Securing Data with Local Differential Privacy: Concepts, Protocols, and Practical Applications

Héber H. Arcolezi (Inria, France)

Selected Areas in Cryptography (SAC) Summer School, 2024

August 27th, 2024

Aims of This Tutorial

To introduce/motivate the privacy model of Local Differential Privacy (LDP):

- Provide technical understanding, scaling of basic LDP protocols.
- Show how some of these LDP protocols that have been used in practice.
- Analysis beyond utility → Privacy and security analysis of LDP protocols.

To suggest directions for future research:

- Identify topics that have just recently been considered.
- Suggest open problems and grand challenges for the area.

Outline

- Module 1 (Introduction):
  - Review of DP and preliminaries
  - LDP introduction
  - State-of-the-art deployments of LDP

- Module 2 (Current research directions):
  - Privacy attacks on LDP protocols
  - Security attacks on LDP protocols
  - Final remarks & open problems

Context

Privacy Leakages in Legal Data Access/Release

- Privacy risks even when access to data is legal:
  - Open datasets (*e.g.*, Census) can allow adversaries to re-identify individuals.
  - Machine learning models subject to attacks (*e.g.*, membership inference).
  - …

Privacy Leakages in Legal Data Access/Release

- Privacy risks even when access to data is legal:
    - Open datasets (*e.g.*, Census) can allow adversaries to re-identify individuals.
    - Machine learning models subject to attacks (*e.g.*, membership inference).
    - …

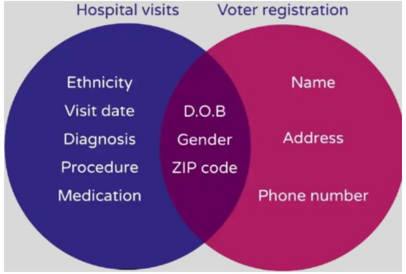
- Maybe we can just remove personally identifying information?
    - Proxy information in the data itself.
    - Multiple sources/background information.
    - "Attackers" may be smarter than we think.

Name	Sex	Blood	···	HIV?
Chen	F	B	···	Y
Jones	M	A	···	N
Smith	M	O	···	N
Ross	M	O	···	Y
Lu	F	A	···	N
Shah	M	B	···	Y

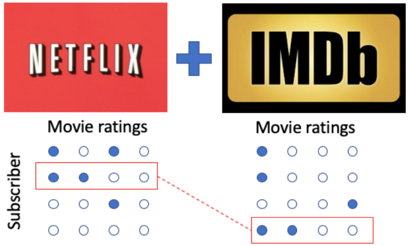
Data “Anonymization” Is Not Safe

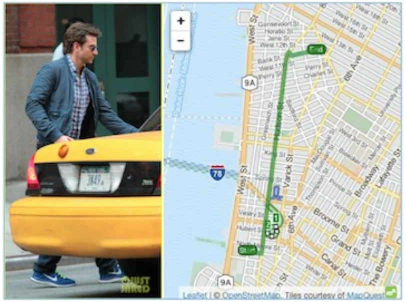
“Oops, we did it again”:

- De-identification (GIC, Sweeney, 2000)
- …
- De-identification (AOL Search Queries, 2006)
- De-identification (Netflix, 2007)
- …
- De-identification (NYC Taxis, 2014)
- …
- De-identification (coming soon in a place near you [C22])...



Thelma Arnold's identity was betrayed by AOL records of her Web searches, like ones for her dog, Dudley, who clearly has a problem.
Erik S. Lesser for The New York Times





[C22] Attacks on Deidentification's Defenses. USENIX Security 2022.

Aggregate Statistics Are Not Safe

How about releasing aggregate statistics about many individuals?

- Problem 1 (Differencing attacks). Combining aggregate queries to obtain precise information about specific individuals.
    - Average salary in a company before and after an employee joins.

Aggregate Statistics Are Not Safe

How about releasing aggregate statistics about many individuals?

- Problem 1 (Differencing attacks). Combining aggregate queries to obtain precise information about specific individuals.
    - Average salary in a company before and after an employee joins.
- Problem 2 (Membership inference attacks) [HSRD...C08, SSSS17]. Inferring presence of known individual in a dataset from (high-dimensional) aggregate statistics.
    - Statistics about genomic variants (*e.g.*, GWAS) or attacks to machine learning models.

[HSRD...C08] Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. PLoS genetics, 2008.
[SSSS17] Membership inference attacks against machine learning models. IEEE S&P 2017.

Aggregate Statistics Are Not Safe

How about releasing aggregate statistics about many individuals?

- Problem 1 (Differencing attacks). Combining aggregate queries to obtain precise information about specific individuals.
    - Average salary in a company before and after an employee joins.
- Problem 2 (Membership inference attacks) [HSRD...C08, SSSS17]. Inferring presence of known individual in a dataset from (high-dimensional) aggregate statistics.
    - Statistics about genomic variants (*e.g.*, GWAS) or attacks to machine learning models.
- Problem 3 (Reconstruction attacks) [DN03]. Inferring (part of) the dataset from the output of many aggregate queries.
    - US Census Bureau's reconstruction attack.

[HSRD...C08] Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. PLoS genetics, 2008.
[SSSS17] Membership inference attacks against machine learning models. IEEE S&P 2017.
[DN03] Revealing information while preserving privacy. PODS 2003.

“Fundamental Law of Information Recovery” [DN03]

Fact #1. Every time you release any statistic calculated from a confidential data source, you “leak” a small amount of private information.

Fact #2. Giving overly accurate answers to too many questions will inevitably “destroy privacy”.

[DN03] Revealing information while preserving privacy. PODS 2003.

Summary of The Key Issues/Requirements

1. Auxiliary knowledge (also called background knowledge or side information): we need to be robust to whatever knowledge the adversary may have, since we cannot predict what an adversary knows or might know in the future.

2. Multiple analyses: we need to be able to track how much information is leaked when asking several questions about the same data and avoid catastrophic leaks.

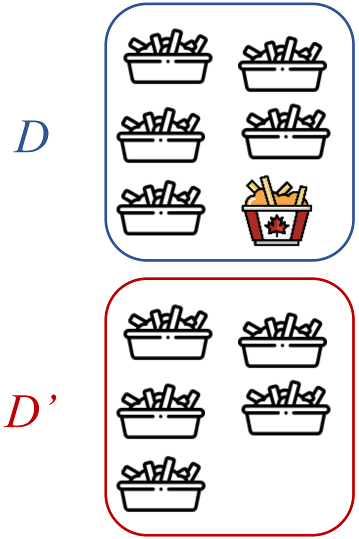
Outline

- **Module 1 (Introduction):**
  - **Review of DP and preliminaries**
  - LDP introduction
  - State-of-the-art deployments of LDP

- Module 2 (Current research directions):
  - Privacy attacks on LDP protocols
  - Security attacks on LDP protocols
  - Final remarks & open problems

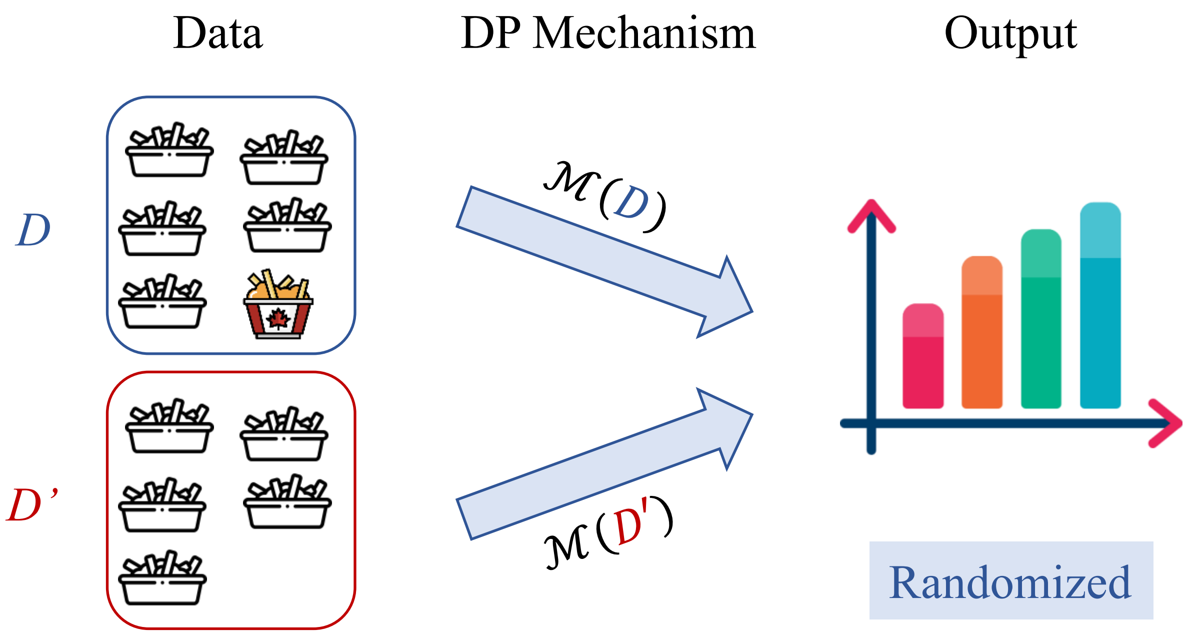
Differential Privacy (DP) [DMNS06]

Data



[DMNS06] Calibrating noise to sensitivity in private data analysis. TCC 2006.

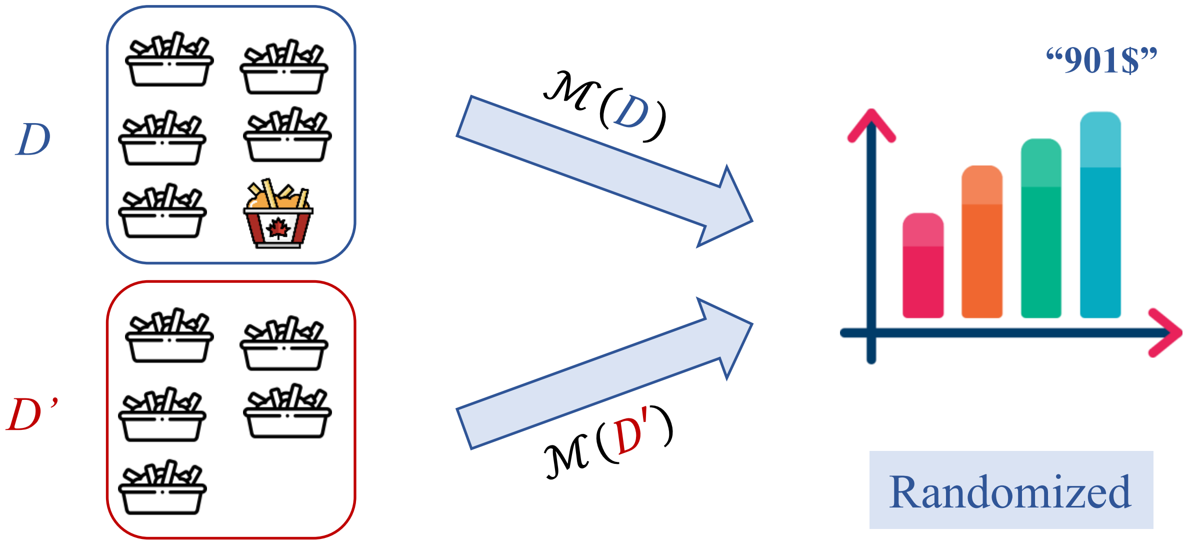
Differential Privacy (DP) [DMNS06]



[DMNS06] Calibrating noise to sensitivity in private data analysis. TCC 2006.

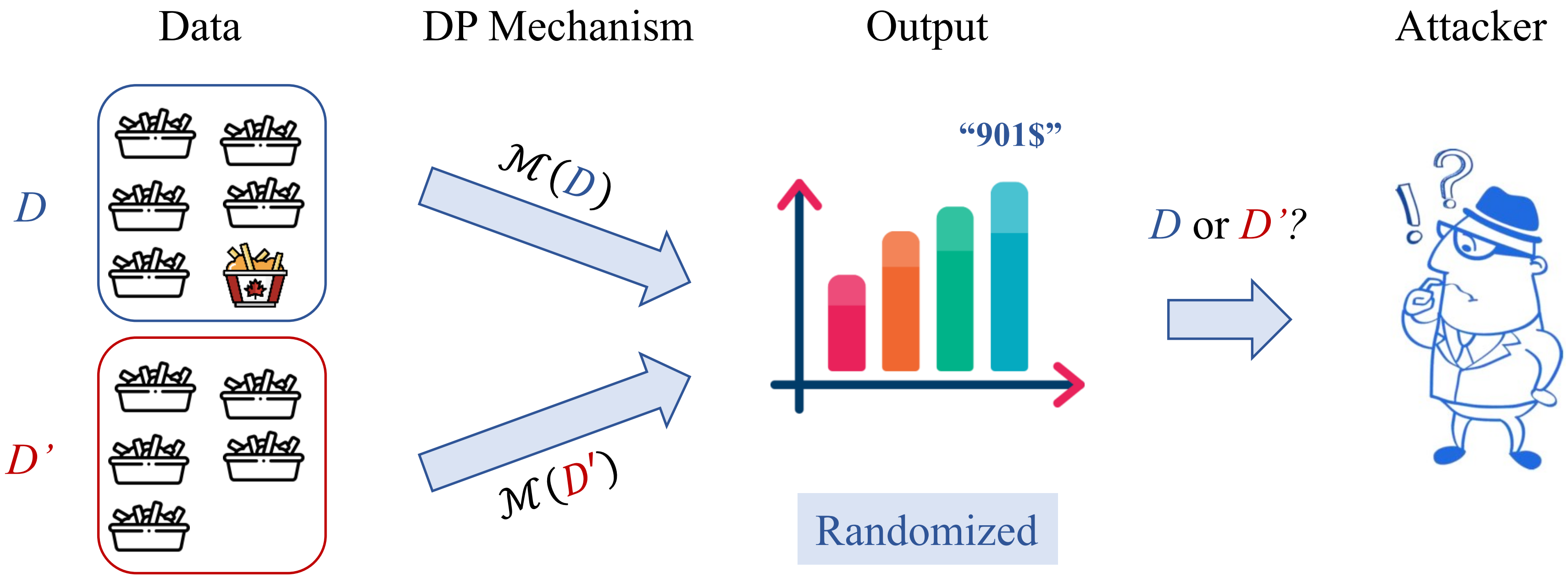
Differential Privacy (DP) [DMNS06]

Data DP Mechanism Output



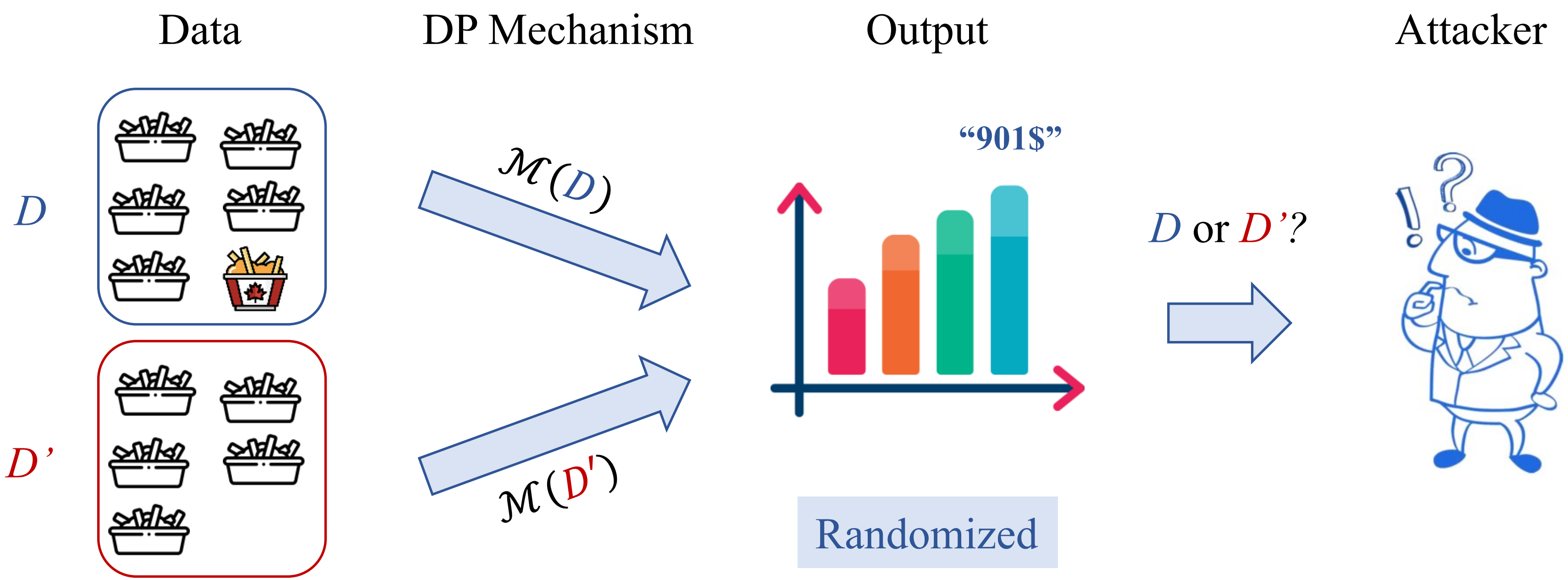
[DMNS06] Calibrating noise to sensitivity in private data analysis. TCC 2006.

