ENHANCING METRIC PRIVACY WITH A SHUFFLER

Andreas Athanasiou
Catuscia Palamidessi, Kostas Chatzikokolakis
Inria
INSTITUT POLYTECHNIQUE DE PARIS

“Have you cheated on the exam?”

—A random (angry) professor

Have you cheated on the exam?

Answers:
(using Differential Privacy)

Class A

20%
Yes

Class A without Bob

~20%
Yes

Bob's answer **does not** *(really)* **change** the **result***!*

Differential Privacy (ε,δ)

	Same	Different
Dataset X:	1, 15, 2, 3 , 50 , … ,	**5**
Dataset X':	1, 15, 2, 3 , 50 , … ,	**90**

} Adjacent Datasets

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

ε = Privacy Loss

δ = (Rare) cases where **ε** doesn't hold

The probability to see the <u>same</u> outcome between every possible set of adjacent datasets:

- Is at most **e^ε**
- Can be above **e^ε** in at most **δ**% of the cases (for 0≤δ≤1)

Differential Privacy (ε,δ)

	Same	Different	
Dataset X:	1, 15, 2, 3 , 50 , … ,	**5**	Adjacent Datasets
Dataset X':	1, 15, 2, 3 , 50 , … ,	**90**	

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

ε = Privacy Loss

δ = (Rare) cases where **ε** doesn't hold

The probability to see the <u>same</u> outcome between every possible set of adjacent datasets:

- Is at most $\mathbf{e^{\varepsilon}}$
- Can be above $\mathbf{e^{\varepsilon}}$ in at most **δ**% of the cases (for 0≤δ≤1)

Example:

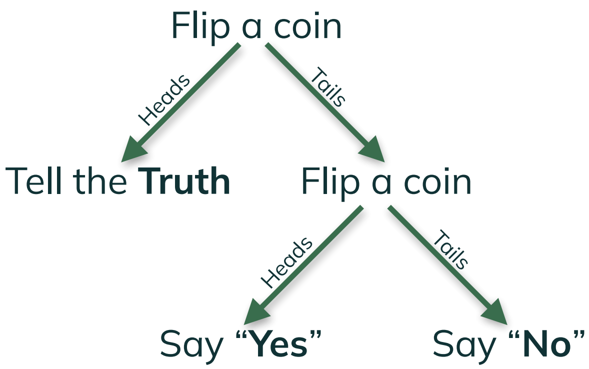
ε = 0.2 *and* **δ = 0.01:**

the mechanism offers 0.2 privacy, in 99% of cases

Randomised Response

Question: *"Have you cheated on the exam?"*

Answer:



Metric Privacy (ε,δ)

(aka d-privacy)

Same | Different **by 85**

Dataset X: 1, 15, 2, 3 , 50 , … , **5**

Dataset X': 1, 15, 2, 3 , 50 , … , **90**

Datasets *with a distance* **d = 85**

$\forall x \in X, x' \in X'$:

$$\mathbb{P}[M(x) \in S] \leq e^{\varepsilon \cdot d_x(x,x')}\, \mathbb{P}[M(x') \in S] + \delta$$

ε·d = Privacy Loss

δ = (Rare) cases where **ε·d** doesn't hold

The probability to see the same outcome between every possible set of datasets **that have a distance d:**

- Is at most $\mathbf{e^{\varepsilon \cdot d}}$
- Can be above $\mathbf{e^{\varepsilon \cdot d}}$ in at most **δ**% of the cases (for 0≤δ≤1)

Metric Privacy (ε,δ)

(aka d-privacy)

	Same	Different by 85
Dataset X:	1, 15, 2, 3 , 50 , … ,	**5**
Dataset X':	1, 15, 2, 3 , 50 , … ,	**90**

Datasets *with a distance* **d = 85**

$\forall x \in X, x' \in X'$:

$$\mathbb{P}[M(x) \in S] \leq e^{\varepsilon \cdot d_x(x,x')}\, \mathbb{P}[M(x') \in S] + \delta$$

