Per Privacy Unit

Example: Consider User x Time privacy unit.

Cap # of impressions per user and time period to K (keeping K random impressions, or K first impressions). Then, we have multiple options including:

1. For each user, set the privacy budget per impression to ε/K.
2. For each user i with $K_i \leq K$ impressions, set the privacy budget per impression to ε/K_i.

Both options satisfy ε-Label-DP for User x Time privacy unit.

Similar options hold for User x Advertiser x Time & User x Publisher x Time privacy units.

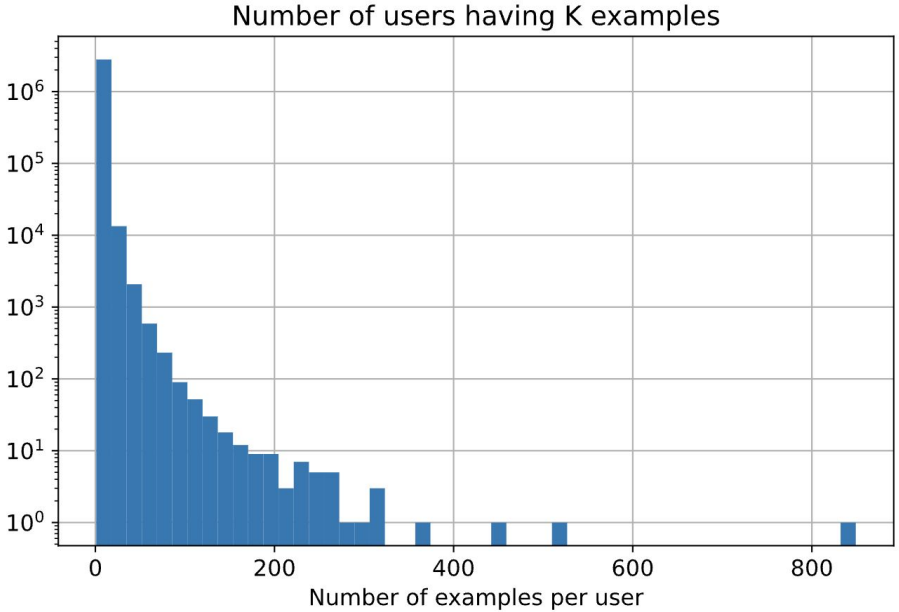
For Impression x Time privacy unit, no capping is needed. RR is applied privacy budget ε.

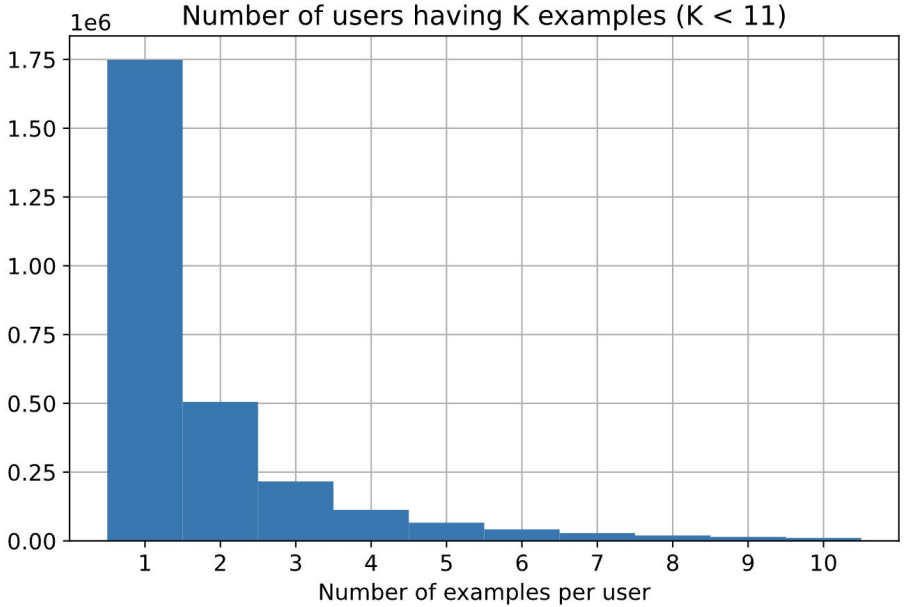
Criteo Attribution Modeling for Bidding Dataset