Differential Privacy (DP) [DMNS06]



[DMNS06] Calibrating noise to sensitivity in private data analysis. TCC 2006.

Differential Privacy (DP) [DMNS06]



The attacker cannot tell if [your data] was used in the analysis!
"your data"

[DMNS06] Calibrating noise to sensitivity in private data analysis. TCC 2006.

The Math of Differential Privacy [DMNS06]

Definition (Differential Privacy).

Let $\epsilon > 0$, a randomized mechanism $\mathcal{M}$ satisfies ϵ-differential privacy (ϵ-DP), if for any two neighbouring databases D and D' and for any output $z \in \text{Range}(\mathcal{M})$:

$$\frac{\Pr[\mathcal{M}(D) = z]}{\Pr[\mathcal{M}(D') = z]} \leq e^{\epsilon}$$

[DMNS06] Calibrating noise to sensitivity in private data analysis. TCC 2006.

The Math of Differential Privacy [DMNS06]

Definition (Differential Privacy).

Let $\epsilon > 0$, a randomized mechanism $\mathcal{M}$ satisfies ϵ-differential privacy (ϵ-DP), if for any two neighbouring databases D and D' and for any output $z \in \text{Range}(\mathcal{M})$:

$$\frac{\Pr[\mathcal{M}(D) = z]}{\Pr[\mathcal{M}(D') = z]} \leq e^{\epsilon}$$

- Informally, DP requires any single user to have only a limited impact on the output.
- ϵ is called the privacy parameter, the privacy loss, or the privacy budget.
- Privacy is a property of the analysis, not of a particular output.

[DMNS06] Calibrating noise to sensitivity in private data analysis. TCC 2006.

The Math of Differential Privacy [DMNS06]

Definition (Differential Privacy).

Let $\epsilon > 0$, a randomized mechanism $\mathcal{M}$ satisfies ϵ-differential privacy (ϵ-DP), if for any two neighbouring databases D and D' and for any output $z \in \text{Range}(\mathcal{M})$:

$$\frac{\Pr[\mathcal{M}(D) = z]}{\Pr[\mathcal{M}(D') = z]} \leq e^{\epsilon}$$

- Informally, DP requires any single user to have only a limited impact on the output.
- ϵ is called the privacy parameter, the privacy loss, or the privacy budget.
- Privacy is a property of the analysis, not of a particular output.

Key Takeaway. The DP definition promises a worst-case guarantee, the worst that could happen against an adversary who knows pretty much everything besides the sensitive data itself.

Side information? ✓ Computational resources? ✓ Arbitrary priors? ✓

[DMNS06] Calibrating noise to sensitivity in private data analysis. TCC 2006.

Properties of Differential Privacy

- DP is immune to post-processing: it is impossible to compute a function of the output of the private algorithm and make it less differentially private.

> If $\mathcal{M}$ is ϵ-DP, then the composition $f(\mathcal{M})$ is ϵ-DP for any function f.

Properties of Differential Privacy

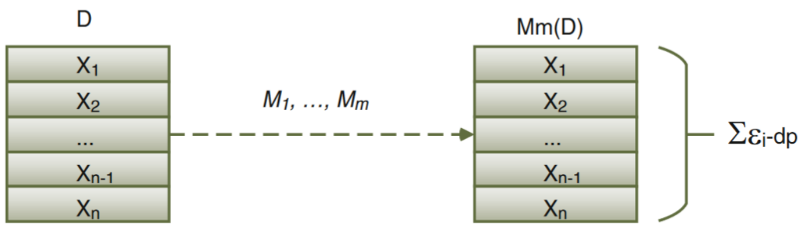
- DP is immune to post-processing: it is impossible to compute a function of the output of the private algorithm and make it less differentially private.

> If $\mathcal{M}$ is ϵ-DP, then the composition $f(\mathcal{M})$ is ϵ-DP for any function f.

- Therefore, additional data post-processing can also be used to address issues such as:
  - Ensuring non-negativity (*e.g.*, there is no negative number of people).
  - Ensuring the sum of the whole population for attribute A is equal to the sum (of the same population) for attribute B.

Properties of Differential Privacy

- DP is robust under composition: If multiple analyses are performed on the same data, if each one satisfies DP, all the information released taken together will still satisfy DP (albeit with a degradation in the privacy parameter).
- Simple rules for composition of DP mechanisms. Let $\mathcal{M}_1$ be ϵ_1-DP and $\mathcal{M}_2$ be ϵ_2-DP:
  - **(Sequential composition)** If inputs overlap, the composed mechanism $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2)$ is $(\epsilon_1 + \epsilon_2)$-DP.



Sequential Composition

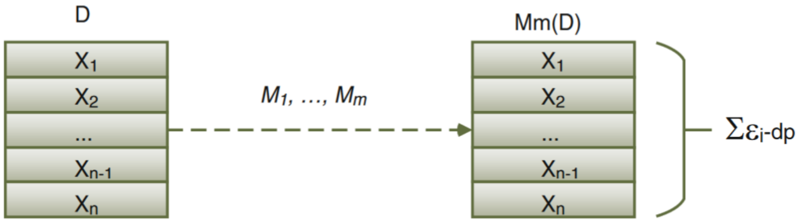
Properties of Differential Privacy

- DP is robust under composition: If multiple analyses are performed on the same data, if each one satisfies DP, all the information released taken together will still satisfy DP (albeit with a degradation in the privacy parameter).

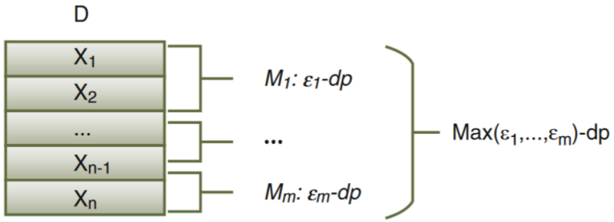
- Simple rules for composition of DP mechanisms. Let $\mathcal{M}_1$ be ϵ_1-DP and $\mathcal{M}_2$ be ϵ_2-DP:

    - **(Sequential composition)** If inputs overlap, the composed mechanism $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2)$ is $(\epsilon_1 + \epsilon_2)$-DP.

    - **(Parallel composition)** If inputs disjoint, the composed mechanism $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2)$ is $\max(\epsilon_1, \epsilon_2)$-DP.



Sequential Composition



Parallel Composition

Satisfying ϵ-DP in the Centralized Setting

Example:

- Satisfy ϵ-DP for counting queries by adding a **random noise value**.
- **Uncertainty** due to noise → **plausible deniability.**

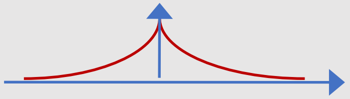
(Global) sensitivity of query f:

$$s = \max_{D,D'} |f(D) - f(D')|, \text{ where } D \text{ and } D' \text{ are neighbors.}$$

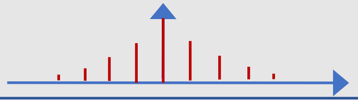
$s = 1$ for counting queries.

For every value that is output:

- Add Laplace noise: $z = f(D) + \mathrm{Lap}(s/\epsilon)$.



- Or Geometric noise (discrete).



Example of Differentially Private Data Publishing

- “True” microdata D $(n = 100)$:

Sex	School
Male	Never
Male	Never
Male	Never
Male	Attending (x12)
Male	Attending
⋮	
Male	Attending
Male	Past (x33)
⋮	
Male	Past

Sex	School
Female	Never (x4)
⋮	
Female	Never
Female	Attending (x17)
⋮	
Female	Attending
Female	Past (x31)
⋮	
Female	Past

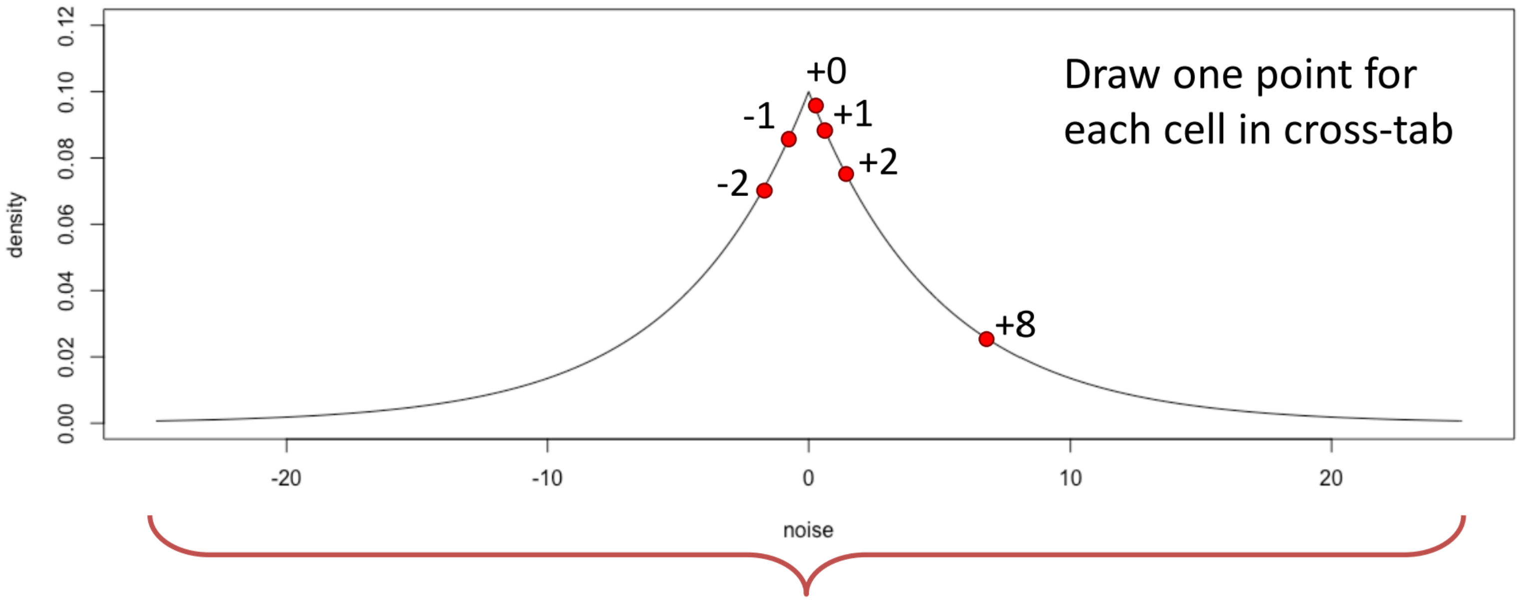
Example of Differentially Private Data Publishing

- Construct cross-tabs (*i.e.*, histogram) from "true" data D ($n = 100$):

	School Attendance		
	Never	Attending	Past
Male	3	12	33
Female	4	17	31

Example of Differentially Private Data Publishing

- Draw noise from Laplace distribution (i.e., Laplace mechanism):



Example of Differentially Private Data Publishing

- Add noise to cross-tab data → $\widetilde{D}$ ($\tilde{n} = 108$):

	School Attendance		
	Never	Attending	Past
Male	3 – 1 = **2**	12 + 0 = **12**	33 + 1 = **34**
Female	4 + 8 = **12**	17 + 2 = **19**	31 – 2 = **29**

Example of Differentially Private Data Publishing

- Construct differentially private microdata $\widetilde{D}$:

Sex	School
Male	Never
Male	Never
Male	Attending
Male	Attending
⋮	⋮
Male	Attending
Male	Past
⋮	⋮
Male	Past

x12 (Male | Attending), x34 (Male | Past)

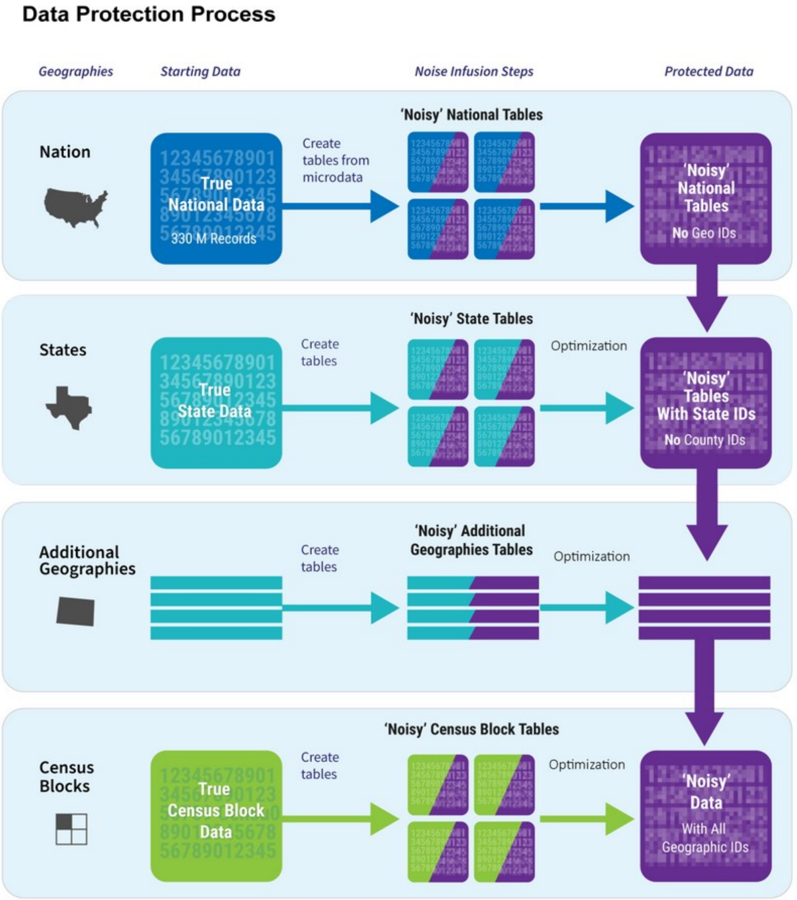
Sex	School
Female	Never
⋮	⋮
Female	Never
Female	Attending
⋮	⋮
Female	Attending
Female	Past
⋮	⋮
Female	Past

x12 (Female | Never), x19 (Female | Attending), x29 (Female | Past)

Real-World Example of Differentially Private Data Publishing

Census TopDown Algorithm (TDA) [AASKLMS19]:

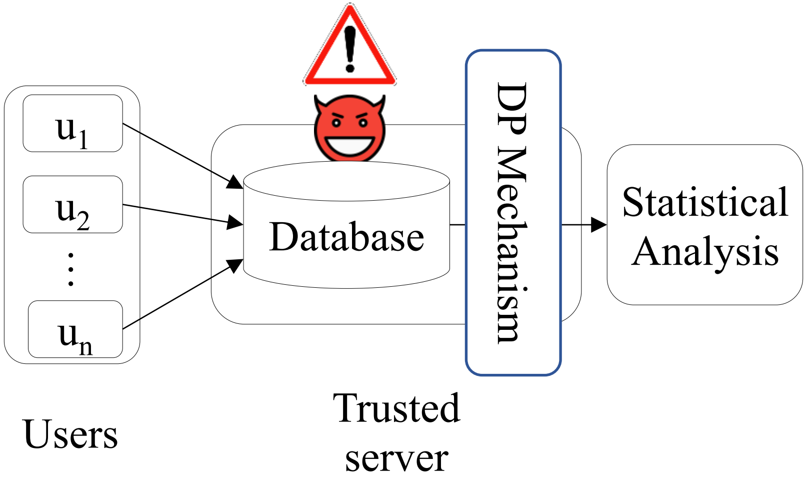
- Computes and protects a histogram for various geographical units at various geographical levels.
- TDA computed statistics, applied noise, and then recomputed statistics at each geographic level of interest, from US, to each state, each county, each census tract, and ultimately each block.

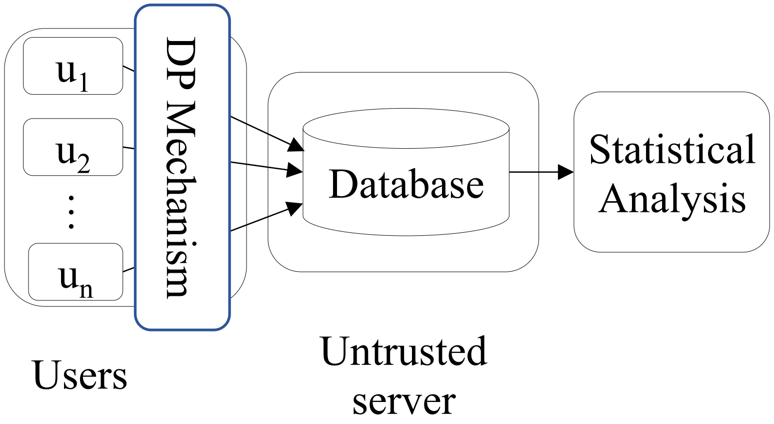


Outline

- **Module 1 (Introduction):**
  - Review of DP and preliminaries
  - **LDP introduction**
  - State-of-the-art deployments of LDP
- Module 2 (Current research directions):
  - Privacy attacks on LDP protocols
  - Security attacks on LDP protocols
  - Final remarks & open problems

What if We Reduce Trust? From Central DP to Local DP





Central DP [DMNS06]:

- ✓ "High utility".
- ✗ Need to trust the server.
- ✗✗ **Data breaches, data misuse, etc.**

Local DP (LDP) [KLNRS11]:

- ✓ No need to trust the server.
- ✗ "Low utility".