ε·d = Privacy Loss

δ = (Rare) cases where **ε·d** doesn't hold

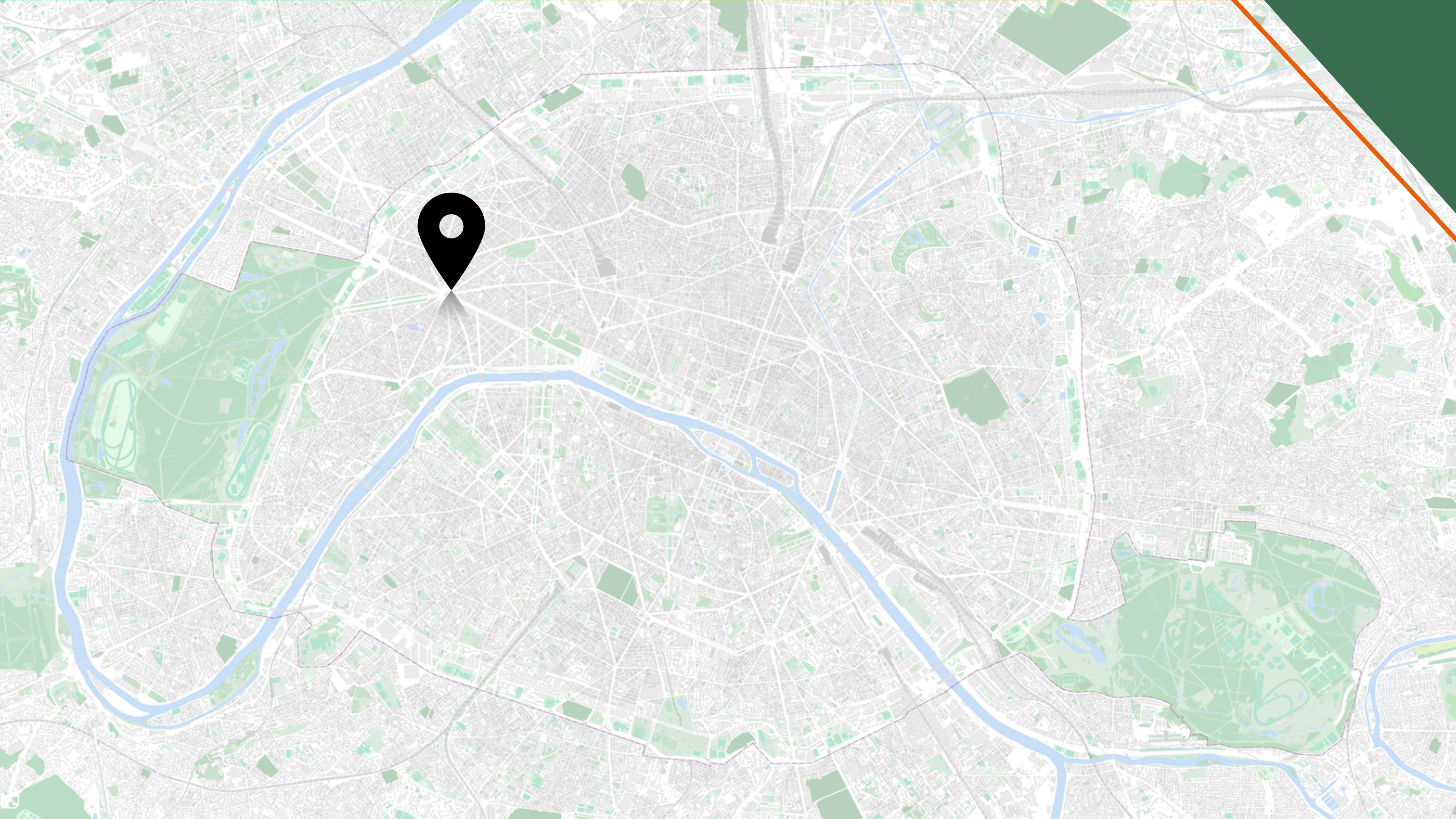
The probability to see the same outcome between every possible set of datasets **that have a distance d:**

- Is at most $\mathbf{e^{\varepsilon \cdot d}}$
- Can be above $\mathbf{e^{\varepsilon \cdot d}}$ in at most **δ**% of the cases (for 0≤δ≤1)

Example:

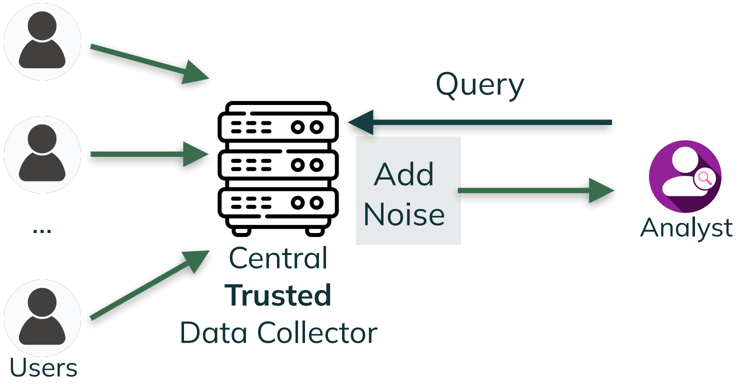
if **ε = 0.2** *and* **δ = 0.01:**

*the mechanism offers 0.2·***d** *metric privacy (for every dataset with* **a distance d***), in 99% of cases*