https://ailab.criteo.com/criteo-attribution-modeling-bidding-dataset/

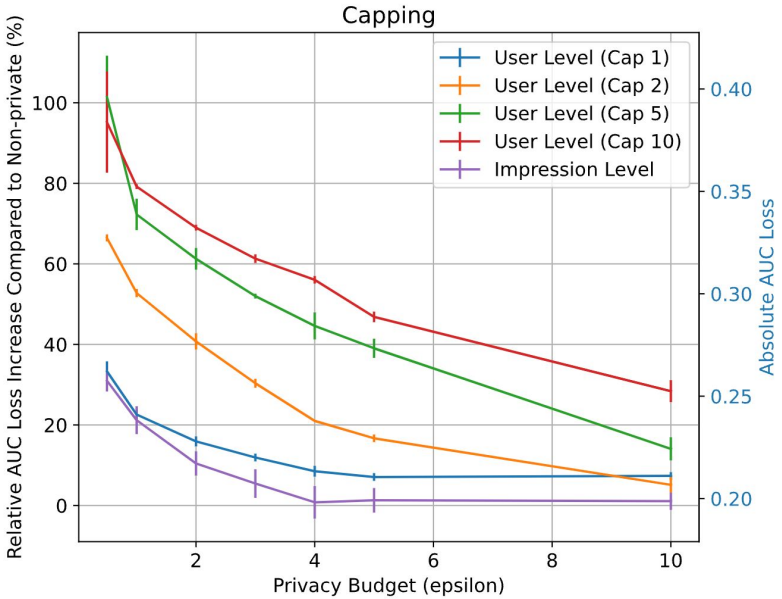
- Sample of 30 days of Criteo live traffic data.
- Each example corresponds to a click and contains:
  - **Features:** campaign ID, 9 contextual features, and the cost paid for the display.
  - **Label:** a 0/1 field indicating whether there was a conversion in the 30 days after the click and that is last-touch attributed to this click.
  - **User ID:** can be used to evaluate User x Time privacy unit.
- Number of rows is 5,947,563. Conversion rate (under last-touch attribution) is 6.74%.
- Feed-forward neural network
  - Embedding dimension is 8
  - Hidden dimensions are 128 x 64.
- Tune hyperparameters with optimizer in {rmsprop, adam, sgd}, learning rate in {0.0005, 0.0008, 0.001, 0.002, 0.005}, and training epochs in {100, 200}.

Criteo Attribution Modeling for Bidding – Statistics





Criteo Attribution Modeling for Bidding – Evaluation Results



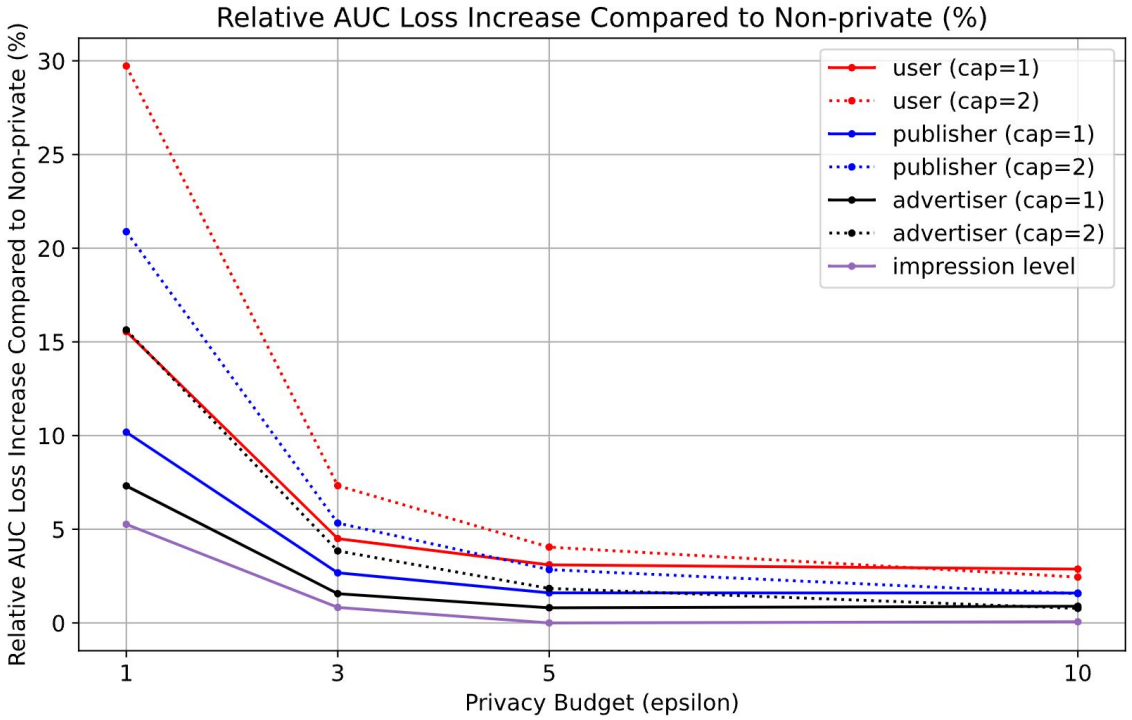
Notes

- For Impression x Time privacy unit and ε = 4, relative AUC loss is 0.79%.
- For User x Time privacy unit with ε = 4, smallest relative AUC loss is 8.51%.
- For User x Time privacy unit, smaller loss is achieved by increasing caps as we increase ε.