[DMNS06] Calibrating noise to sensitivity in private data analysis. TCC 2006.
[KLNRS11] What can we learn privately?. SIAM Journal on Computing 2011.

Local Differential Privacy (LDP) [KLNRS11]

Definition (Local Differential Privacy).

Let $\epsilon > 0$, a randomized mechanism $\mathcal{M}$ satisfies ϵ-local differential privacy (ϵ-LDP), if for any two inputs $v, v' \in \text{Domain}(\mathcal{M})$ and for any output $z \in \text{Range}(\mathcal{M})$:

$$\frac{\Pr[\mathcal{M}(v) = z]}{\Pr[\mathcal{M}(v') = z]} \leq e^{\epsilon}$$

- Informally, any output should be about as likely regardless of the input value.

[KLNRS11] What can we learn privately?. SIAM Journal on Computing 2011.

Local Differential Privacy (LDP) [KLNRS11]

Definition (Local Differential Privacy).

Let $\epsilon > 0$, a randomized mechanism $\mathcal{M}$ satisfies ϵ-local differential privacy (ϵ-LDP), if for any two inputs $v, v' \in \text{Domain}(\mathcal{M})$ and for any output $z \in \text{Range}(\mathcal{M})$:

$$\frac{\Pr[\mathcal{M}(v) = z]}{\Pr[\mathcal{M}(v') = z]} \leq e^{\epsilon}$$

- Informally, any output should be about as likely regardless of the input value.
- Works in LDP consist of designing algorithms with provable upper bounds.
- Properties (like central DP):
    - Post-processing does not consume privacy budget.
    - Sequential and parallel composition hold.

[KLNRS11] What can we learn privately?. SIAM Journal on Computing 2011.

Key Differences Between Central and Local DP

- DP concerns any two neighboring datasets.
  - Let f be the mean query on database D: $z = f(D) + \mathrm{Lap}({}^{s}/_{\epsilon})$.
- LDP concerns any two values.
  - Let user's value v lies in range $[-1, 1]$: $z = v + \mathrm{Lap}({}^{2}/_{\epsilon})$.
  - Server aggregates LDP data to estimate mean: $\tilde{\mu} = \frac{1}{n}\sum_{i=1}^{n} z_i$.

Key Differences Between Central and Local DP

- DP concerns any two neighboring datasets.
    - Let f be the mean query on database D: $z = f(D) + \text{Lap}({}^{s}/_{\epsilon})$.
- LDP concerns any two values.
    - Let user's value v lies in range $[-1, 1]$: $z = v + \text{Lap}({}^{2}/_{\epsilon})$.
    - Server aggregates LDP data to estimate mean: $\tilde{\mu} = \frac{1}{n}\sum_{i=1}^{n} z_i$.
- As a result, the amount of noise is different (each sample).
- So, one seeks to design new LDP algorithms that:
    - Maximize the accuracy of the results.
    - Minimize the costs to the users (*e.g.*, space, time, communication).

Ex. of LDP: Randomized Response (RR) [W65]

- Motivated by surveying people on sensitive/embarrassing topics.
- Main idea → Providing **deniability** to users’ answer (yes/no → binary).
- Ask: “*Did you test positive for HIV (human immunodeficiency virus)?*”

[W65] Randomized response: A survey technique for eliminating evasive answer bias. JASA 1965.

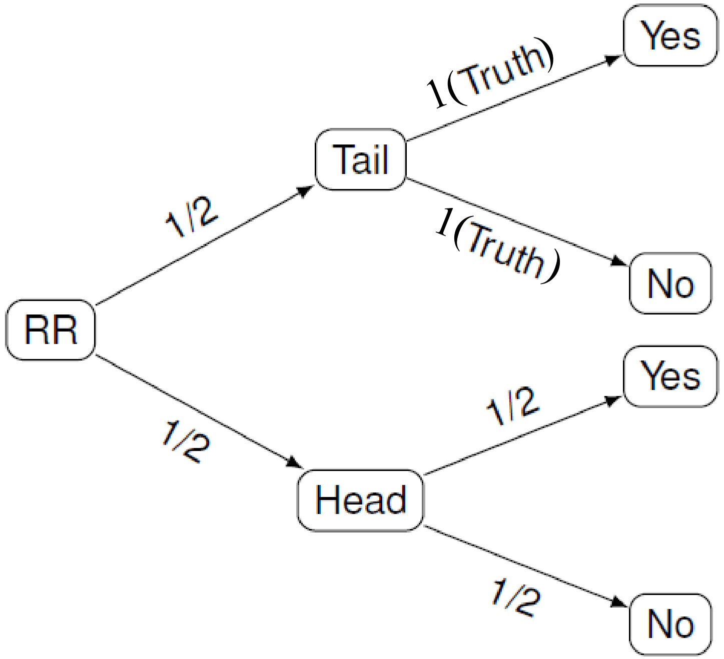
Ex. of LDP: Randomized Response (RR) [W65]

- Motivated by surveying people on sensitive/embarrassing topics.
- Main idea → Providing **deniability** to users' answer (yes/no → binary).
- Ask: "*Did you test positive for HIV (human immunodeficiency virus)?*"
- RR → Throw a secret unbiased coin:
    - If tail, throw the coin again (ignoring the outcome) and answer honestly.
    - If head, then throw the coin again and answer at random, *e.g.*, "Yes" if head, "No" if tail.

Seeing answer, still not certain about the secret.

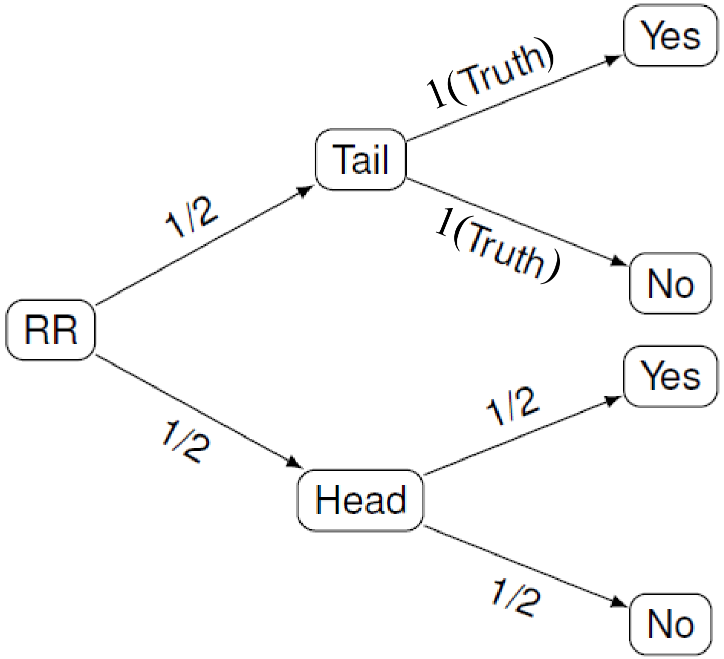
[W65] Randomized response: A survey technique for eliminating evasive answer bias. JASA 1965.

Frequency Estimation and ϵ Study of RR



$$p = \Pr[RR(Yes) = Yes] = \Pr[RR(No) = No] = 0.75$$
$$q = \Pr[RR(No) = Yes] = \Pr[RR(Yes) = No] = 0.25$$

Frequency Estimation and ϵ Study of RR



$p = \Pr[RR(Yes) = Yes] = \Pr[RR(No) = No] = 0.75$

$q = \Pr[RR(No) = Yes] = \Pr[RR(Yes) = No] = 0.25$

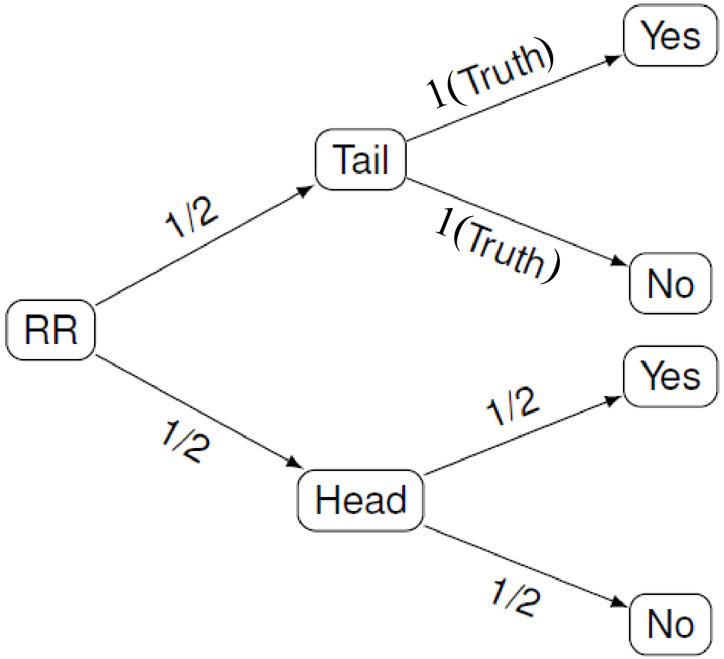
Frequency (or histogram) estimation

$f(v_Y) \rightarrow$ frequency of *true Yes (or No – v_N)*

$C(v_Y) \rightarrow$ frequency of *observed Yes*

- $C(v_Y) \approx \frac{1}{2} f(v_Y) + \frac{1}{4} n$
- $f(v_Y) \approx 2C(v_Y) - \frac{1}{2} n$

Frequency Estimation and ϵ Study of RR



$p = \Pr[RR(Yes) = Yes] = \Pr[RR(No) = No] = 0.75$
$q = \Pr[RR(No) = Yes] = \Pr[RR(Yes) = No] = 0.25$

Frequency (or histogram) estimation

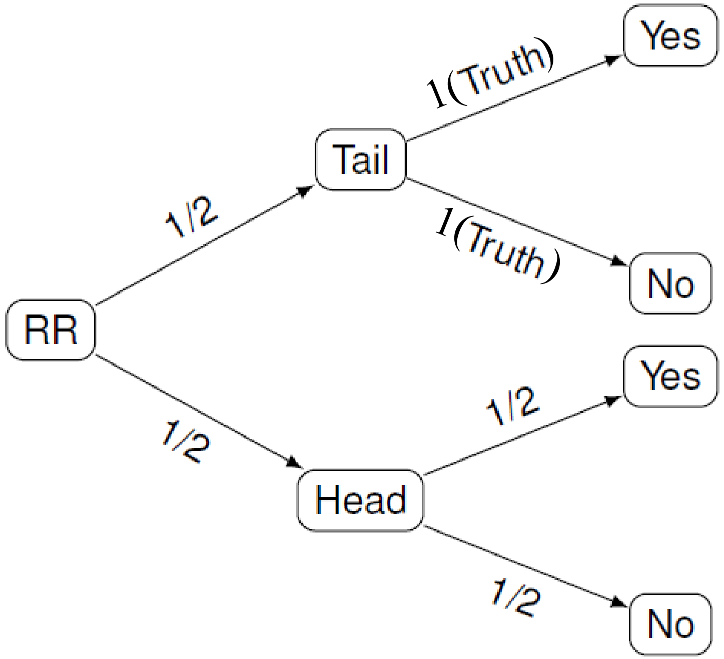
$f(v_Y) \rightarrow$ frequency of *true Yes (or No –* v_N*)*

$C(v_Y) \rightarrow$ frequency of *observed Yes*

- $C(v_Y) \approx \frac{1}{2} f(v_Y) + \frac{1}{4} n$
- $f(v_Y) \approx 2C(v_Y) - \frac{1}{2} n \quad \approx \hat{f}(v) = \frac{C(v) - nq}{(p-q)}, \forall_{v \in \{v_Y, v_N\}}$

Estimated frequency

Frequency Estimation and ϵ Study of RR



Frequency (or histogram) estimation

$f(v_Y) \rightarrow$ frequency of *true* *Yes (or No – v_N)*

$C(v_Y) \rightarrow$ frequency of *observed* *Yes*

- $C(v_Y) \approx \frac{1}{2} f(v_Y) + \frac{1}{4} n$
- $f(v_Y) \approx 2C(v_Y) - \frac{1}{2} n \quad \approx \hat{f}(v) = \frac{C(v)-nq}{(p-q)}, \forall_{v \in \{v_Y, v_N\}}$

Estimated frequency

RR satisfies ϵ-LDP w/:

$p = \Pr[RR(Yes) = Yes] = \Pr[RR(No) = No] = 0.75$

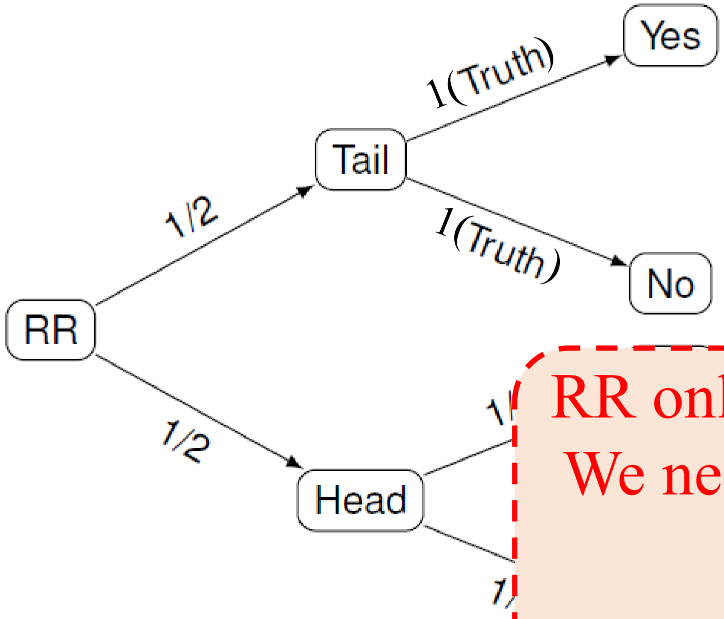
$q = \Pr[RR(No) = Yes] = \Pr[RR(Yes) = No] = 0.25$

$\frac{\Pr(y|x)}{\Pr(y|x')} \leq e^{\epsilon} \Rightarrow e^{\epsilon} = \frac{0.75}{0.25}, \epsilon = \ln(3)$

prob. p of 'being honest'

prob. q of 'lying'

Frequency Estimation and ϵ Study of RR



Frequency (or histogram) estimation

$f(v_Y) \rightarrow$ frequency of *true Yes (or No – v_N)*

$C(v_Y) \rightarrow$ frequency of *observed Yes*

- $C(v_Y) \approx \frac{1}{2} f(v_Y) + \frac{1}{4} n$

Estimated frequency

$\hat{f}(v) = \frac{C(v) - nq}{(p-q)}, \forall_{v \in \{v_Y, v_N\}}$

RR only handles **binary** attribute.
We need a more **general setting**:
- generic ϵ.
- $k \geq 2$.

$p = \Pr[RR(Yes) = Yes] = \Pr[RR(No) = No] = 0.75$

$q = \Pr[RR(No) = Yes] = \Pr[RR(Yes) = No] = 0.25$

$\frac{\Pr(y|x)}{\Pr(y|x')} \leq e^{\epsilon} \Rightarrow e^{\epsilon} = \frac{0.75}{0.25}, \epsilon = \ln(3)$

prob. p of 'being honest'

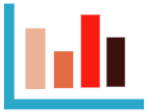
prob. q of 'lying'

LDP Frequency Estimation Protocols

Frequency Estimation Under LDP

Assumption: each user i has a single value v^i from a categorical (or discrete) domain $V = \{v_1, v_2, \dots, v_k\}$ of size $k = |V|$.

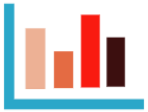
Goal: estimate the frequency (or histogram) of any value $v \in V$.



Frequency Estimation Under LDP

Assumption: each user i has a single value v^i from a categorical (or discrete) domain $V = \{v_1, v_2, \ldots, v_k\}$ of size $k = |V|$.

Goal: estimate the frequency (or histogram) of any value $\mathrm{v} \in V$.



General scheme for frequency estimation under LDP

Input: Original data of users, privacy parameter ϵ, and LDP protocol $\mathcal{M}$.

Output: k-bins histogram.