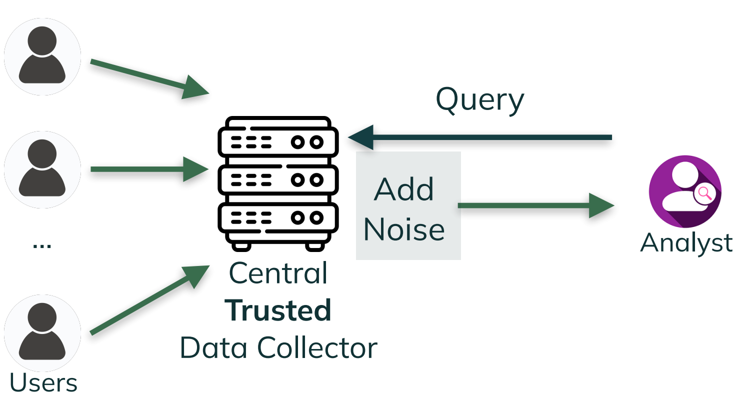

Protection
with
Metric Privacy
The **closer** two locations are, the **more indistinguishable** they should be

Models of Privacy

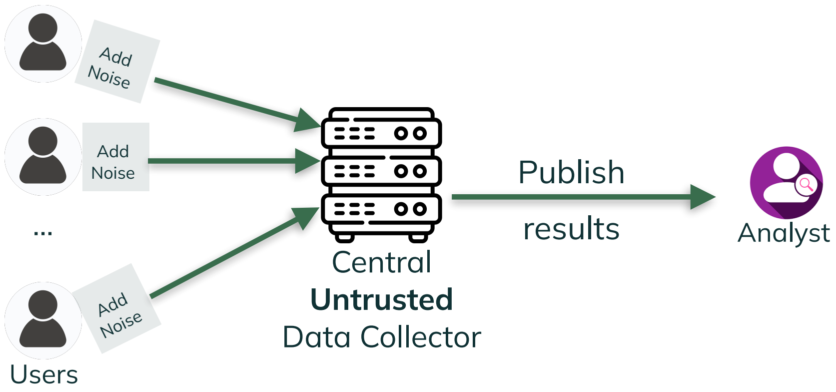


Central Model

Models of Privacy

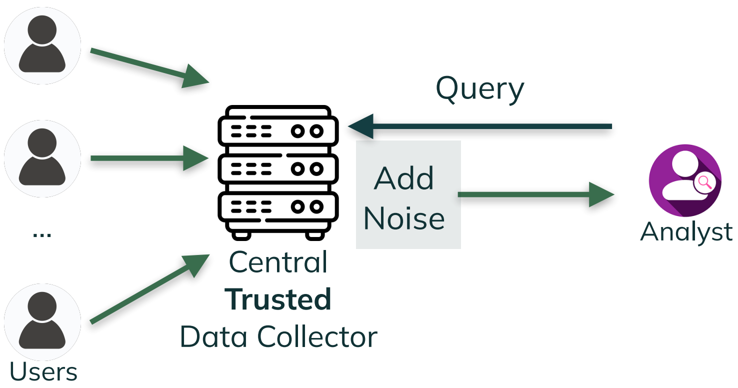


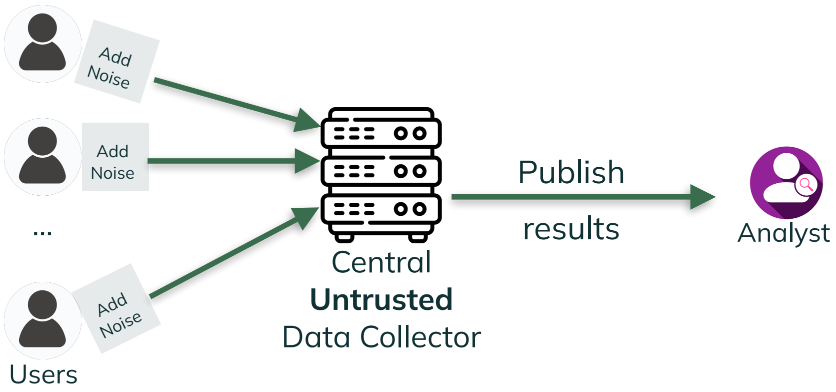
Central Model



Local Model

Models of Privacy





Central Model

+ Better utility

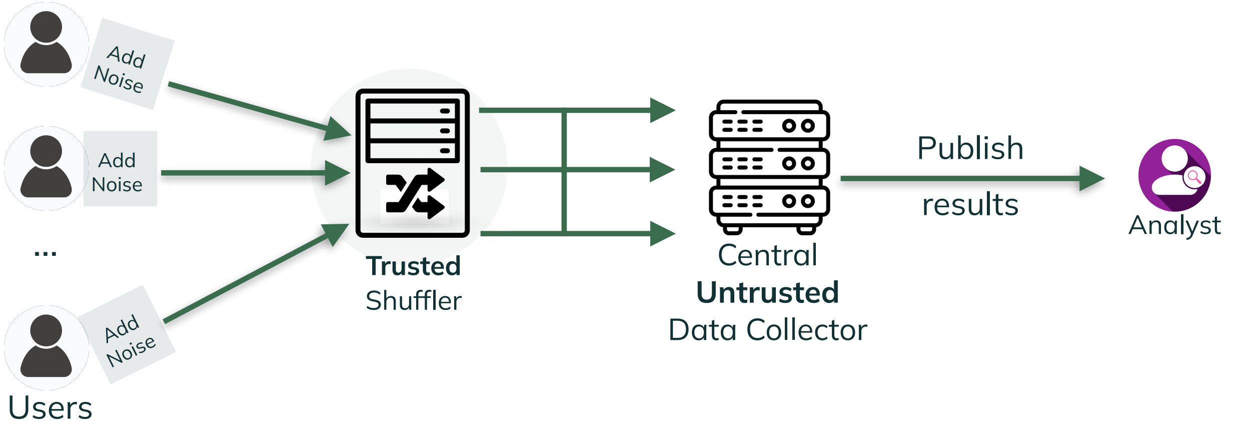
- Must trust the data collector

Local Model

+ No need to trust a central entity

- Worse utility

The Shuffle Model



Shuffle Model

+ Better utility than the Local Model
+ Less trust than the Central Model

Metric Privacy

in the

Shuffle Model

<u>Problem:</u> Private summation of integers

n users | each user **i** has a value $\mathbf{x_i} \in \{0,1, \ldots, \mathbf{k}\}$ for $\mathbf{k} \in \mathbb{N}$

Metric Privacy

in the

Shuffle Model

<u>Problem:</u> Private summation of integers

n users | each user **i** has a value $\mathbf{x_i} \in \{0,1, \dots, \mathbf{k}\}$ for $\mathbf{k} \in \mathbb{N}$