Proprietary Ads Dataset

- App install ads
- Contains data from multiple advertisers and publishers.
  - Can be used to evaluate User x Time, User x Publisher x Time, and User x Advertiser x Time privacy units
- **Features:** use categorical features, and pass the concatenation of their embeddings through multiple layers of a fully connected feedforward neural network.
- **Labels:** 0/1 corresponding to installs (= conversions)

Proprietary Ads Dataset – Evaluation Results



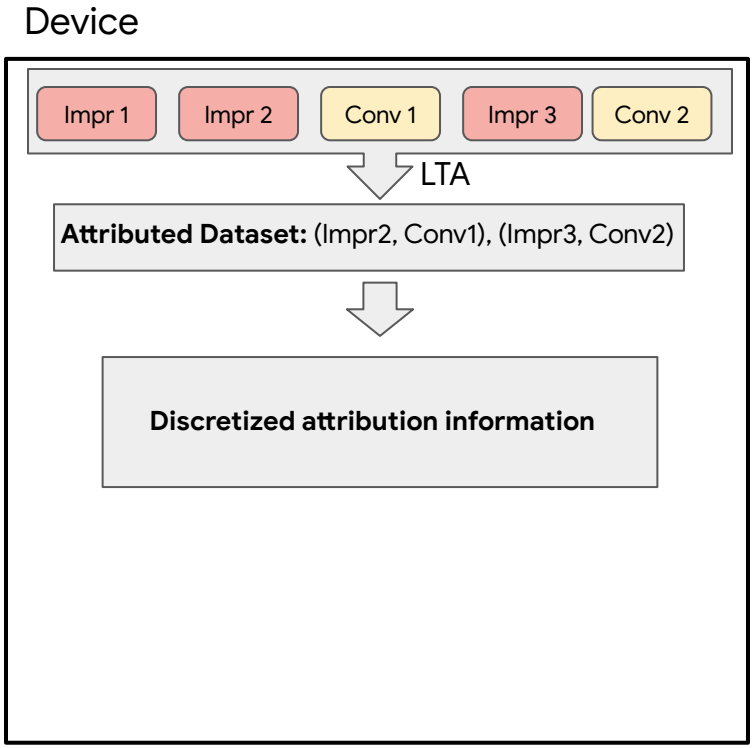
Notes

- For Impression x Time privacy unit and ε = 3, relative AUC loss is 0.83%.
- For User x Time privacy unit and ε = 3, smallest relative AUC loss is 4.50%.
- For User x Publisher x Time privacy unit and ε = 3, smallest Relative AUC loss is 2.67%.
- For User x Advertiser x Time unit with ε = 3, best Relative AUC loss is 1.56%.
- In this experiment, for same ε, loss for Impression x Time < loss for User x Advertiser x Time < loss for User x Publisher x Time < loss for User x Time.

Evaluation limitations and Future Directions

- Utility might be improvable for binary conversion models
  - Debiased loss functions
- Evaluation focused on binary conversion models
  - Label DP algorithms (and RR in particular) can be extended to non-binary predictions (such as predicting number of conversions, and conversion value etc.)
- Evaluation was done offline and assumed one reporting window
  - Online performance can be impacted by delays, which might require multiple reporting windows (and more noise for the same privacy)