 # User-side

for each user i with input value $v_i \in V$ **do**:

x_i = **Encode**(v_i) (if needed)

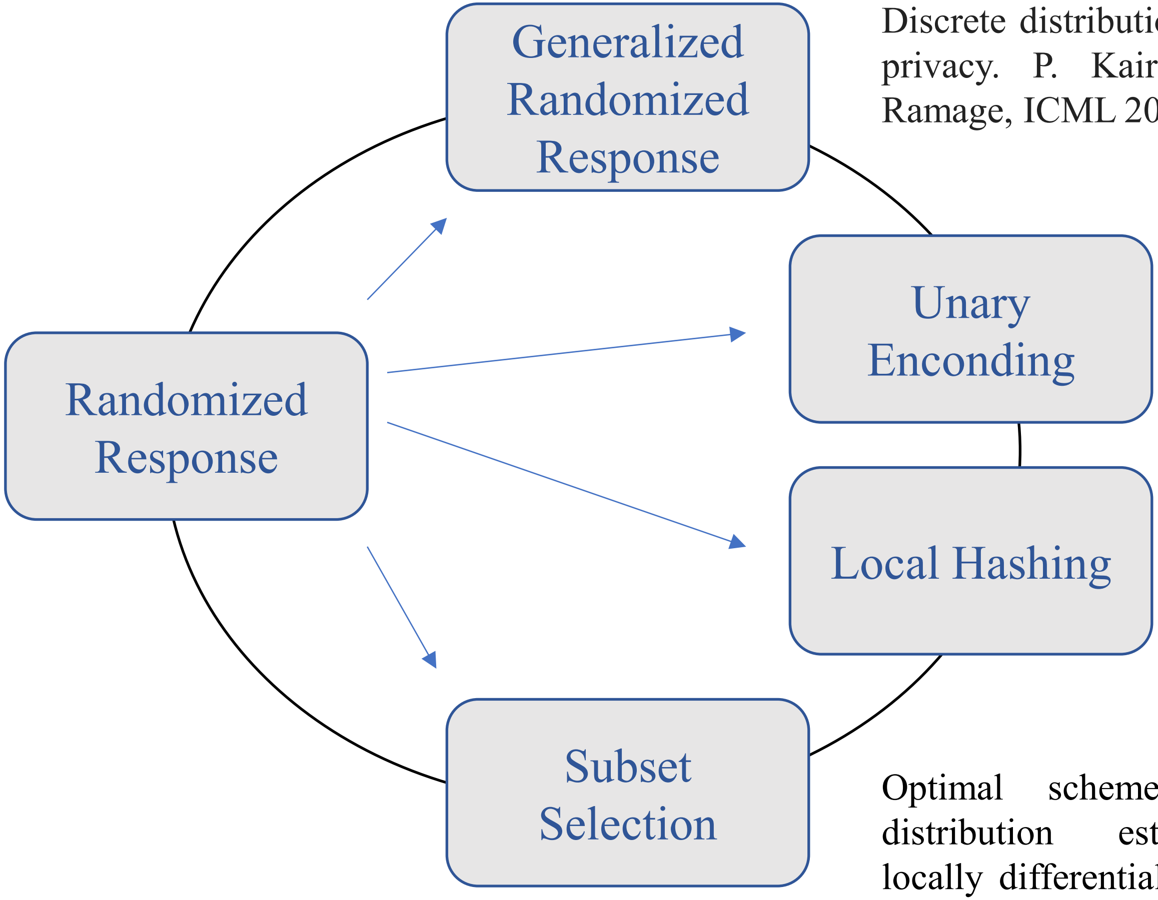
y_i = **Perturb**(x_i) with $\mathcal{M}$

Transmit y_i to the aggregator.

 # Server-side

The server **Aggregates** the reported values and estimate their frequency.

From Two to Many Categories: State-of-the-Art LDP Protocols



Discrete distribution estimation under local privacy. P. Kairouz, K. Bonawitz, D. Ramage, ICML 2016.

RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. Ú. Erlingsson, V. Pihur, A. Korolova, CCS 2014.

Locally Differentially Private Protocols for Frequency Estimation T. Wang, J. Blocki, N. Li, S. Jha: USENIX Security 2017.

Optimal schemes for discrete distribution estimation under locally differential privacy. M. Ye, A. Barg: IEEE TIT 2018.

Generalized Randomized Response (GRR) [KBR16]

 # User-side

- Encode $v = v$ (direct encoding).
- Toss a coin with bias $p = \frac{e^{\epsilon}}{e^{\epsilon}+k-1}$.
- If it is head, report the true value $z = v$.
- Otherwise, report any other value $z = \text{Uniform}(V \setminus \{v\})$ *w.p.* $q = \frac{1-p}{k-1} = \frac{1}{e^{\epsilon}+k-1}$.

[KBR16] Discrete distribution estimation under local privacy. ICML 2016.

Generalized Randomized Response (GRR) [KBR16]

 # User-side

- Encode $v = v$ (direct encoding).
- Toss a coin with bias $p = \frac{e^{\epsilon}}{e^{\epsilon}+k-1}$.
- If it is head, report the true value $z = v$.
- Otherwise, report any other value $z = \text{Uniform}(V \setminus \{v\})$ *w.p.* $q = \frac{1-p}{k-1} = \frac{1}{e^{\epsilon}+k-1}$.
- $\Rightarrow \frac{\Pr[\text{GRR}(v)=y]}{\Pr[\text{GRR}(v')=y]} = \frac{p}{q} = e^{\epsilon}$.

 # Server-side

- $C(v) \rightarrow$ number of times the value $v \in V$ has been reported.
- Unbiased Estimation: $\hat{f}(v) = \frac{C(v)-nq}{(p-q)}$.

[KBR16] Discrete distribution estimation under local privacy. ICML 2016.

Generalized Randomized Response (GRR) [KBR16]

 # User-side

- Encode $v = v$ (direct encoding).
- Toss a coin with bias $p = \frac{e^{\epsilon}}{e^{\epsilon}+k-1}$.
- If it is head, report the true value $z = v$.
- Otherwise, report any other value $z = \text{Uniform}(V \setminus \{v\})$ *w.p.* $q = \frac{1-p}{k-1} = \frac{1}{e^{\epsilon}+k-1}$.
- $\Rightarrow \frac{\Pr[\text{GRR}(v)=y]}{\Pr[\text{GRR}(v')=y]} = \frac{p}{q} = e^{\epsilon}$.

Utility issue: The probability of "being honest" is inversely proportional to k.

 # Server-side

- $C(v) \rightarrow$ number of times the value $v \in V$ has been reported.
- Unbiased Estimation: $\hat{f}(v) = \frac{C(v)-nq}{(p-q)}$.

[KBR16] Discrete distribution estimation under local privacy. ICML 2016.

Unary Encoding (UE) [EPK14, WBLJ17]

 # User-side

- Encode the value v into a bit vector $\vec{\boldsymbol{v}} = \vec{0}$, $\vec{\boldsymbol{v}}[v] = 1$.
- Generate $\vec{\boldsymbol{z}}$ by perturbing each bit in $\vec{\boldsymbol{v}}$ independently *w.p.*:
  - Symmetric UE: $p_{1\to1} = p_{0\to0} = p = \frac{e^{\epsilon/2}}{e^{\epsilon/2}+1}$, $p_{1\to0} = p_{0\to1} = q = \frac{1}{e^{\epsilon/2}+1}$.
  - Optimal UE: $p_{1\to1} = \frac{1}{2}$, $p_{1\to0} = \frac{1}{2}$, $p_{0\to0} = \frac{e^{\epsilon}}{e^{\epsilon}+1}$, $p_{0\to1} = \frac{1}{e^{\epsilon}+1}$.

[EPK14] RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. CCS 2014.
[WBLJ17] Locally Differentially Private Protocols for Frequency Estimation. USENIX Security 2017.

Unary Encoding (UE) [EPK14, WBLJ17]

 # User-side

- Encode the value v into a bit vector $\vec{\boldsymbol{v}} = \vec{0}$, $\vec{\boldsymbol{v}}[v] = 1$.
- Generate $\vec{\boldsymbol{z}}$ by perturbing each bit in $\vec{\boldsymbol{v}}$ independently *w.p.*:
  - Symmetric UE: $p_{1\to1} = p_{0\to0} = p = \frac{e^{\epsilon/2}}{e^{\epsilon/2}+1}$, $p_{1\to0} = p_{0\to1} = q = \frac{1}{e^{\epsilon/2}+1}$.
  - Optimal UE: $p_{1\to1} = \frac{1}{2}$, $p_{1\to0} = \frac{1}{2}$, $p_{0\to0} = \frac{e^{\epsilon}}{e^{\epsilon}+1}$, $p_{0\to1} = \frac{1}{e^{\epsilon}+1}$.

Example:
$v = 2, k = 4$
$\vec{\boldsymbol{v}} = [0, 0, 1, 0]$
$\vec{\boldsymbol{z}} = [0, 0, 1, 1]$

[EPK14] RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. CCS 2014.
[WBLJ17] Locally Differentially Private Protocols for Frequency Estimation. USENIX Security 2017.

Unary Encoding (UE) [EPK14, WBLJ17]

 # User-side

- Encode the value v into a bit vector $\vec{\boldsymbol{v}} = \vec{0}, \vec{\boldsymbol{v}}[v] = 1$.
- Generate $\vec{\boldsymbol{z}}$ by perturbing each bit in $\vec{\boldsymbol{v}}$ independently *w.p.*:
  - Symmetric UE: $p_{1\to1} = p_{0\to0} = p = \frac{e^{\epsilon/2}}{e^{\epsilon/2}+1}$, $p_{1\to0} = p_{0\to1} = q = \frac{1}{e^{\epsilon/2}+1}$.
  - Optimal UE: $p_{1\to1} = \frac{1}{2}$, $p_{1\to0} = \frac{1}{2}$, $p_{0\to0} = \frac{e^{\epsilon}}{e^{\epsilon}+1}$, $p_{0\to1} = \frac{1}{e^{\epsilon}+1}$.
- $\Rightarrow \frac{\Pr[\mathrm{UE}(\vec{\boldsymbol{v}})=\vec{\boldsymbol{z}}]}{\Pr\left[\mathrm{UE}(\vec{\boldsymbol{v'}})=\vec{\boldsymbol{z}}\right]} \le \frac{p_{1\to1}}{p_{0\to1}} \times \frac{p_{0\to0}}{p_{1\to0}} = e^{\epsilon}$.

Example:
$v = 2, k = 4$
$\vec{\boldsymbol{v}} = [0, 0, 1, 0]$
$\vec{\boldsymbol{z}} = [0, 0, 1, 1]$

 # Server-side

- To estimate frequency of each value v, do it for each bit.
- Unbiased Estimation: $\hat{f}(v) = \frac{C(v)-nq}{(p-q)}$.

[EPK14] RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. CCS 2014.
[WBLJ17] Locally Differentially Private Protocols for Frequency Estimation. USENIX Security 2017.

Local Hashing (LH) [BS15,WBLJ17]

 # User-side

- Each user uses a random hash function H that maps $V \rightarrow \{0,1,\ldots,g\}$.
    - Binary LH: $g = 2$.
    - Optimal LH: $g = e^{\epsilon} + 1$.
- The user then perturbs the hashed ("encoded") value with GRR.
- The user reports the perturbed value and the hash function: $\langle \mathrm{GRR}(H(v)), H \rangle$.

[BS15] Local, Private, Efficient Protocols for Succinct Histograms. STOC 2015.
[WBLJ17] Locally Differentially Private Protocols for Frequency Estimation. USENIX Security 2017.

Local Hashing (LH) [BS15,WBLJ17]

 # User-side

- Each user uses a random hash function H that maps $V \rightarrow \{0,1,\ldots,g\}$.
    - Binary LH: $g = 2$.
    - Optimal LH: $g = e^{\epsilon} + 1$.
- The user then perturbs the hashed ("encoded") value with GRR.
- The user reports the perturbed value and the hash function: $\langle \mathrm{GRR}(H(v)), H \rangle$.

Example:
$v = 2, k = 4, g = 2$
$H(v) = 0$
$z = 0$

[BS15] Local, Private, Efficient Protocols for Succinct Histograms. STOC 2015.
[WBLJ17] Locally Differentially Private Protocols for Frequency Estimation. USENIX Security 2017.

Local Hashing (LH) [BS15,WBLJ17]

 # User-side

- Each user uses a random hash function H that maps $V \rightarrow \{0,1,\ldots,g\}$.
    - Binary LH: $g = 2$.
    - Optimal LH: $g = e^{\epsilon} + 1$.
- The user then perturbs the hashed ("encoded") value with GRR.
- The user reports the perturbed value and the hash function: $\langle \text{GRR}(H(v)), H \rangle$.
- $\Rightarrow \frac{\Pr[\text{GRR}(H(v))=z]}{\Pr[\text{GRR}(H(v'))=z]} = \frac{p}{q} \leq e^{\epsilon}$.

 # Server-side

- $C(v) \rightarrow |\{u \in U | H^u(z) = v^u\}|$, $q' = \frac{1}{g}p + \left(1 - \frac{1}{g}\right)q = \frac{1}{g}$.
- Unbiased Estimation: $\hat{f}(v) = \frac{C(v) - nq'}{(p - q')}$.

Example:
$v = 2, k = 4, g = 2$
$H(v) = 0$
$z = 0$

[BS15] Local, Private, Efficient Protocols for Succinct Histograms. STOC 2015.
[WBLJ17] Locally Differentially Private Protocols for Frequency Estimation. USENIX Security 2017.

Subset Selection (SS) [YB18]

 # User-side

- Initialize an empty subset Ω and add v to Ω *w.p.*: $p = \frac{\omega e^{\epsilon}}{\omega e^{\epsilon}+k-\omega}$, where $\omega = \frac{k}{e^{\epsilon}+1}$.
- Finally, add values to Ω as follows:
  - If $v \in \Omega$, sample $\omega - 1$ values (wo/ replacement) from $V \setminus \{v\}$.
  - Else, sample ω values (wo/ replacement) from $V \setminus \{v\}$.

[YB18] Optimal Schemes for Discrete Distribution Estimation Under LDP. TIT 2018.

Subset Selection (SS) [YB18]

 # User-side

- Initialize an empty subset Ω and add v to Ω *w.p.*: $p = \frac{\omega e^{\epsilon}}{\omega e^{\epsilon} + k - \omega}$, where $\omega = \frac{k}{e^{\epsilon}+1}$.
- Finally, add values to Ω as follows:
  - If $v \in \Omega$, sample $\omega - 1$ values (wo/ replacement) from $V \setminus \{v\}$.
  - Else, sample ω values (wo/ replacement) from $V \setminus \{v\}$.

Example:
$v = 2, k = 4, \omega = 2$
$\Omega = \{0, 2\}$

[YB18] Optimal Schemes for Discrete Distribution Estimation Under LDP. TIT 2018.

Subset Selection (SS) [YB18]

 # User-side

- Initialize an empty subset Ω and add v to Ω *w.p.*: $p = \frac{\omega e^{\epsilon}}{\omega e^{\epsilon}+k-\omega}$, where $\omega = \frac{k}{e^{\epsilon}+1}$.
- Finally, add values to Ω as follows:
  - If $v \in \Omega$, sample $\omega - 1$ values (wo/ replacement) from $V \setminus \{v\}$.
  - Else, sample ω values (wo/ replacement) from $V \setminus \{v\}$.
- $\Rightarrow \frac{\Pr[SS(v)=\boldsymbol{\Omega}]}{\Pr[SS(v')=\boldsymbol{\Omega}]} \leq \frac{p(k-\omega)}{\omega(1-p)} = e^{\epsilon}$.

Example:
$v = 2, k = 4, \omega = 2$
$\Omega = \{0, 2\}$

 # Server-side

- $C(v) \rightarrow$ number of times the value $v \in V$ has been reported, $q = \frac{\omega e^{\epsilon}(\omega-1)+(k-\omega)\omega}{(k-1)(\omega e^{\epsilon}+k-\omega)}$.
- Unbiased Estimation: $\hat{f}(v) = \frac{C(v)-nq}{(p-q)}$.

[YB18] Optimal Schemes for Discrete Distribution Estimation Under LDP. TIT 2018.

Probabilistic Analysis [WBLJ17]

Same estimator $\hat{f}(v)$ for all LDP protocols (GRR, SUE, OUE, BLH, OLH, and SS).

- $\hat{f}(v)$ is a random variable.
- The estimation $\hat{f}(v)$ is unbiased: $\mathbb{E}[\hat{f}(v)] = f(v)$.
- (Approximate) variance of $\hat{f}(v)$: $\mathrm{Var}^*\left[\hat{f}(v)/n\right] = \frac{q(1-q)}{n(p-q)^2} + \frac{f(v)(1-p-q)}{n(p-q)}$.
- Since $\hat{f}(v)$ is unbiased, the variance is equal to the MSE metric.
- Transform from variance to error bound.

$f(v) \approx 0$

(Approximate) Variance and Utility Comparison

Variance in terms of k, n, and ϵ.

GRR

$$\frac{k + e^{\epsilon} - 2}{n(1 - e^{\epsilon})^2}$$

SUE

$$\frac{1}{4n \sinh^2\left(\frac{\epsilon}{4}\right)}$$

OUE

$$\frac{1.0}{n \sinh^2\left(\frac{\epsilon}{2}\right)}$$

BLH

$$\frac{1.0}{n \tanh^2\left(\frac{\epsilon}{2}\right)}$$

OLH

$$\frac{1}{n \sinh^2\left(\frac{\epsilon}{2}\right)}$$

SS

$$\frac{(2k - e^{\epsilon} - 1)(-2k + 2(k-1)(e^{\epsilon} + 1) + e^{\epsilon} + 1)}{n(-2k + (k-1)(e^{\epsilon} + 1) + e^{\epsilon} + 1)^2}$$

(Approximate) Variance and Utility Comparison

Variance in terms of k, n, and ϵ.