<u>Contributions:</u>

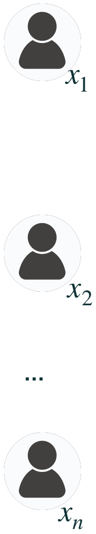
- RR - Shuffle
  Randomised Response mechanism

- geo - Shuffle
  Geometric mechanism

- $SGDL$ - Shuffle
  Symmetric Generalised Discrete Laplace distribution

RR - Shuffle

n users | each user **i** has a value $\mathbf{x_i} \in \{0,1, ..., \mathbf{k}\}$



RR - Shuffle

n users | each user **i** has a value $\mathbf{x_i} \in \{0, 1, \ldots, \mathbf{k}\}$

x_1 — Encode to b — *unary vector of size k* → $b_{1,1}, b_{1,2}, b_{1,3} \ldots b_{1,k}$

x_2 — Encode to b — *unary vector of size k* → $b_{2,1}, b_{2,2}, b_{2,3} \ldots b_{2,k}$

… … …

x_n — Encode to b — *unary vector of size k* → $b_{n,1}, b_{n,2}, b_{n,3} \ldots b_{n,k}$

$$\sum_i b_i = x_i$$

RR - Shuffle

n users | each user **i** has a value $\mathbf{x_i} \in \{0, 1, \dots, \mathbf{k}\}$

x_1 — Encode to b → unary vector of size k — $b_{1,1}, b_{1,2}, b_{1,3} \dots b_{1,k}$ → Run RR $\forall\, b_i$

x_2 — Encode to b → unary vector of size k — $b_{2,1}, b_{2,2}, b_{2,3} \dots b_{2,k}$ → Run RR $\forall\, b_i$