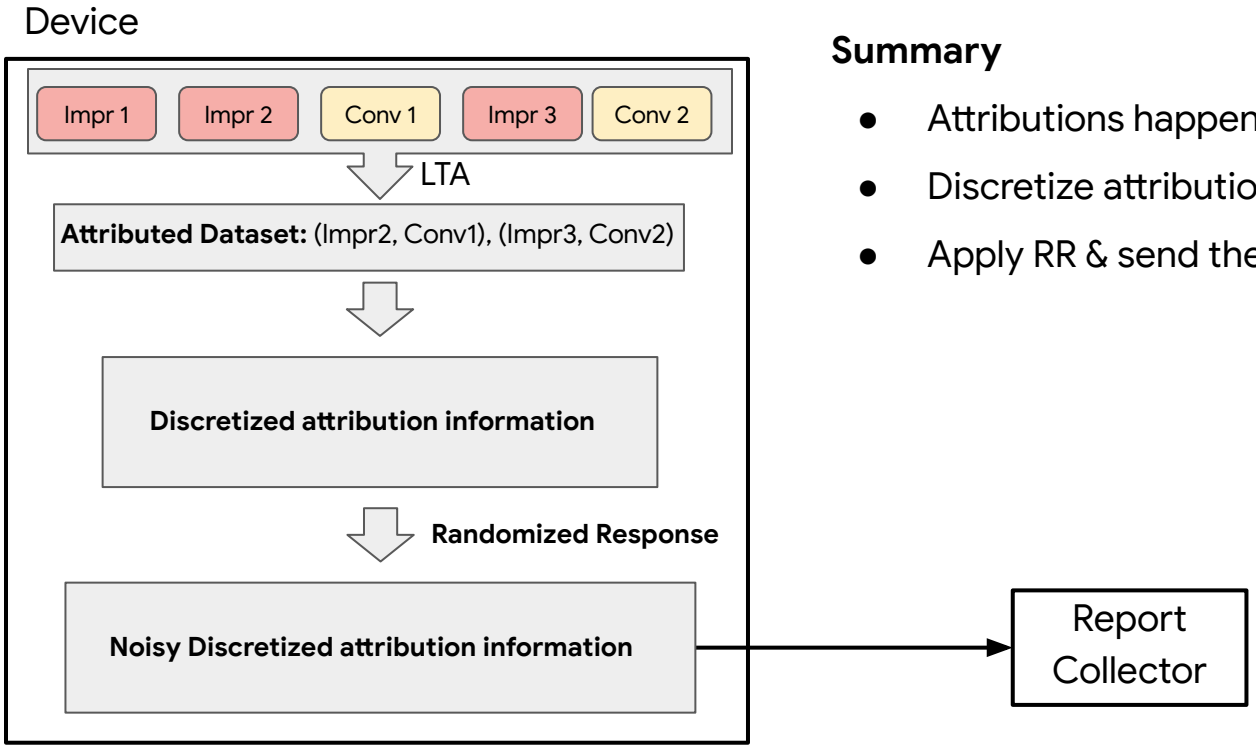
High-Level Overview of APIs: Event



Summary

- Attributions happen on device / browser
- Discretize attribution information to a finite space

High-Level Overview of APIs: Event



Summary

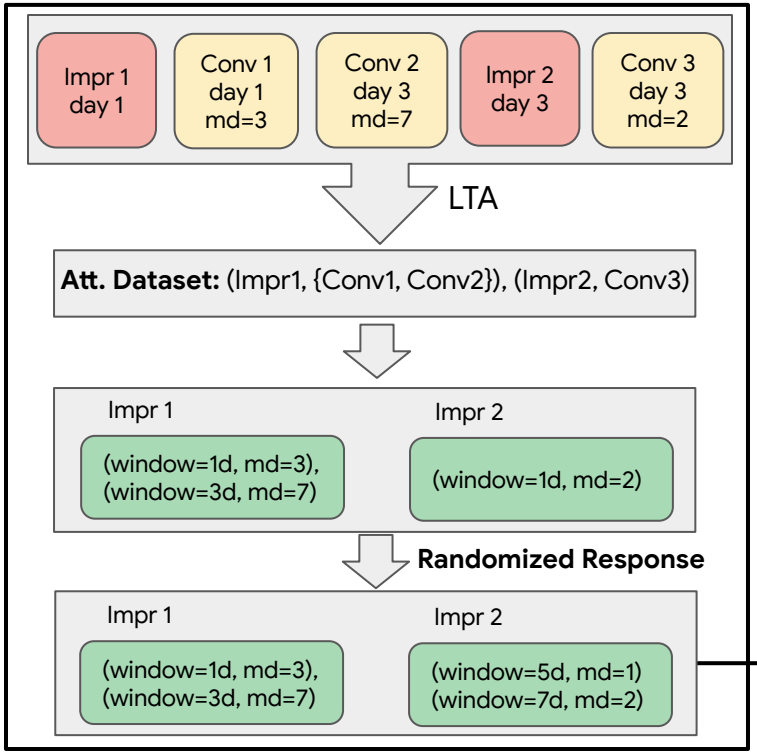
- Attributions happen on device / browser
- Discretize attribution information to a finite space
- Apply RR & send the output to report collector

ARA Event-Level Reports

Device

Reporting windows = 1d, 3d, 7d

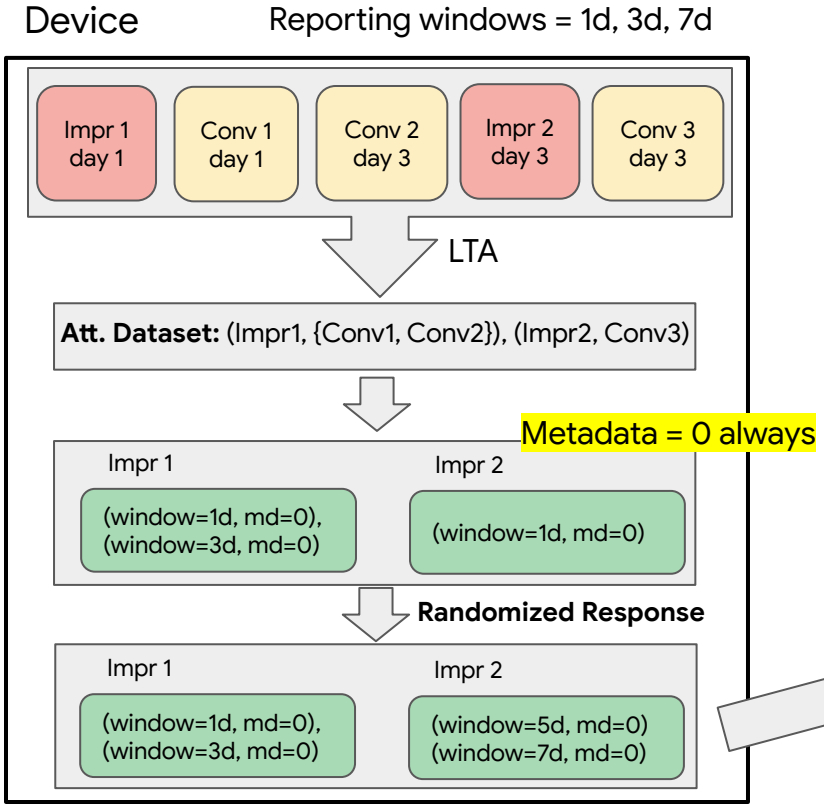
Total # of states = $\binom{w \cdot 2^b + C}{C}$