GRR

$$\frac{k + e^{\epsilon} - 2}{n(1 - e^{\epsilon})^2}$$

SUE

$$\frac{1}{4n \sinh^2\left(\frac{\epsilon}{4}\right)}$$

OUE

$$\frac{1.0}{n \sinh^2\left(\frac{\epsilon}{2}\right)}$$

BLH

$$\frac{1.0}{n \tanh^2\left(\frac{\epsilon}{2}\right)}$$

OLH

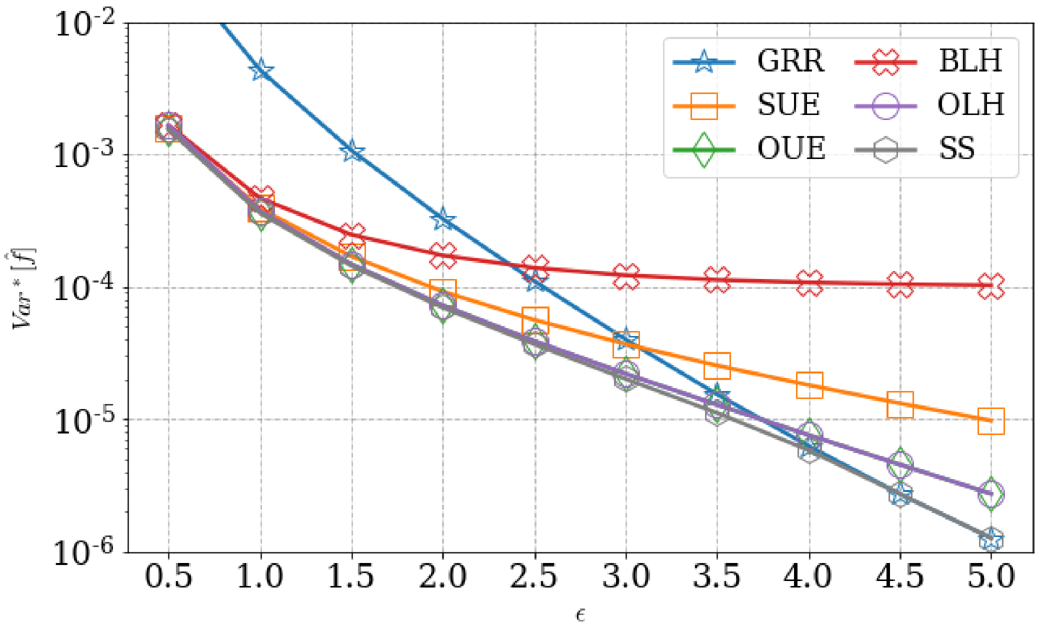
$$\frac{1}{n \sinh^2\left(\frac{\epsilon}{2}\right)}$$

SS

$$\frac{(2k - e^{\epsilon} - 1)\left(-2k + 2(k-1)(e^{\epsilon} + 1) + e^{\epsilon} + 1\right)}{n(-2k + (k-1)(e^{\epsilon} + 1) + e^{\epsilon} + 1)^2}$$

Analytical measure of variance:

e.g., $k = 128$ and $n = 10000$.



Outline

- **Module 1 (Introduction):**
  - Review of DP and preliminaries
  - LDP introduction
  - **State-of-the-art deployments of LDP**

- Module 2 (Current research directions):
  - Privacy attacks on LDP protocols
  - Security attacks on LDP protocols
  - Final remarks & open problems

LDP in Practical Applications

Task: Frequency ("monitoring") estimation throughout time $t \in [\tau]$.

- **Assumption:** each user i has a sequence of values $\vec{\boldsymbol{v}}^i = [v_1^i, \dots, v_\tau^i]$, where v_t^i represents the discrete value v $\in V$ of user i at time $t \in [\tau]$ and $k = |V|$.
- **Goal:** at each time $t \in [\tau]$, estimate the k-bins histogram.

LDP in Practical Applications

Task: Frequency ("monitoring") estimation throughout time $t \in [\tau]$.

- **Assumption:** each user i has a sequence of values $\vec{\boldsymbol{v}}^i = [v_1^i, \ldots, v_\tau^i]$, where v_t^i represents the discrete value v $\in V$ of user i at time $t \in [\tau]$ and $k = |V|$.
- **Goal:** at each time $t \in [\tau]$, estimate the k-bins histogram.

What is the prefered webpage of each user along time?

Time 1   Time 2   …   Time τ

Challenge: Bound the privacy loss ϵ, avoid tracking, and minimize the estimation error.

LDP in Practical Applications

Differential privacy based on “coin tossing” is (or has been) widely deployed!

- In Google Chrome browser, to collect browsing statistics (now deprecated).
- In Microsoft Windows, to collect telemetry data over time.
- In Apple iOS and MacOS, to collect typing statistics.
- In Google Gboard, for out-of-vocabulary word discovery.

LDP in Practical Applications

Differential privacy based on "coin tossing" is (or has been) widely deployed!

- In Google Chrome browser, to collect browsing statistics (now deprecated).
- In Microsoft Windows, to collect telemetry data over time.
- In Apple iOS and MacOS, to collect typing statistics.
- In Google Gboard, for out-of-vocabulary word discovery.

- This yields deployments of over more than 100 million users…
- All deployments are based on RR (improved protocols to handle large k).
- LDP is state-of-the-art in 2024 ↔ RR invented in 1965, **six decades ago**!

Naïve Solution: Repeated Usage of an LDP Protocol

Let a user has a secret sequence $\vec{\boldsymbol{v}} = [v, v, \dots, v]$ (static value for τ time steps):

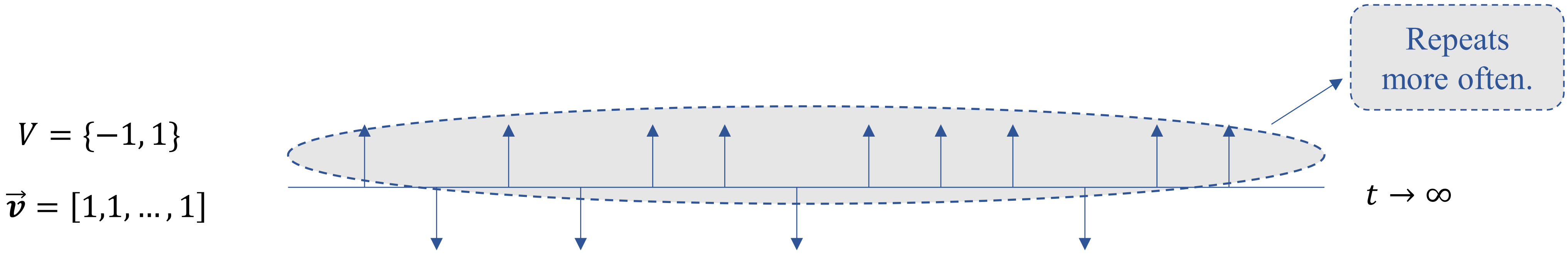
- Naïve solution → At time $t \in [\tau]$, encode/perturb v with an ϵ-LDP protocol.
- Following the sequential composition, the privacy loss is at most $\tau\epsilon$-LDP.
- This solution is subject to "averaging attacks" as t gets large.

Inria

Naïve Solution: Repeated Usage of an LDP Protocol

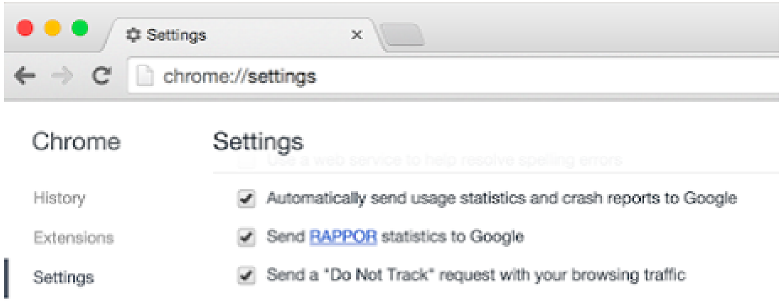
Let a user has a secret sequence $\vec{\boldsymbol{v}} = [v, v, \dots, v]$ (static value for τ time steps):

- Naïve solution → At time $t \in [\tau]$, encode/perturb v with an ϵ-LDP protocol.
- Following the sequential composition, the privacy loss is at most $\tau\epsilon$-LDP.
- This solution is subject to "averaging attacks" as t gets large.
  - For all analyzed LDP protocols (GRR, SUE, OUE, BLH, OLH, and SS) the probability of 'being honest' p is always higher than q.



Google's RAPPOR Solution for Chrome [EPK14]

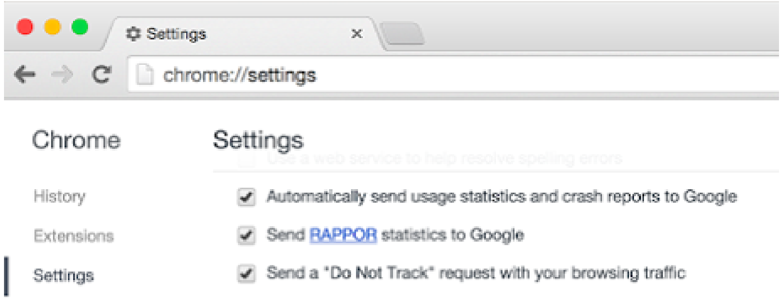
- Each user has one value out of a very large set of possibilities (*e.g.*, favourite URL).
- Reduce domain size through hashing.
- Two obfuscation rounds to avoid tracking.

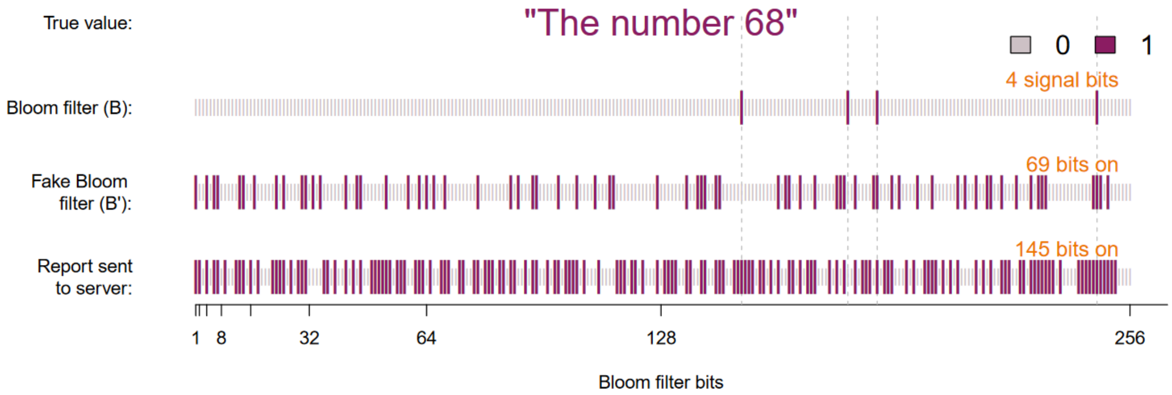


[EPK14] RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. CCS 2014.

Google's RAPPOR Solution for Chrome [EPK14]

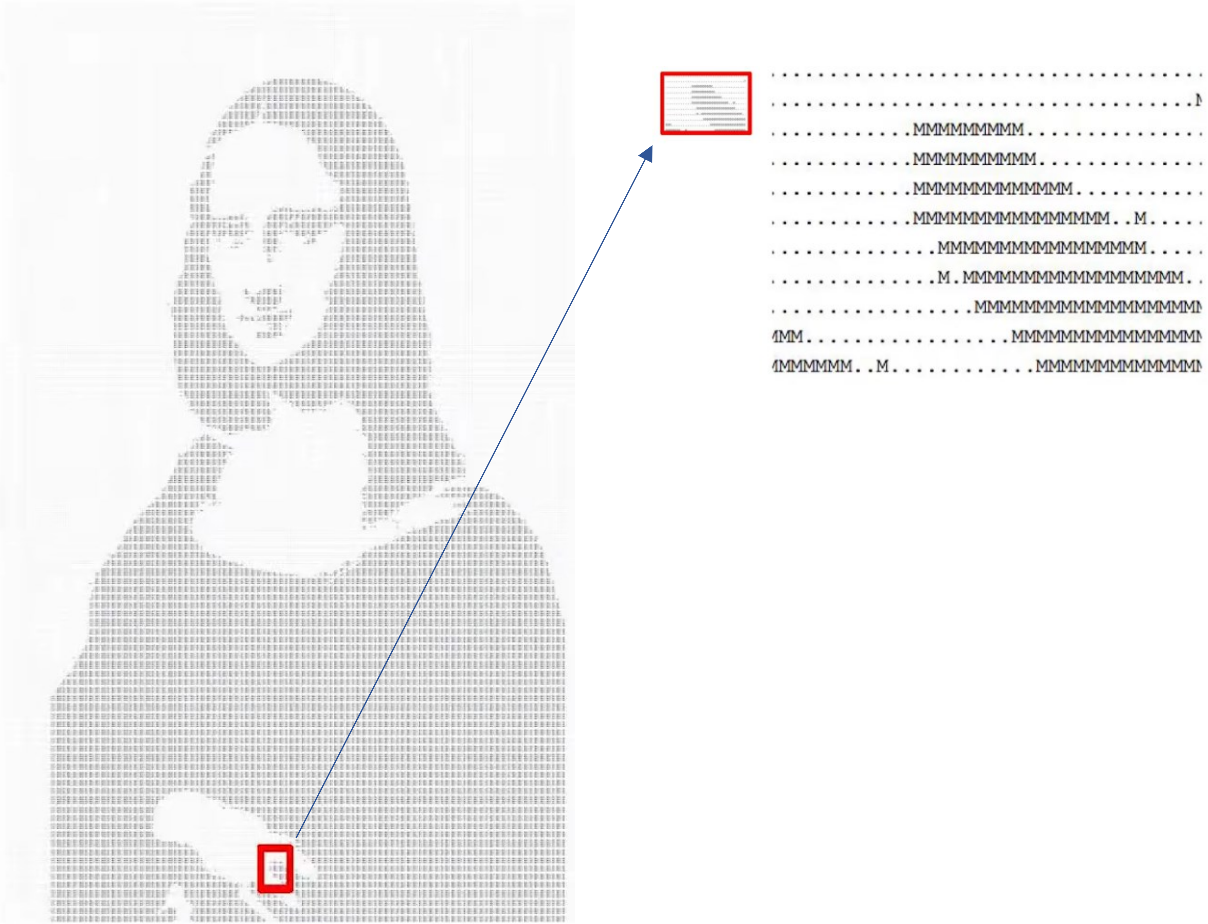
- Each user has one value out of a very large set of possibilities (*e.g.*, favourite URL).
- Reduce domain size through hashing.
- Two obfuscation rounds to avoid tracking.





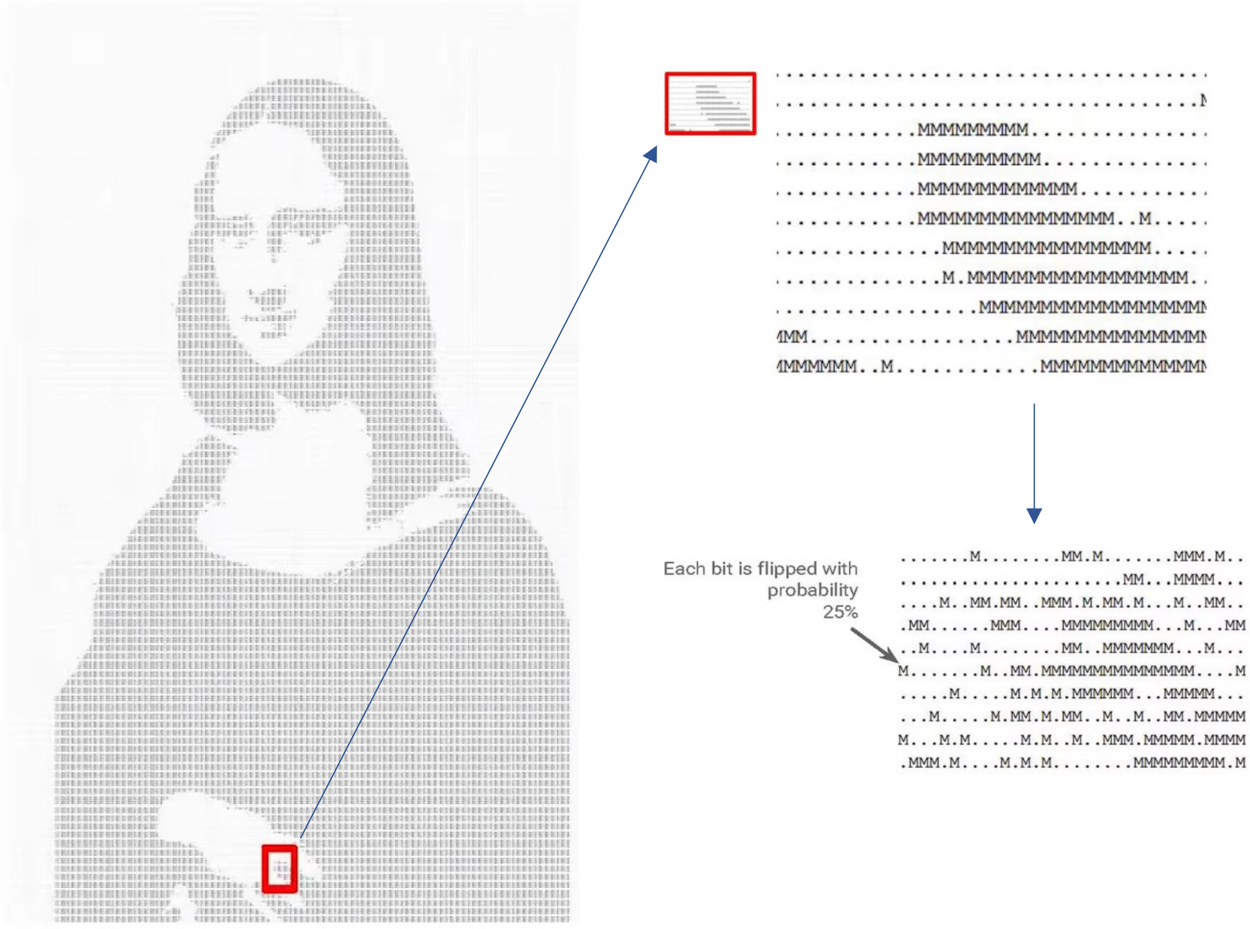
[EPK14] RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. CCS 2014.