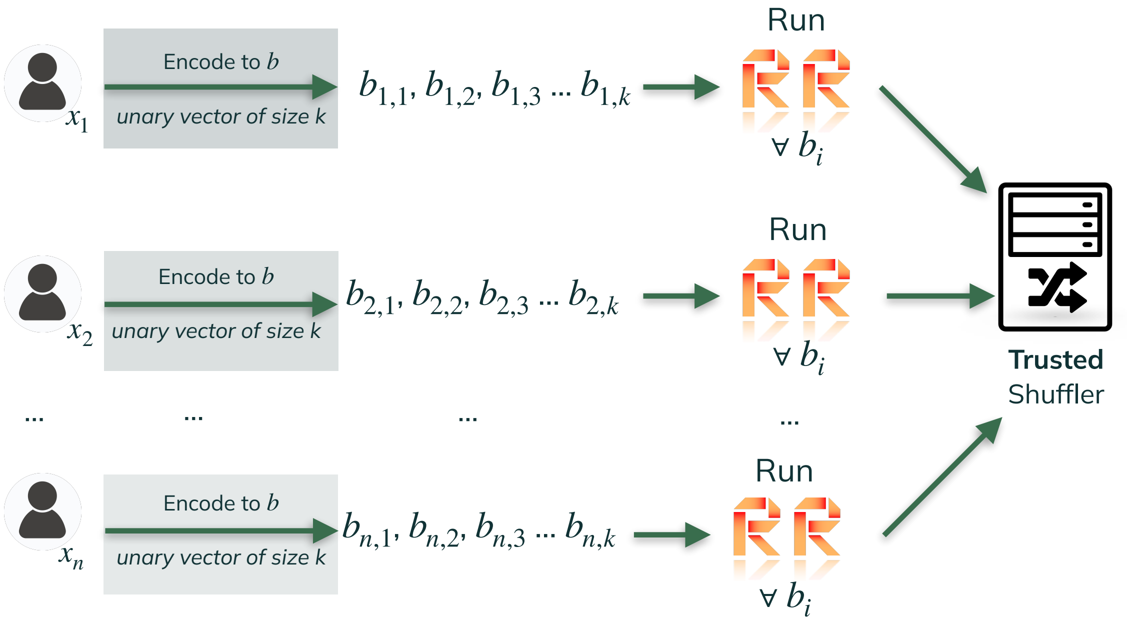
… … … …

x_n — Encode to b → unary vector of size k — $b_{n,1}, b_{n,2}, b_{n,3} \dots b_{n,k}$ → Run RR $\forall\, b_i$

$$\sum_i b_i = x_i$$

RR - Shuffle

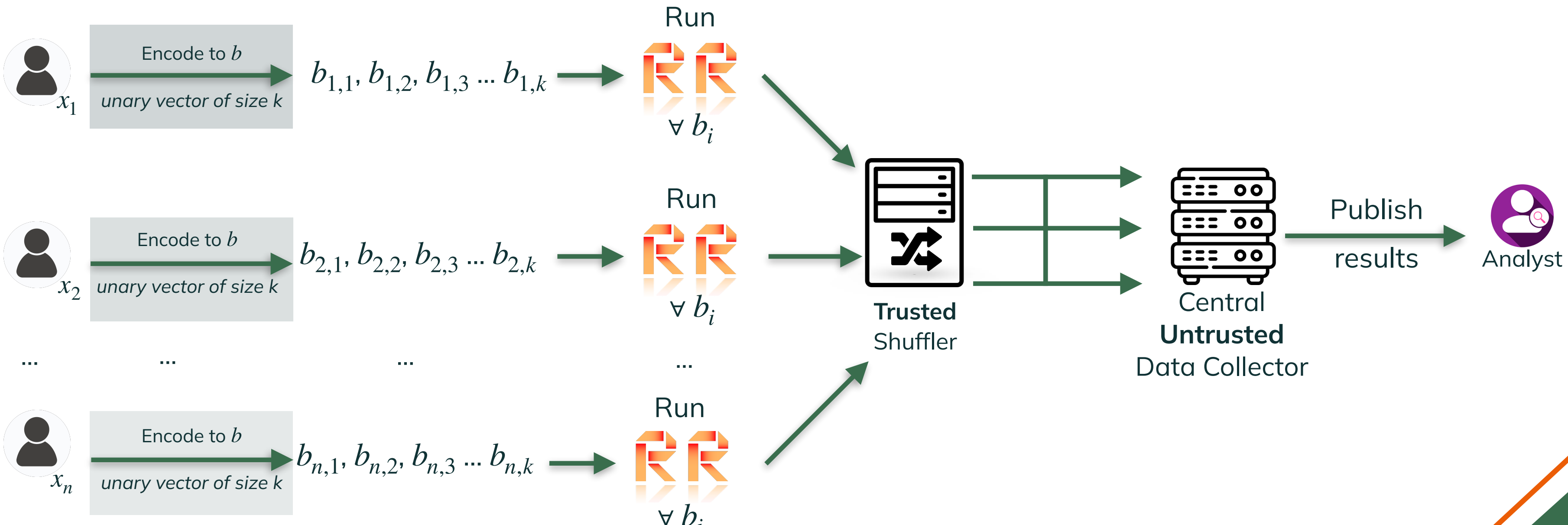
n users | each user **i** has a value $\mathbf{x_i} \in \{0, 1, \ldots, \mathbf{k}\}$