- RR at impression level
- There are w (≤ 5) “reporting (time) windows”
  - E.g. reporting windows = 1d, 3d, 7d means that reports will be sent 1d, 3d, 7d after the impression.
- Up to C (≤ 20) conversions per impression:
  - For each conversion, up to b (≤ 3) bits of “metadata” information

Report Collector

Example Usage of Event-Level Reports



Example Goal

- Train model to predict
  - p(conversion within 1d | impression)
  - p(conversion within 3d | impression)
  - p(conversion within 7d | impression)

Training Data for 1d

(Impr 1, y_1)

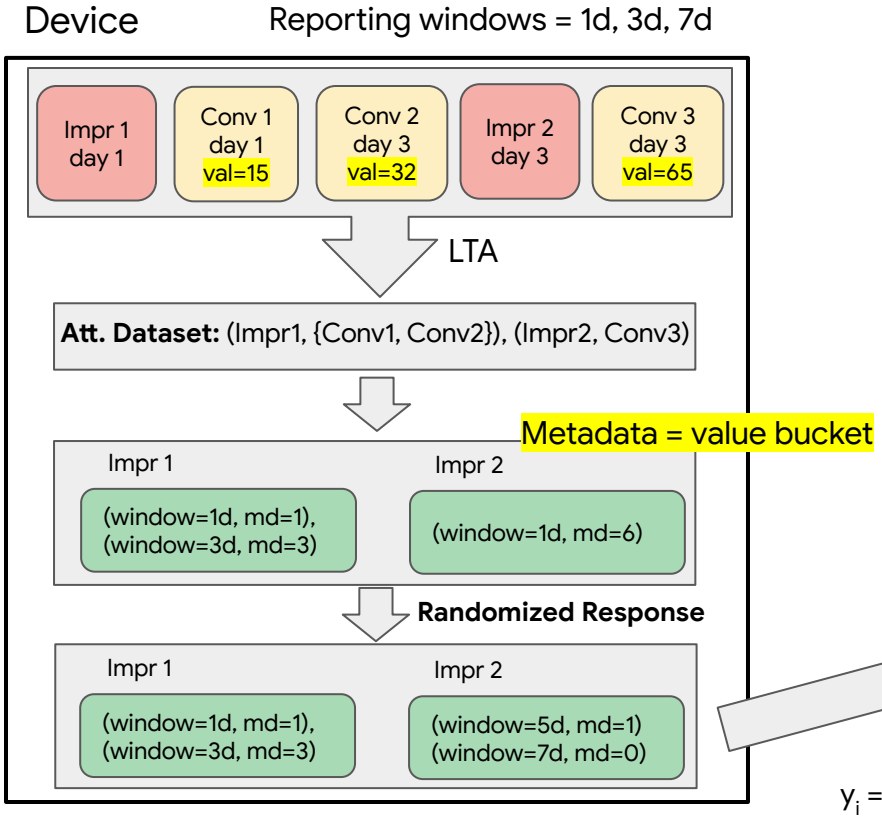
(Impr 2, y_2)

⋮

Training → ML model

y_i = noisy label indicating whether impr i has converted within 1d

Example Usage of Event-Level Reports



Example Goal

- Train model to predict
  - E(conversion value within 1d | impression)
  - E(conversion value within 3d | impression)
  - E(conversion value within 7d | impression)

Example value buckets: [0, 10), [10, 20), ..., [70, 80)

Training Data for 1d

(Impr 1, y_1)

(Impr 2, y_2)