Metaphor for RAPPOR*



* Utilizing Large-Scale Randomized Response at Google: RAPPOR and its lessons by Ananth Raghunathan: https://www.youtube.com/watch?v=tuOBz5AzivM&ab_channel=RutgersUniversity.

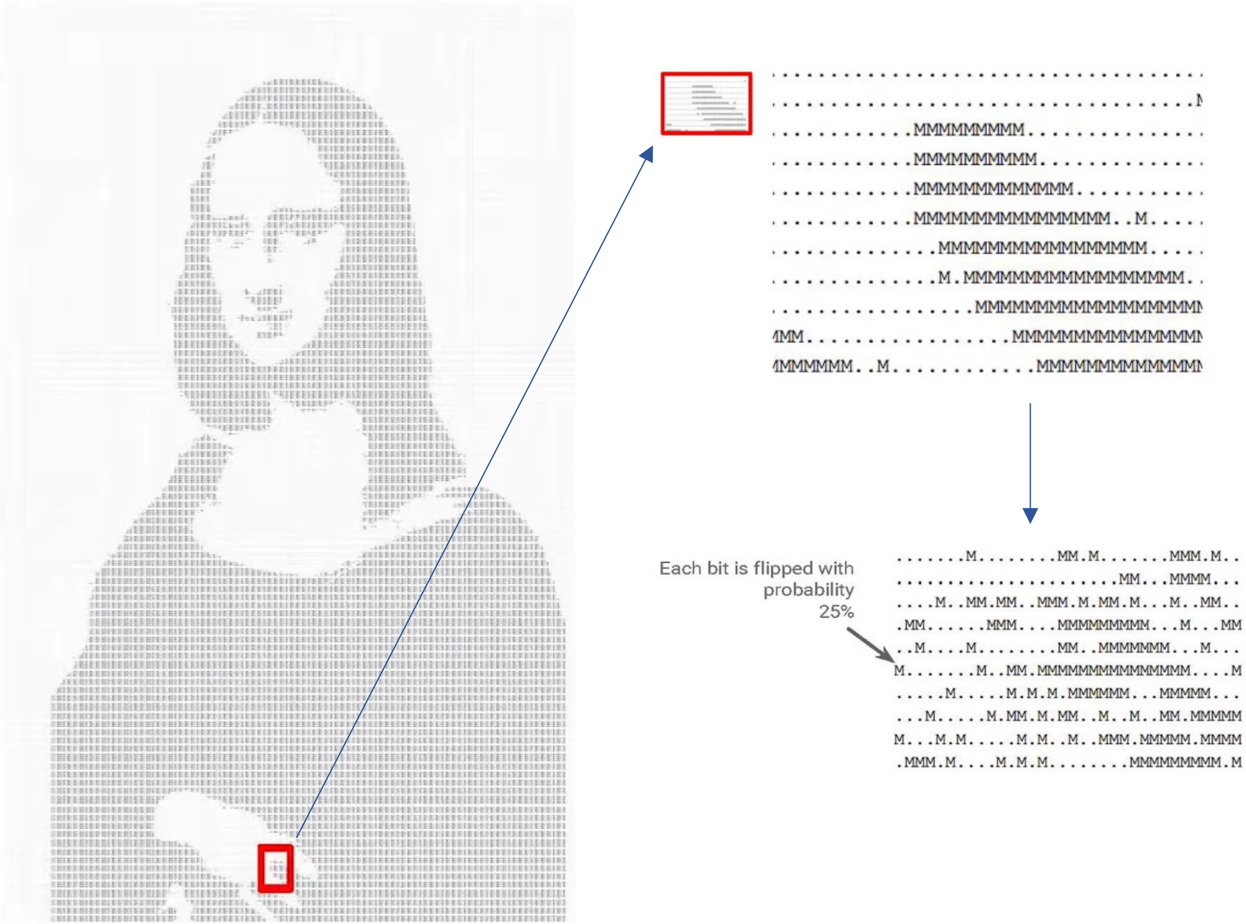
Metaphor for RAPPOR*

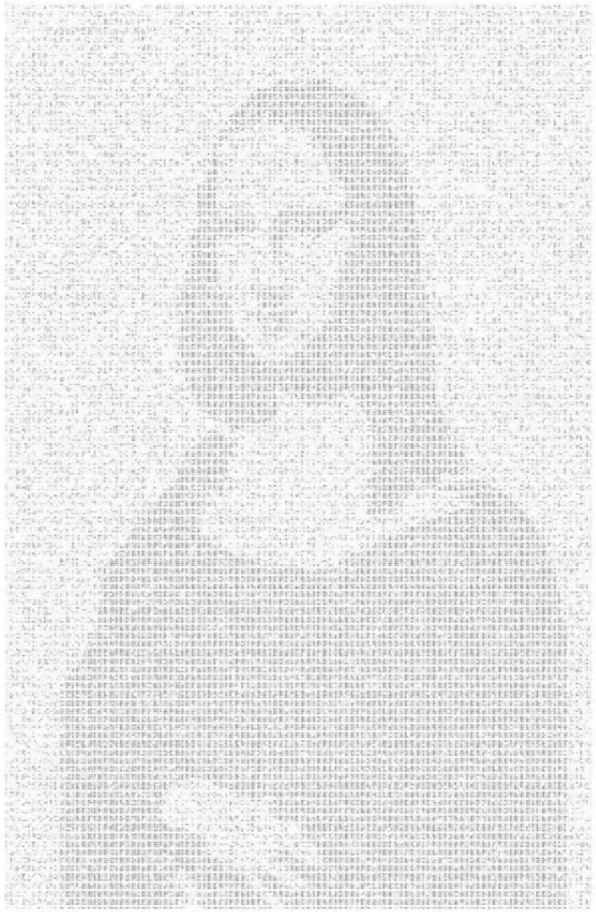


* Utilizing Large-Scale Randomized Response at Google: RAPPOR and its lessons by Ananth Raghunathan: https://www.youtube.com/watch?v=tuOBz5AzivM&ab_channel=RutgersUniversity.

Metaphor for RAPPOR*

Big picture remains!





* Utilizing Large-Scale Randomized Response at Google: RAPPOR and its lessons by Ananth Raghunathan: https://www.youtube.com/watch?v=tuOBz5AzivM&ab_channel=RutgersUniversity.

Google's RAPPOR Solution for Chrome [EPK14]

Basic RAPPOR (deterministic UE) → utility-oriented version of RAPPOR.

 # User-side

- Encode the value v into a bit vector $\vec{\boldsymbol{v}} = \vec{0}, \vec{\boldsymbol{v}}[v] = 1$.
- Perturb each bit independently with SUE:
    - **Memoize and reuse for each time the value v repeats.** → **Permanent RR (PRR)**

[EPK14] RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. CCS 2014.

Google's RAPPOR Solution for Chrome [EPK14]

Basic RAPPOR (deterministic UE) → utility-oriented version of RAPPOR.

 # User-side

- Encode the value v into a bit vector $\vec{\boldsymbol{v}} = \vec{0}$, $\vec{\boldsymbol{v}}[v] = 1$.
- Perturb each bit independently with SUE:
    - **Memoize and reuse for each time the value v repeats.** → **Permanent RR (PRR)**
- **For each time $t \in [\tau]$, apply SUE (again) to the memoized value.** → **Instantaneous RR (IRR)**

 # Server-side (for each time $t \in [\tau]$)

- $\mathrm{c}(v)$ → number of times the bit corresponding to $v \in V$ has been reported.
- Unbiased estimator: $\hat{f}(v) = \frac{\mathrm{c}(v) - nq_1(p_2 - q_2) - nq_2}{n(p_1 - q_1)(p_2 - q_2)}$.

[EPK14] RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. CCS 2014.

Google's RAPPOR Solution for Chrome [EPK14]

Pros:

- RAPPOR upper bounds the privacy loss (*i.e.*, PRR).
- The IRR step also prevents tracking (when excluding users' IDs).
- Original RAPPOR makes use of Bloom filters (generic), and UE improves utility.

[EPK14] RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. CCS 2014.

Google's RAPPOR Solution for Chrome [EPK14]

Pros:

- RAPPOR upper bounds the privacy loss (*i.e.*, PRR).
- The IRR step also prevents tracking (when excluding users' IDs).
- Original RAPPOR makes use of Bloom filters (generic), and UE improves utility.

Limitations:

- Practical deployment → needs ~10K reports to identify a value with confidence.
- Does not support even small data changes of the user's actual data:
    - Need to run RAPPOR for each value $v \in V$.
    - Worst-case longitudinal privacy loss linear on domain size k.

$$\forall_{u \in U}: \quad \check{\epsilon}_{\infty}^{(u)} \leq k\epsilon_{\infty}$$

[EPK14] RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. CCS 2014.

Microsoft Telemetry Data collection [DKY17]

Microsoft collect data on app usage:

- How much time was spent on a particular app today?
- Allows finding patterns over time…

[DKY17] Collecting telemetry data privately. NeurIPS 2017.

Microsoft Telemetry Data collection [DKY17]

Microsoft collect data on app usage:

- How much time was spent on a particular app today?
- Allows finding patterns over time…

Makes use of multiple subroutines:

- 1BitMean to collect numeric data for mean estimation.
- dBitFlipPM to collect (sparse) histogram data.
- Memoization and output perturbation to allow repeated data collection.

[DKY17] Collecting telemetry data privately. NeurIPS 2017.

Microsoft's dBitFlipPM Solution for Windows [DKY17]

Permanent Memoization → PRR only

dBitFlipPM → a memoization-based solution as alternative to RAPPOR.

 # User-side

- Bucketize domain k to b buckets (*e.g.*, with equal width): $V \rightarrow [b]$.
- User samples d buckets without replacement and perturb them with SUE:
    - **Memoize and reuse for all values falling into the same bucket.**

[DKY17] Collecting telemetry data privately. NeurIPS 2017.

Microsoft's dBitFlipPM Solution for Windows [DKY17]

Permanent Memoization → PRR only

dBitFlipPM → a memoization-based solution as alternative to RAPPOR.

 # User-side

- Bucketize domain k to b buckets (*e.g.*, with equal width): $V \rightarrow [b]$.
- User samples d buckets without replacement and perturb them with SUE:
  - **Memoize and reuse for all values falling into the same bucket.**

 # Server-side

- Aggregator counts and unbiases the noisy reports: $\hat{f}(v) = \frac{b}{nd}\frac{(c(v)-nq)}{(p-q)}$.
- Error proportional to $\sqrt{(b/d)}$: trades off error and cost.

[DKY17] Collecting telemetry data privately. NeurIPS 2017.

Microsoft's dBitFlipPM Solution for Windows [DKY17]

Permanent Memoization → PRR only

$V = \{1, 2, \ldots, 100\}, k = 100$

$[b] = \{1, \ldots, 10\}, b = 10$

Run dBitFlipPM for each bucket and permanently memoize them.

$\vec{v} = [1,1,1,9,2,1,1,1,8,9]$ ⟶ Same bucket 1

[DKY17] Collecting telemetry data privately. NeurIPS 2017.

Microsoft's dBitFlipPM Solution for Windows [DKY17]

Permanent Memoization → PRR only

Pros:

- Less computation and communication costs ($d \leq b$ bits).
- Creates uncertainty on values falling into the same bucket.

[DKY17] Collecting telemetry data privately. NeurIPS 2017.

Microsoft's dBitFlipPM Solution for Windows [DKY17]

Permanent Memoization → PRR only

Pros:

- Less computation and communication costs ($d \leq b$ bits).
- Creates uncertainty on values falling into the same bucket.

Limitations:

- Information loss due to $V \rightarrow [b]$ and sampling only d out of b bits.
- Supports only small data changes of the user's actual data:
  - Possibility of (real-time) detection of large data change.
  - Need to run dBitFlipPM for each bucket in $[b]$.
  - Worst-case longitudinal privacy loss linear on new domain size $b \leq k$.

$$\forall_{u \in U}: \quad \check{\epsilon}_{\infty}^{(u)} \leq \min(d+1, b)\, \epsilon_{\infty}$$

[DKY17] Collecting telemetry data privately. NeurIPS 2017.

LOLOHA: LOngitudinal LOcal Hashing [APGP23]

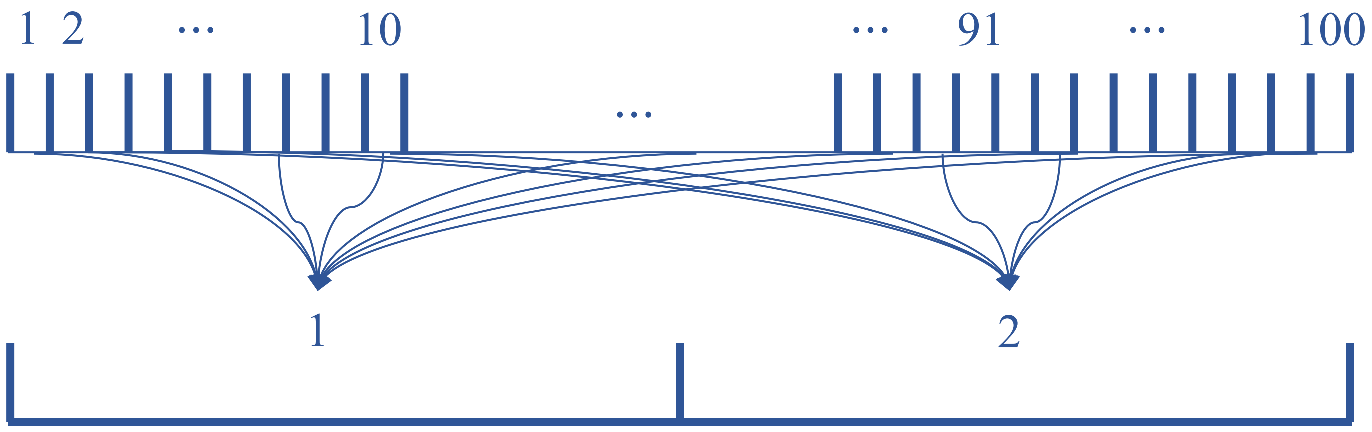
Our proposal → join forces of RAPPOR + dBitFlipPM:

- Double randomization to minimize data change detection → PRR and IRR.
- Several values are mapped to the same randomized value → Local hashing.

Given a (universal) family of hash functions $\mathcal{H}$: $\forall v_1, v_2 \in V, v_1 \neq v_2 : \Pr_{H \in \mathcal{H}}[H(v_1) = H(v_2)] \leq \frac{1}{g}$

$V = \{1,2,\dots,100\}, k = 100$

$[g] = \{1,2\}, g = 2$



[APGP23] Frequency estimation of evolving data under local differential privacy. EDBT 2023.

LOLOHA: LOngitudinal LOcal Hashing [APGP23]

 # User-side

- Each user uses a (unique) random hash function H that maps $V \rightarrow \{0,1,\ldots,g\}$.
- The user then perturbs the hashed ("encoded") value with GRR:
    - **Memoize and reuse for all values hashing into the same value in $[g]$.** → **PRR**
- **For each time $t \in [\tau]$, apply GRR (again) to the memoized value.** → **IRR**
- BiLOLOHA ($g = 2$) and OLOLOHA (optimal g, large equation).

[APGP23] Frequency estimation of evolving data under local differential privacy. EDBT 2023.

LOLOHA: LOngitudinal LOcal Hashing [APGP23]

 # User-side

- Each user uses a (unique) random hash function H that maps $V \rightarrow \{0,1,\ldots,g\}$.
- The user then perturbs the hashed ("encoded") value with GRR:
  - **Memoize and reuse for all values hashing into the same value in $[g]$.** → **PRR**
- **For each time $t \in [\tau]$, apply GRR (again) to the memoized value.** → **IRR**
- BiLOLOHA ($g = 2$) and OLOLOHA (optimal g, large equation).

 # Server-side

- $c(v) \rightarrow |\{u \in U | H^u(v) = v^u\}|$, $q_1' = \frac{1}{g}$.
- Unbiased estimator: $\hat{f}(v) = \frac{c(v) - nq_1'(p_2 - q_2) - nq_2}{n(p_1 - q_1')(p_2 - q_2)}$.

[APGP23] Frequency estimation of evolving data under local differential privacy. EDBT 2023.