$$\sum_i b_i = x_i$$

RR - Shuffle

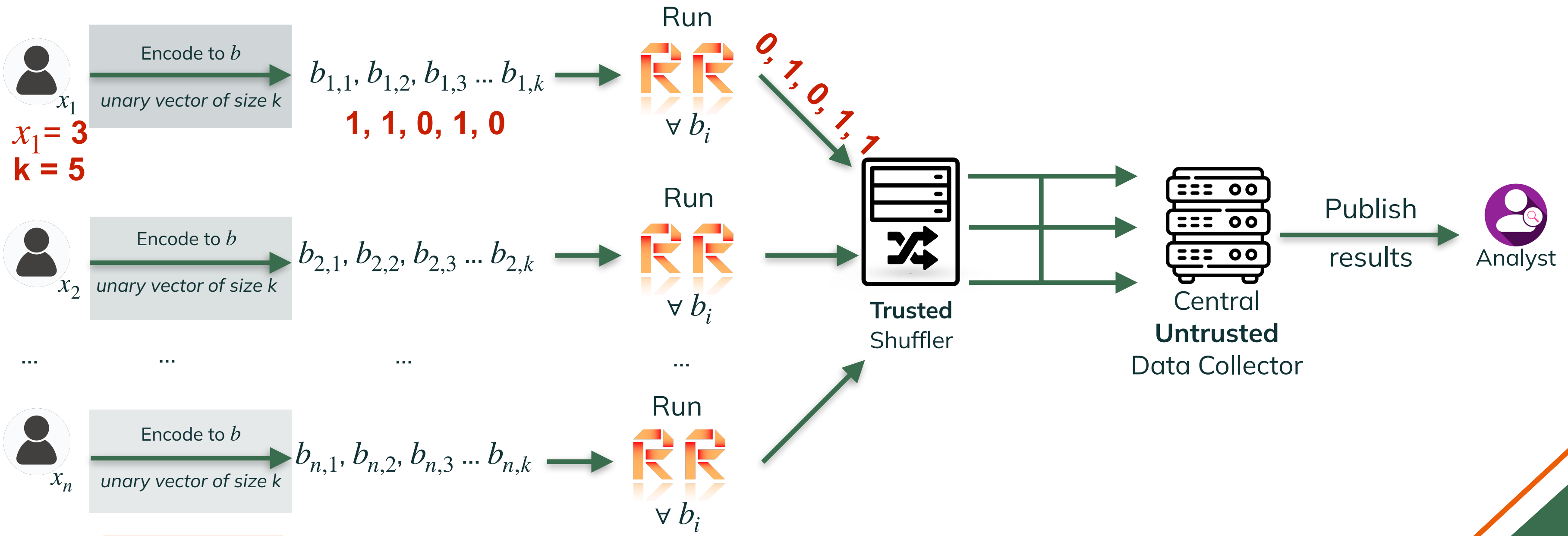
n users | each user **i** has a value $\mathbf{x_i} \in \{0,1, ..., \mathbf{k}\}$



$$\sum_i b_i = x_i$$

RR - Shuffle

n users | each user **i** has a value $\mathbf{x_i} \in \{0,1, \ldots, \mathbf{k}\}$



$$\sum_i b_i = x_i$$

RR - Shuffle: Privacy

Shuffle Model Property

Shuffling unary bits
is privacy-wise equivalent to
Summing unary bits

RR - Shuffle: Privacy

Shuffle Model Property

Shuffling unary bits

is privacy-wise equivalent to

Summing unary bits

Unary Vector

1, 1, 0, 1, 0

sum is 3

=

there are 3 ones and 2 zeros

Integer Vector

5, 8, 30, 42, 1

sum is 86

≠

there is a 5, an 8, a 30, a 42 and a 1

RR - Shuffle: Privacy

Shuffle Model Property

Shuffling unary bits
is privacy-wise equivalent to
Summing unary bits

Binomial Distribution

The sum of all the random bits follows the Binomial Distribution.

RR-Shuffle needs a minimum number of users!

RR - Shuffle: Privacy

Shuffle Model Property

Shuffling unary bits
is privacy-wise equivalent to
Summing unary bits

ε Binomial Distribution

The sum of all the random bits follows the Binomial Distribution.

RR-Shuffle needs a minimum number of users!

δ Rare cases

- The **number** of **random bits** is *unexpectedly "small/large"*
- The **result** of **random bits** is *unexpectedly "small/large"*

Other Mechanisms

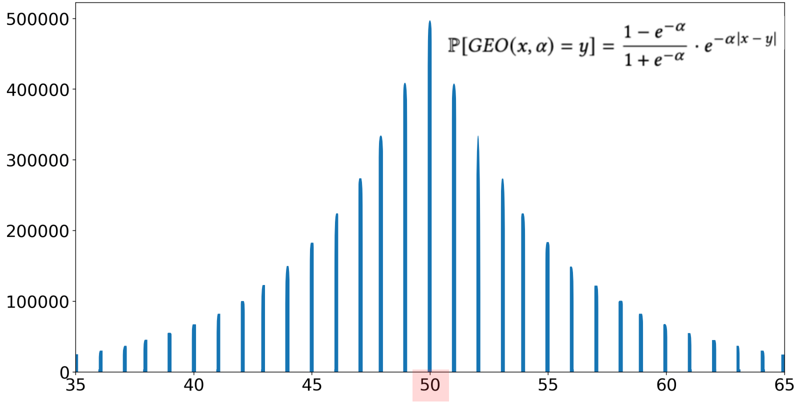
geo - Shuffle

Users sample noise from:

Geometric Mechanism

Applied with a parameter α to a user's value x

Produces y with exponentially decreasing probability wrt $d(x, y)$



Histogram:
Reported values of the Geometric Mechanism with parameter $\alpha = 0.2$ and input value $x = 50$