⋮

Training → ML model

y_i = noisy conversion value attributed to impr i within 1d

Event Reports: Utility Optimization & Challenges

Discretizing the values

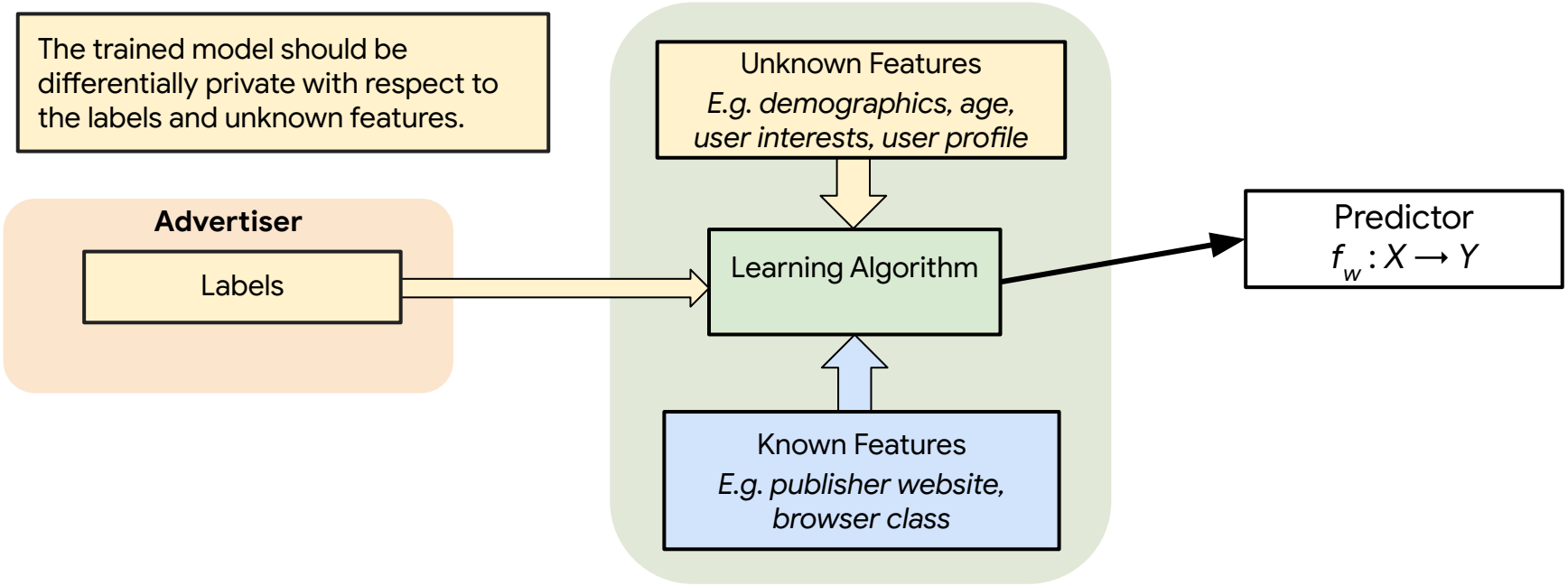
- Shown in *[Ghazi et al.'23]* that "RR-on-bins" are optimal for certain loss between randomized labels and true labels: Led to Flexible Reports on Chrome/Android
- Optimizing for the "bins" can be done efficiently using historical data (or private aggregate queries)
- Training methods that are robust to noisy labels: E.g. regularization, debiasing, self-supervised learning
- **Challenge:** Sometimes these do not give best results after training: E.g. sometimes less bias ⇒ better accuracy *[Badanidiyuru et al. 23]*

Extending APIs to support more complex algorithms:

- Other *Label DP* algorithms
  - RRWithPrior *[Ghazi et al.'21]*
  - PATE-FM *[Malek et al. '21]*
  - ALIBI *[Malek et al. '21]*
  - Clustering *[Esfandiari et al. '22]*
- **Research question:** other better *Label DP* algorithms?

DP with Partially Known Features

The trained model should be differentially private with respect to the labels and unknown features.



Privacy Sandbox Protected Audience API on Chrome and Android

- Some features depend on cross-site information and use 3p cookies, some do not
- E.g. remarketing ads use case

DP Training with Partially Known Features

Hybrid Algorithm: Label-DP phase followed by DP-SGD phase *[Chua et al. '24]*

Label-DP Phase: Train truncated model with randomized response labels and unknown embeddings set to 0, with $(\varepsilon_1, 0)$-DP

DP-SGD Phase: Train entire model with (ε_2, δ)-DP

Total privacy budget (ε, δ) is split between two phases as:

$\varepsilon_1 := \min\{0.6\,\varepsilon, 3\}$ $\qquad\qquad$ $\varepsilon_2 := \varepsilon - \varepsilon_1$

Two baselines:

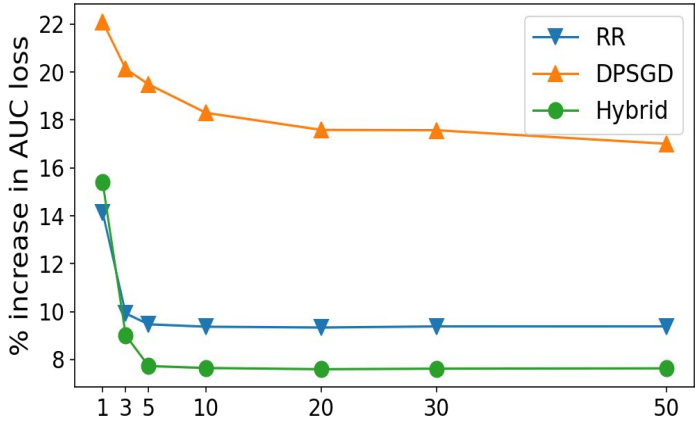
1) **RR**: labels privatized with noise, unknown features discarded $(\varepsilon_1 = \varepsilon, \varepsilon_2 = 0)$
2) **DP-SGD**: all features treated as unknown $(\varepsilon_1 = 0, \varepsilon_2 = \varepsilon)$

We train binary classification models with binary cross entropy loss and report the test AUC loss % relative to the non-private baseline.