LOLOHA: LOngitudinal LOcal Hashing [APGP23]

Pros:

- Creates uncertainty on values hashed to the same value in [g].
- Smallest communication cost than all competitors.
- Allows to balance privacy ($g = 2$) and utility (optimal g).
- Worst-case longitudinal privacy loss linear on $g \ll k$ only.

$$\forall_{u \in U}: \quad \check{\epsilon}_{\infty}^{(u)} \leq g\epsilon_{\infty}$$

Limitations:

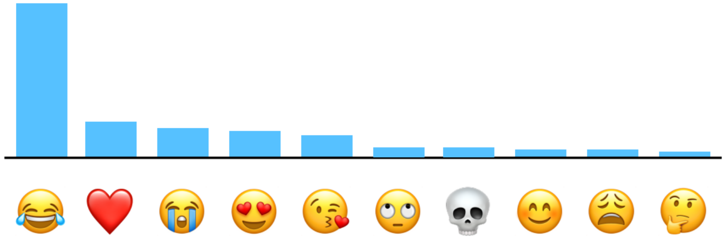
- The unique random hash function can be used to track user. However, LDP assumes to know users' identifier but not their private data.

[APGP23] Frequency estimation of evolving data under local differential privacy. EDBT 2023.

Other LDP Deployments [DPT17, SKSGS24]

Apple: Common emoji & out-of-vocabulary word discovery:

- Sketches and Transforms.
- Count Mean Sketch (CMS) + RR.

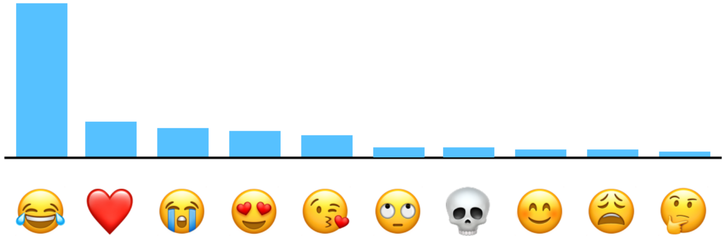


[DPT17] Learning with Privacy at Scale. Apple's white paper 2017.
[SKSGS24] Private federated discovery of out-of-vocabulary words for Gboard. arXiv 2024.

Other LDP Deployments [DPT17, SKSGS24]

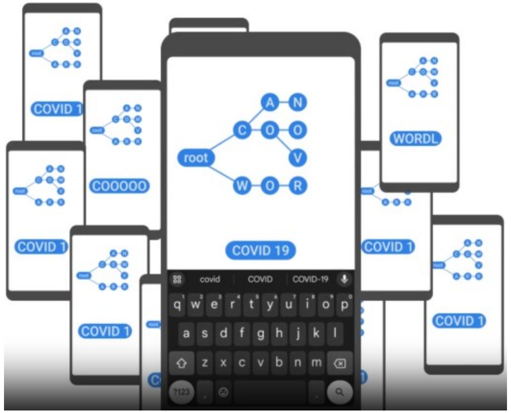
Apple: Common emoji & out-of-vocabulary word discovery:

- Sketches and Transforms.
- Count Mean Sketch (CMS) + RR.



Gboard: Out-of-vocabulary word discovery:

- Prefix Tree and Sampling.
- SS protocol + Sampling.



[DPT17] Learning with Privacy at Scale. Apple's white paper 2017.
[SKSGS24] Private federated discovery of out-of-vocabulary words for Gboard. arXiv 2024.

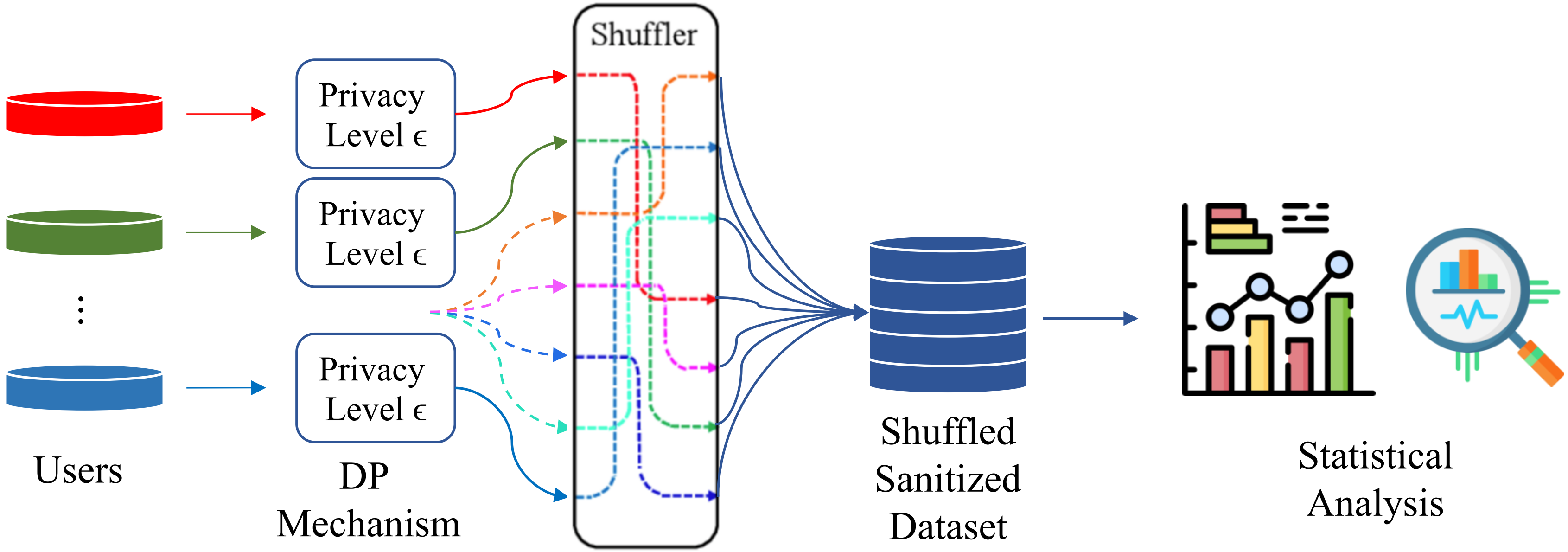
Outline

- Module 1 (Introduction):
  - Review of DP and preliminaries
  - LDP introduction
  - State-of-the-art deployments of LDP

- **Module 2 (Current research directions):**
  - Privacy attacks on LDP protocols
  - Security attacks on LDP protocols
  - Final remarks & open problems

Shuffle DP: LDP + Anonymity [CSUZZ19, EFMRTT19]

- Remove all metadata that can link users to their (perturbed) reported values.
- Amplification by shuffling → from ϵ-LDP to (ϵ', δ)-DP where $\epsilon' > \epsilon$.
- Challenge: prove tighter bounds and design optimal Shuffle DP mechanisms.



[CSUZZ19] Distributed differential privacy via shuffling. EUROCRYPT 2019.
[EFMRTT19] Amplification by shuffling: From local to central differential privacy via anonymity. SODA 2019.

LDP Tasks Based on Frequency Estimation

Frequency Estimation: A Building Block for More Complex Tasks

Heavy hitter estimation [CCDFHJMT24]**:**

- Goal: Find the t most frequent values from a large V.
- V is large (when V is small, LDP frequency estimation suffices).

[CCDFHJMT24] Differentially private heavy hitter detection using federated analytics. SaTML 2024.

Frequency Estimation: A Building Block for More Complex Tasks

Heavy hitter estimation [CCDFHJMT24]**:**

- Goal: Find the t most frequent values from a large V.
- V is large (when V is small, LDP frequency estimation suffices).

Marginal estimation [CKS18]**:**

- User has d bits of data and the server want (all) marginals over m attributes.
- Each marginal is a frequency distribution → could apply RR… (optimal?)

	Gender	Obese	…	Smoke	Disease
Alice	1	0	…	1	0
Bob	0	1	…	1	1
Carl	0	0	…	0	0

→

Gender/Obese	0	1
0	0.28	0.22
1	0.29	0.21

Disease/Smoke	0	1
0	0.55	0.15
1	0.10	0.20

[CCDFHJMT24] Differentially private heavy hitter detection using federated analytics. SaTML 2024.
[CKS18] Marginal release under local differential privacy. SIGMOD 2018.

Frequency Estimation: A Building Block for More Complex Tasks

Frequent itemset mining [LGGWY22]**:**

- Each user has a set of values.
- The goal is to find the frequent singletons and itemsets.

{a, c, e} {b, e} {a, b, e} {a, d, e} {a, b, c, d, e, f} → Top-3 singletons: e(5), a(4), b(3)
Top-3 itemsets: {e}(5), {a}(4), {a, e}(4)

[LGGWY22] Frequent itemset mining with local differential privacy. In CIKM 2022.

Frequency Estimation: A Building Block for More Complex Tasks

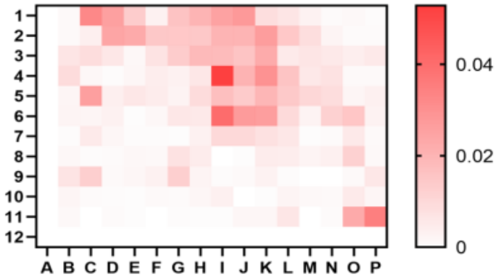
Frequent itemset mining [LGGWY22]**:**

- Each user has a set of values.
- The goal is to find the frequent singletons and itemsets.

{a, c, e} {b, e} {a, b, e} {a, d, e} {a, b, c, d, e, f} → Top-3 singletons: e(5), a(4), b(3)
Top-3 itemsets: {e}(5), {a}(4), {a, e}(4)

Spatial data (*e.g.*, crowd density estimation) [TG24]**:**

- Impose a hierarchical grid structure and count.
- If small grid → LDP frequency estimation suffices.
- Identify heavy regions → a heavy hitter problem!



[LGGWY22] Frequent itemset mining with local differential privacy. In CIKM 2022.
[TG24] Answering Spatial Density Queries Under Local Differential Privacy. IEEE IoT 2024.

Frequency Estimation: A Building Block for More Complex Tasks

Frequency monitoring (*i.e.*, longitudinal data):

- Current deployment → weak longitudinal guarantees:
  - Google & Microsoft → Memoization:
    - Small or no data change.
    - Violates DP guarantees.
  - Apple → independent fresh noise [TKBWW17].

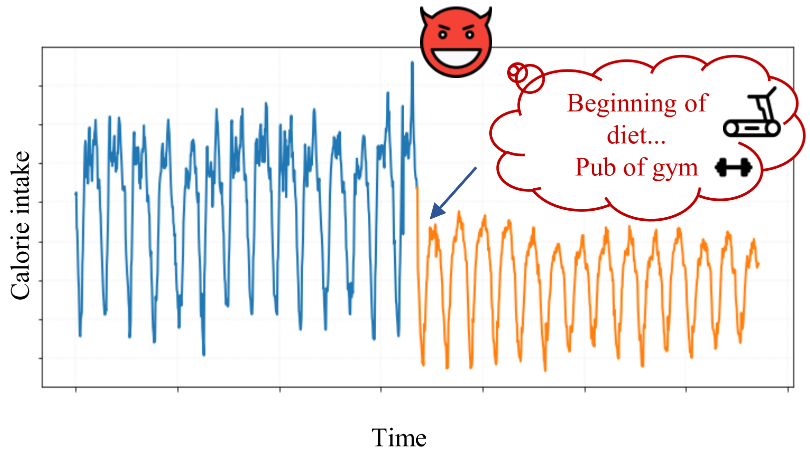
[TKBWW17] Privacy loss in apple's implementation of differential privacy on macos 10.12. Arxiv 2017.

Inria

Frequency Estimation: A Building Block for More Complex Tasks

Frequency monitoring (*i.e.*, longitudinal data):

- Current deployment → weak longitudinal guarantees:
  - Google & Microsoft → Memoization:
    - Small or no data change.
    - Violates DP guarantees.
  - Apple → independent fresh noise [TKBWW17].

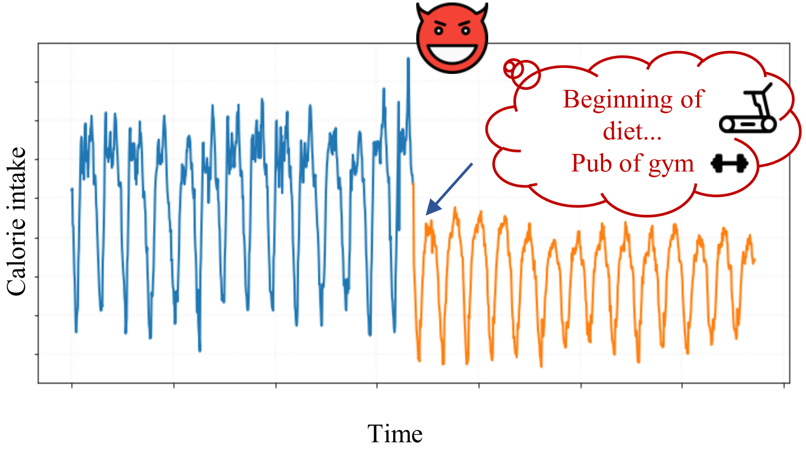


[TKBWW17] Privacy loss in apple's implementation of differential privacy on macos 10.12. Arxiv 2017.

Frequency Estimation: A Building Block for More Complex Tasks

Frequency monitoring (*i.e.*, longitudinal data):

- Current deployment → weak longitudinal guarantees:
  - Google & Microsoft → Memoization:
    - Small or no data change.
    - Violates DP guarantees.
  - Apple → independent fresh noise [TKBWW17].



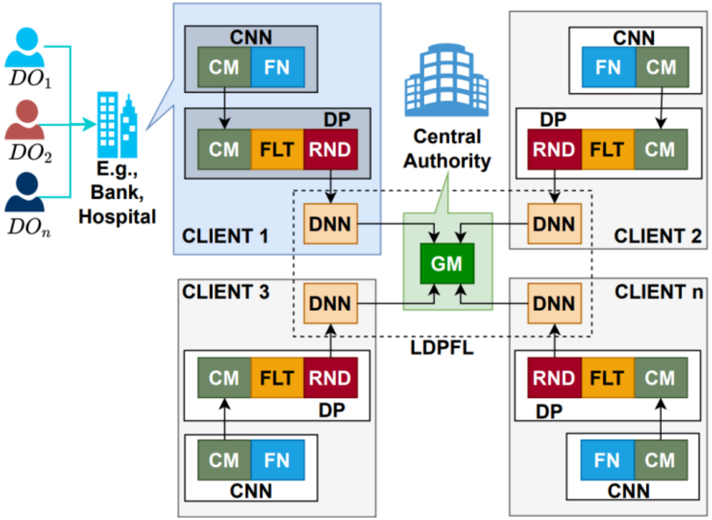
- Data change-based solutions [JRUW18, EFMRTT19]:
  - Consider the infrequent data changes on the user side.
  - Privacy loss & accuracy proportional to number of changes.
  - Mainly designed for Boolean data.
  - Restriction on the number of data changes & number of data collections.

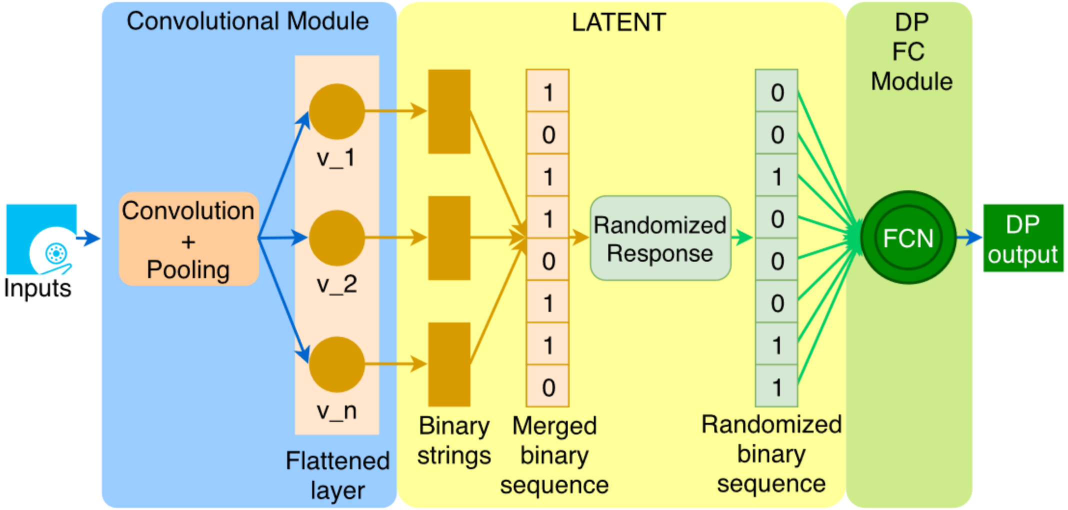
[TKBWW17] Privacy loss in apple's implementation of differential privacy on macos 10.12. Arxiv 2017.
[JRUW18] Local differential privacy for evolving data. NeurIPS 2018.
[EFMRTT19] Amplification by shuffling: From local to central differential privacy via anonymity. SODA 2019.

Frequency Estimation: A Building Block for More Complex Tasks

Learning tasks [ABKLCA20, YAC20, CLCNGBK22]**:**

- The goal is to learn a model for prediction purposes (*e.g.*, binary classification).
- Train machine (or federated) learning models using LDP-based statistics or NN layer.