Other Mechanisms

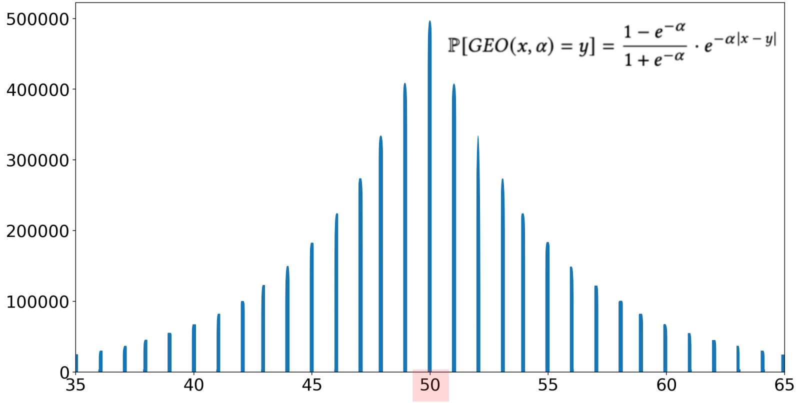
$\mathfrak{geo}$ - Shuffle

Users sample noise from:

Geometric Mechanism

Applied with a parameter α to a user's value x

Produces y with exponentially decreasing probability wrt $d(x, y)$



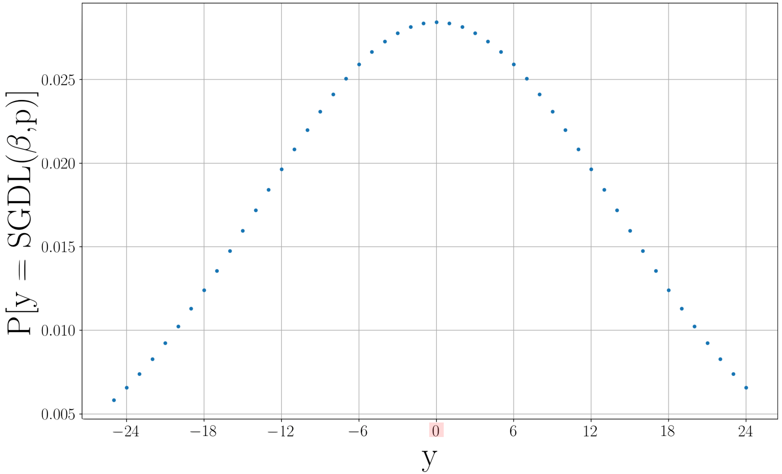
Histogram:
Reported values of the Geometric Mechanism with parameter $\alpha = 0.2$ and input value $x = 50$

$SGDL$ - Shuffle

Users sample noise from:

Symmetric Generalised Discrete Laplace distribution (β, p)

:= difference between two **Negative Binomial** distributions $(\beta, 1-p)$



PMF of SGDL:
Probability to report y for $\beta = 50, p = 0.5$

Other Mechanisms

$\mathfrak{geo}$ - Shuffle

Users sample noise from:

Geometric Mechanism

+

Unary Encode

+

Shuffle

$SGDL$ - Shuffle

Users sample noise from:

Symmetric Generalised Discrete Laplace distribution (β, p)