Hybrid DP Algorithm Evaluation

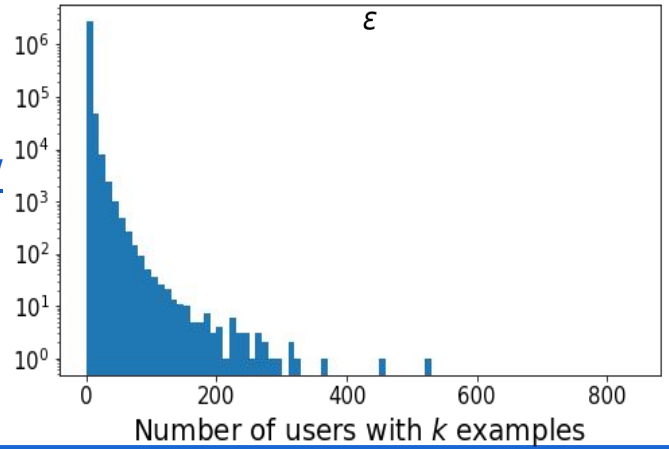
Evaluation 1: Criteo Display Ads pCTR Dataset

- 40M examples over 7 days of Criteo traffic
  kaggle.com/c/criteo-display-ad-challenge/overview
- Treat even-numbered features as unknown and odd-numbered features as known
- Limit # of DP-SGD epochs to ≤3 because of high cost
- Goal: Predict probability of click



Evaluation 2: Criteo Attribution Modeling for Bidding Dataset

- 16M impressions from 30 days of Criteo traffic
  ailab.criteo.com/criteo-attribution-modeling-bidding-dataset/
- Treat cat1, cat2 as unknown, cat[3-9] and campaign as known.
- Goal: Predict probability of conversion
- Evaluate both example-level DP and user-level DP

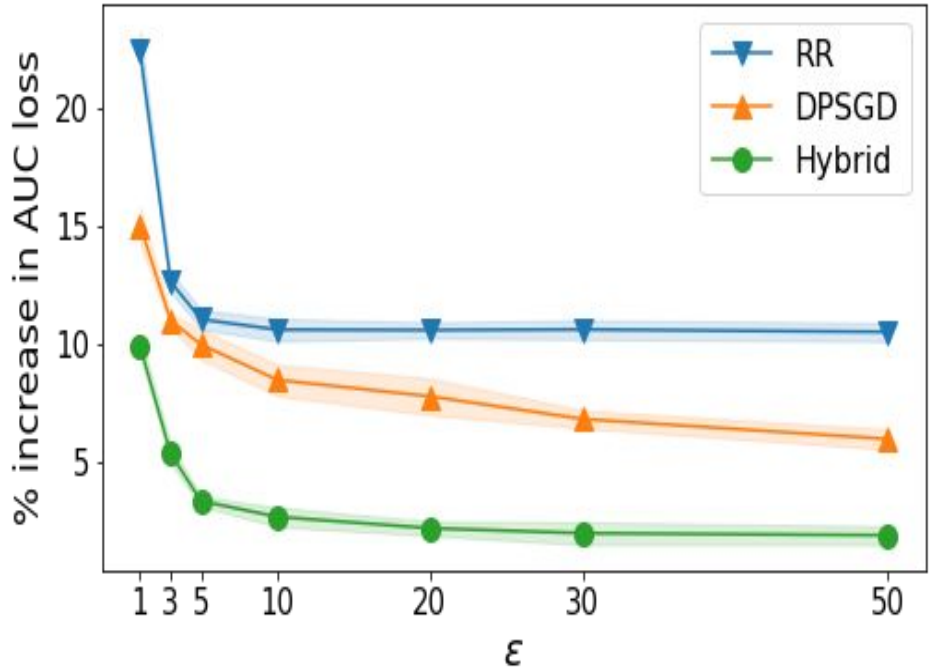


User level DP with Partially Known Features

Translate example-level DP to user-level DP via group privacy:

If a mechanism M satisfies (ε, δ)-DP, then the mechanism $M^{(k)}$ obtained by retaining ≤ k examples per user satisfies $(k\varepsilon, \delta(e^{k\varepsilon} - 1)/(e^{\varepsilon} - 1))$-user-level-DP

Hybrid algorithm improves over RR and DP-SGD over a wide range of privacy budgets, especially when $\varepsilon \geq 3$.



DP with a Known Set of Feature Values

Alternative notion applicable in some ad modeling settings

- Only set of values for a feature is known (but not the specific value on a per-example basis)
- E.g., the set of all ads are known to the ad-tech but not the specific feature associated with each example
- Provides stronger privacy guarantee than DP with partially known features, but by definition cannot provide higher utility
- Recent algorithm in this setting *[Krichene et al. ‘23]*

Some Practical Insights

- DP-SGD is feasible even for tasks with very sparse gradients, enabled by
  - Advances in compilation methods
  - Larger batch sizes
  - Tighter privacy accounting, eg, Privacy Loss Distributions (PLD)
  - Tailored algorithms
- Label DP often allows superior utility
- Tailoring privacy model to application provides further improvements
- Prior information on labels can be
  - Leveraged to improve utility
  - Obtained from historical data, current model predictions, or clustering and histograms
- Known features could improve utility

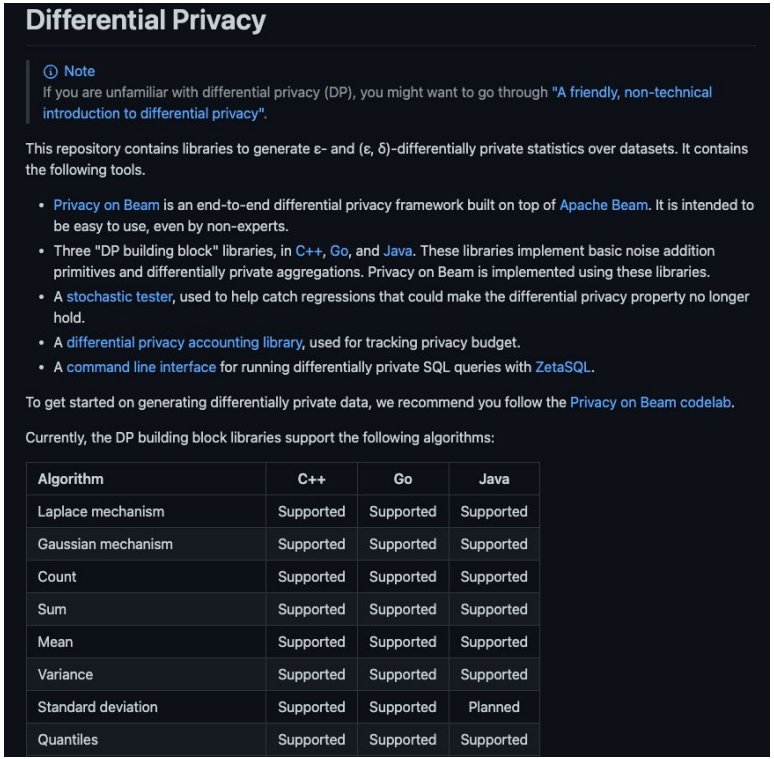
Modeling Research Questions

- Better algorithms for ads modeling with full DP, label DP, and DP with partially known features?
- Label DP
  - Full characterization of optimal unbiased randomizers?
  - Better algorithms for computing optimal unbiased randomizers?
- Algorithms for when some features are partially known while others belong to a known set of values?

Conclusion & Future Directions

Open-Source Libraries

- Some open-source DP libraries:
  - Generic DP Libraries:
    - Google DP Library
    - IBM Diffprivlib Library
    - OpenDP Library
  - DP ML Libraries:
    - Tensorflow privacy
    - Pytorch Opacus



Differential Privacy

> Note
> If you are unfamiliar with differential privacy (DP), you might want to go through "A friendly, non-technical introduction to differential privacy".

This repository contains libraries to generate ε- and (ε, δ)-differentially private statistics over datasets. It contains the following tools.

- Privacy on Beam is an end-to-end differential privacy framework built on top of Apache Beam. It is intended to be easy to use, even by non-experts.
- Three "DP building block" libraries, in C++, Go, and Java. These libraries implement basic noise addition primitives and differentially private aggregations. Privacy on Beam is implemented using these libraries.
- A stochastic tester, used to help catch regressions that could make the differential privacy property no longer hold.
- A differential privacy accounting library, used for tracking privacy budget.
- A command line interface for running differentially private SQL queries with ZetaSQL.

To get started on generating differentially private data, we recommend you follow the Privacy on Beam codelab.

Currently, the DP building block libraries support the following algorithms:

Algorithm	C++	Go	Java
Laplace mechanism	Supported	Supported	Supported
Gaussian mechanism	Supported	Supported	Supported
Count	Supported	Supported	Supported
Sum	Supported	Supported	Supported
Mean	Supported	Supported	Supported
Variance	Supported	Supported	Supported
Standard deviation	Supported	Supported	Planned
Quantiles	Supported	Supported	Supported

github.com/google/differential-privacy

Pointers

- Private Attribution Measurement APIs:
  - List of proposals (beyond attribution): see W3C github repo
  - Join WICG: https://www.w3.org/community/wicg/
  - Join meetings / discussions on github (e.g. for ARA see here)
- Project Halo
  - WFA project website
  - Open-source github repo for Cross-media measurement

Conclusion

- Formal privacy guarantees for ad analytics and modeling functionalities are possible
- Better algorithms are pushing the privacy-utility trade-off curve
- Well-defined functionalities and utility metrics are invaluable for designing privacy-preserving algorithms

Future Directions

- General-purpose DP synthetic ad data remains an important research problem
- Combining known data that the analyst might have access to with outputs of privacy-preserving algorithms and APIs is an important future direction.