+

Unary Encode

+

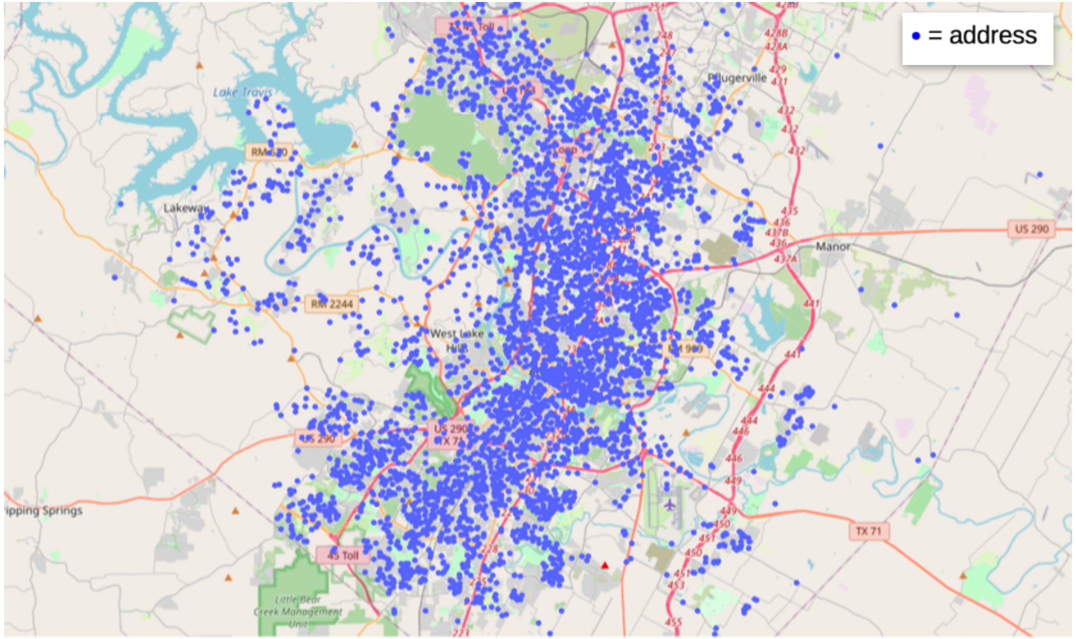
Shuffle

Comparison of Mechanisms

RR - Shuffle	• Simple	• **“Not great, not terrible”** utility • Needs a **minimum** number of **users**
geo **- Shuffle**	• **Excellent** utility • **Medium trust** on the **shuffler**: *the protocol retains some* **privacy** *even if the* **shuffler** *has been* **compromised**	• **Not optimal** utility
SGDL **- Shuffle**	• **Optimal** utility	• **Heavy trust** on the **shuffler**: *the protocol provides almost* **no privacy** *if the* **shuffler** *has been* **compromised**

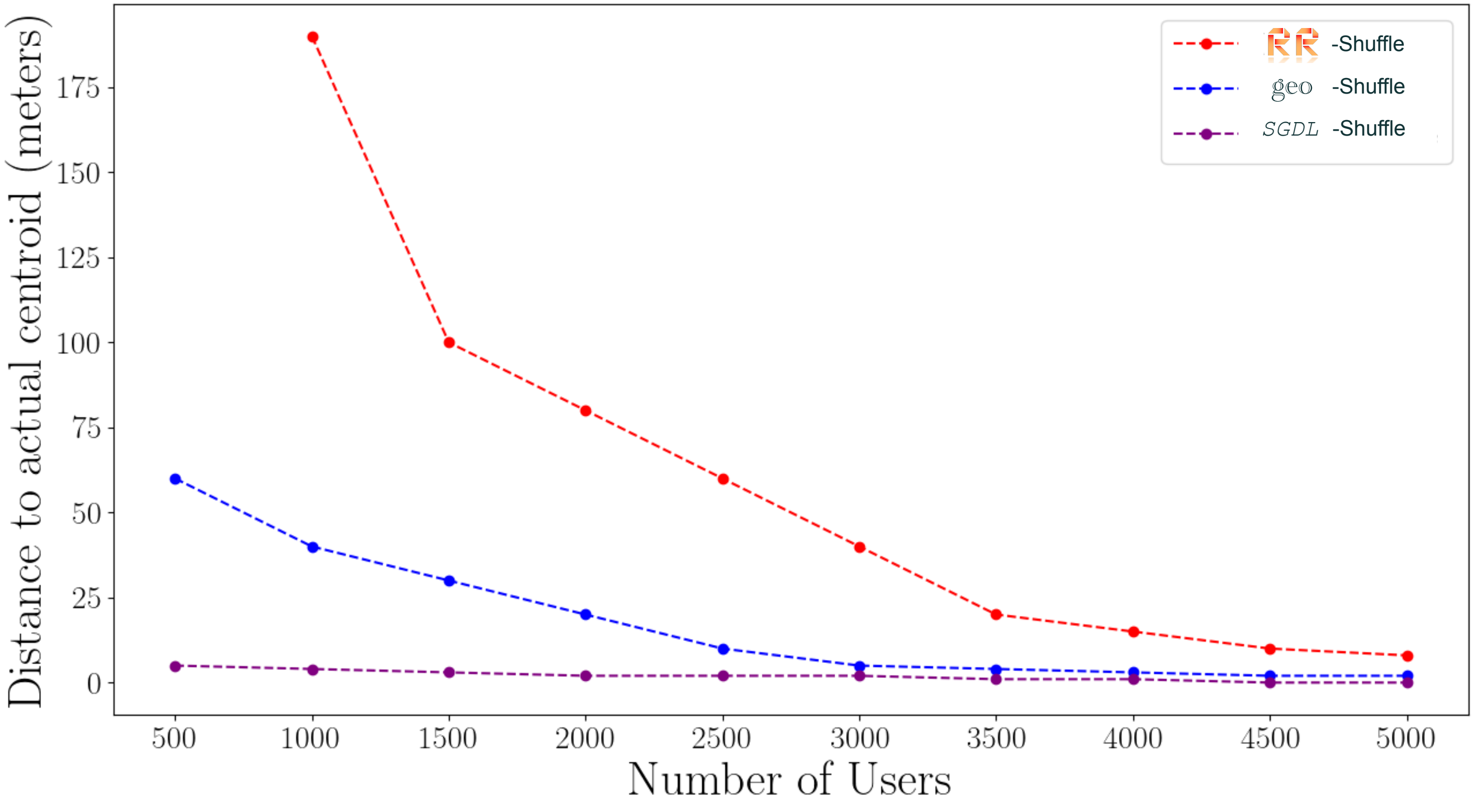
Utility Experiment

Find the centroid of addresses in **Austin, Texas**



$$\varepsilon = 0.15 \mid \delta = 10^{-4}$$

Utility Experiment: Results



Thank you :)

Questions?

Andreas Athanasiou
andreas.athanasiou@inria.fr