[ABKLCA20] Local Differential Privacy for Deep Learning. IEEE IoT 2020.
[YAC20] Naive Bayes classification under local differential privacy. DSAA 2020.
[CLCNGBK22] Local differential privacy for federated learning. ESORICS 2022.

Open-Source (Python) Implementations

RAPPOR [CMM21]:

- https://github.com/google/rappor.
- Frequency estimation

SCAN ME

pure-ldp [CMM21]:

- https://pypi.org/project/pure-ldp/.
- Frequency estimation:
  - Unidimensional data.
- Heavy hitter estimation.

SCAN ME

multi-freq-ldpy [ACGPZ22]:

- https://pypi.org/project/multi-freq-ldpy/.
- Frequency estimation:
  - Unidimensional data.
  - Multidimensional data.
  - Longitudinal data.

SCAN ME

[EPK14] RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. CCS 2014.
[CMM21] Frequency estimation under local differential privacy. VLDB 2021.
[ACGPZ22] Multi-Freq-LDPy: Multiple Frequency Estimation Under LDP in Python. ESORICS 2022.

Outline

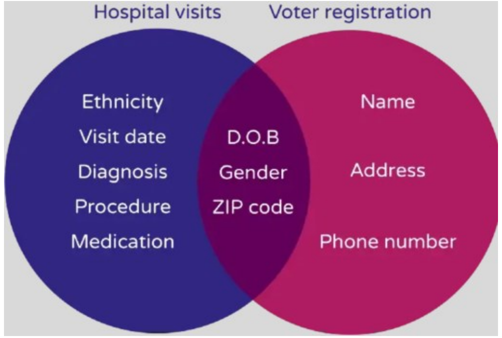
- Module 1 (Introduction):
  - Review of DP and preliminaries
  - LDP introduction
  - State-of-the-art deployments of LDP

- **Module 2 (Current research directions):**
  - **Privacy attacks on LDP protocols**
  - Security attacks on LDP protocols
  - Final remarks & open problems

Exploiting the "Good Side" of Privacy Attacks*

Privacy attacks play an essential role in privacy research!

Re-identification



Failures of pseudonymization
& invention of k-anonymity

Homogeneity

	Non-Sensitive			Sensitive
	Zip Code	Age	Nationality	Condition
1	130**	< 30	$*$	Heart Disease
2	130**	< 30	$*$	Heart Disease
3	130**	< 30	$*$	Viral Infection
4	130**	< 30	$*$	Viral Infection
5	1485*	≥ 40	$*$	Cancer
6	1485*	≥ 40	$*$	Heart Disease
7	1485*	≥ 40	$*$	Viral Infection
8	1485*	≥ 40	$*$	Viral Infection
9	130**	$3*$	$*$	Cancer
10	130**	$3*$	$*$	Cancer
11	130**	$3*$	$*$	Cancer
12	130**	$3*$	$*$	Cancer

Failures of k-anonymity
& invention of l-diversity

Re-construction

TABLE 4: A SINGLE SATISFYING ASSIGNMENT

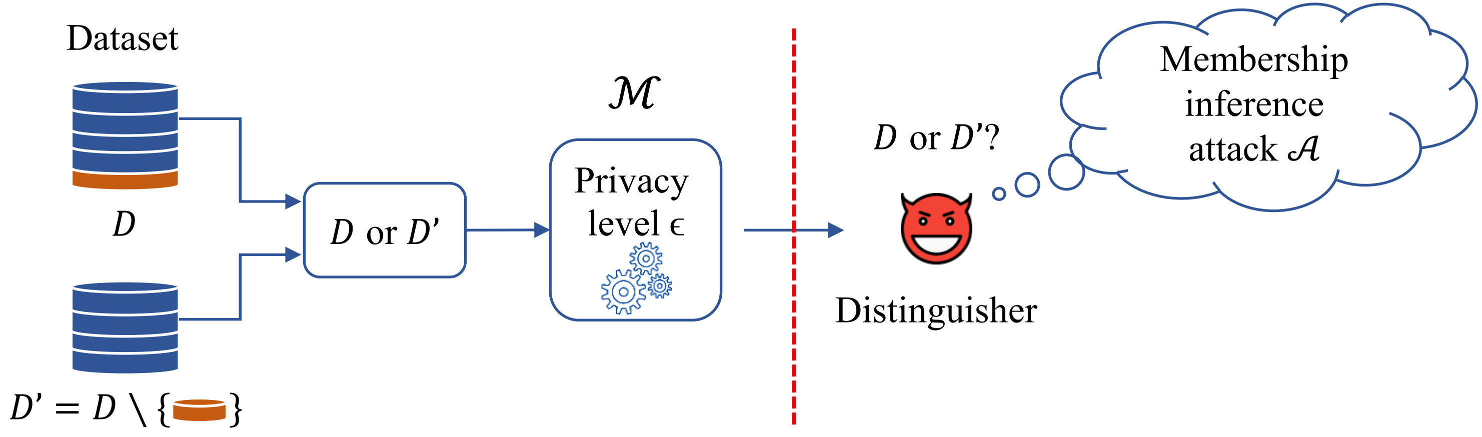
AGE	SEX	RACE	MARITAL STATUS		SOLUTION #1
8	F	B	S		8FBS
18	M	W	S		18MWS
24	F	W	S	=	24FWS
30	M	W	M		30MWM
36	F	B	M		36FBM
66	F	B	M		66FBM
84	M	B	M		84MBM

Inspired invention and
adoption of differential privacy

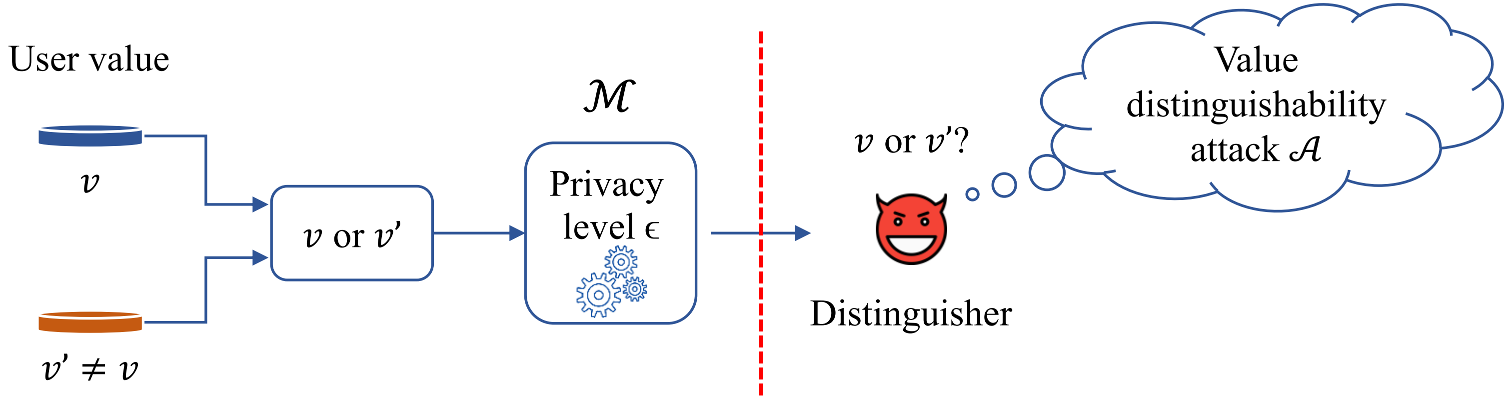
* Slide Inspired by: "[U23] Auditing Differentially Private Machine Learning. PPAI 2023 (Invited Talk)".

Adversarial Privacy Game of Central and Local DP

Central DP



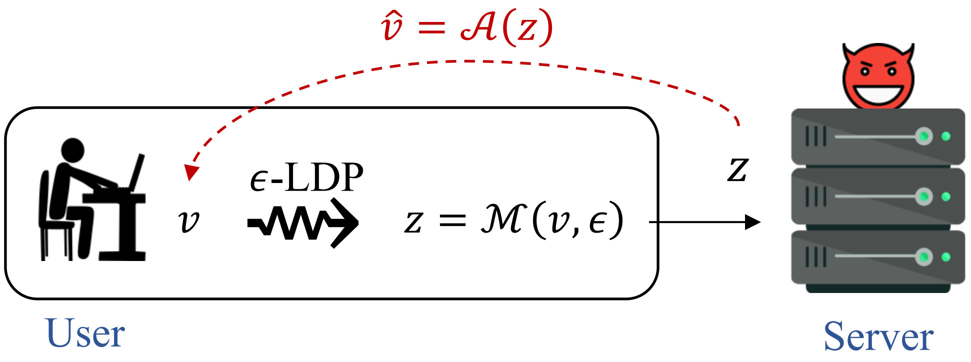
Local DP



Privacy Threats to LDP Protocols [GLCTW22, AGCP23]

Value distinguishability attack:

- Users obfuscate v with an ϵ-LDP protocol $\mathcal{M}$.
- Bayesian adversary predicts $\hat{v}$ given $z = \mathcal{M}(v)$, *i.e.*, $\hat{v} = \underset{v \in V}{\text{argmax}} \Pr[v \mid z]$.
- Metric: Adversarial Success Rate ($\text{ASR} = \Pr[v = \hat{v}]$).

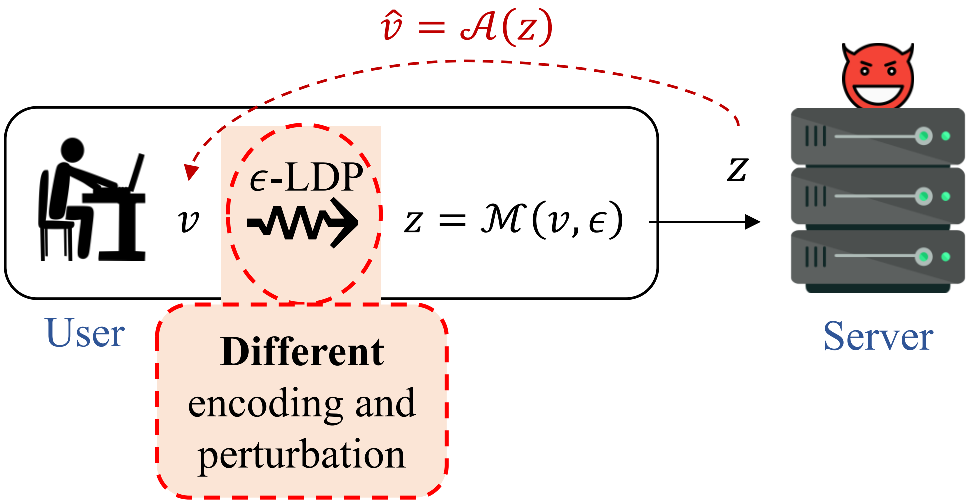


[GLCTW22] An Adversarial Approach to Protocol Analysis and Selection in LDP. TIFS 2022.
[AGCP23] On the Risks of Collecting Multidimensional Data Under LDP. VLDB 2023.

Privacy Threats to LDP Protocols [GLCTW22, AGCP23]

Value distinguishability attack:

- Users obfuscate v with an ϵ-LDP protocol $\mathcal{M}$.
- Bayesian adversary predicts $\hat{v}$ given $z = \mathcal{M}(v)$, *i.e.*, $\hat{v} = \underset{v \in V}{\operatorname{argmax}} \Pr[v \mid z]$.
- Metric: Adversarial Success Rate ($\text{ASR} = \Pr[v = \hat{v}]$).



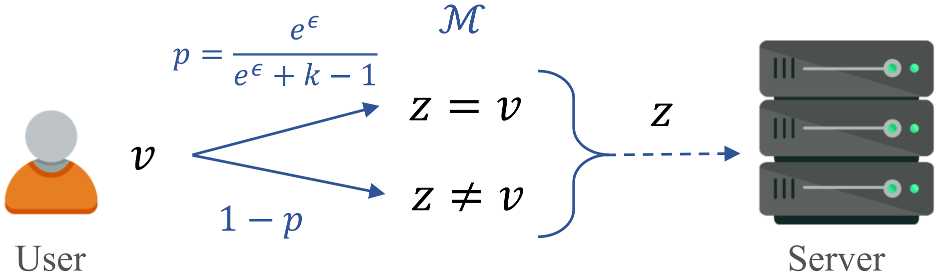
Designed attacks $\mathcal{A}$ tailored to the LDP protocol

[GLCTW22] An Adversarial Approach to Protocol Analysis and Selection in LDP. TIFS 2022.
[AGCP23] On the Risks of Collecting Multidimensional Data Under LDP. VLDB 2023.

Generalized Randomized Response (GRR) [W65, KBR16]

Bayesian adversary $\mathcal{A}_{\text{GRR}}$:

- Optimal prediction strategy is to assume user is honest.
- For any value $v \in V$, $\Pr[z = v] > \Pr[z = v']$ for all $v' \in V \setminus \{v\}$.
- $\mathcal{A}_{\text{GRR}}$: $\hat{v} = z$.

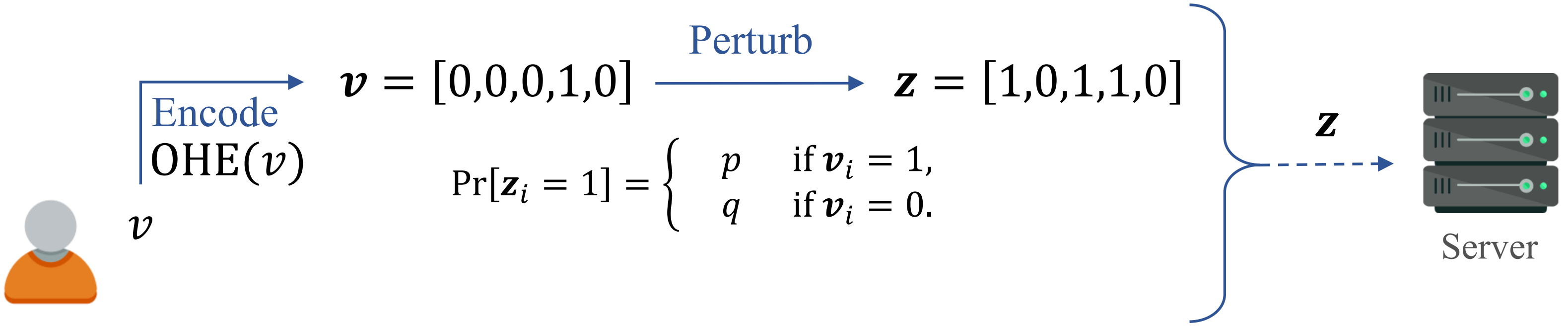


[W65] Randomized response: A survey technique for eliminating evasive answer bias. JASA 1965.
[KBR16] Discrete distribution estimation under local privacy. ICML 2016.

Unary Encoding (UE) Protocols [EPK14, WBLJ17]

Bayesian adversary $\mathcal{A}_{\text{UE}}$:

- Optimal prediction strategy is to pick among indexes set to 1.
- Construct: $\mathbb{I} = \{v \mid \boldsymbol{z}_v = 1\}$.
- $\mathcal{A}^0_{\text{UE}}$: $\hat{v} = \text{Uniform}([k])$, if $\mathbb{I} = \{\emptyset\}$.
- $\mathcal{A}^1_{\text{UE}}$: $\hat{v} = \text{Uniform}(\mathbb{I})$, otherwise.

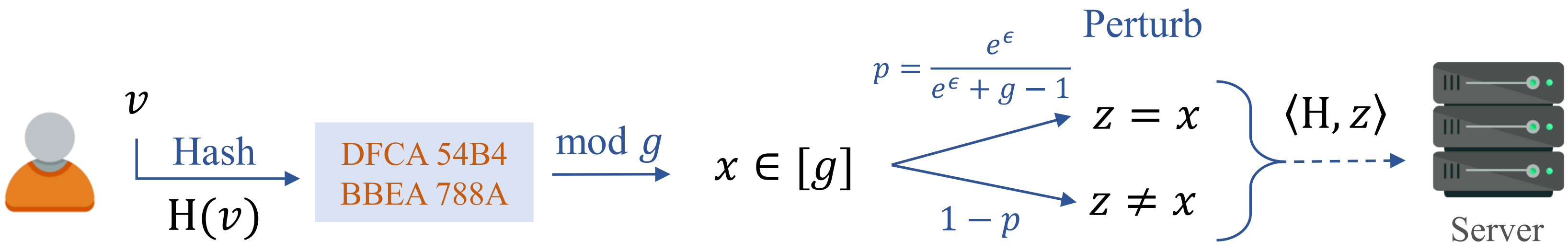


[EPK14] RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. CCS 2014.
[WBLJ17] Locally Differentially Private Protocols for Frequency Estimation. USENIX Security 2017.

Local Hashing (LH) Encoding Protocols [WBLJ17, BS15]

Bayesian adversary $\mathcal{A}_{\mathrm{LH}}$:

- Optimal prediction strategy is a random choice from subset of values that hash to z.
- Construct: $\mathbb{I} = \{v \mid \mathrm{H}(v) = z\}$.
- $\mathcal{A}^0_{\mathrm{LH}}$: $\hat{v} = \mathrm{Uniform}([k])$, if $\mathbb{I} = \{\emptyset\}$.
- $\mathcal{A}^1_{\mathrm{LH}}$: $\hat{v} = \mathrm{Uniform}(\mathbb{I})$, otherwise.

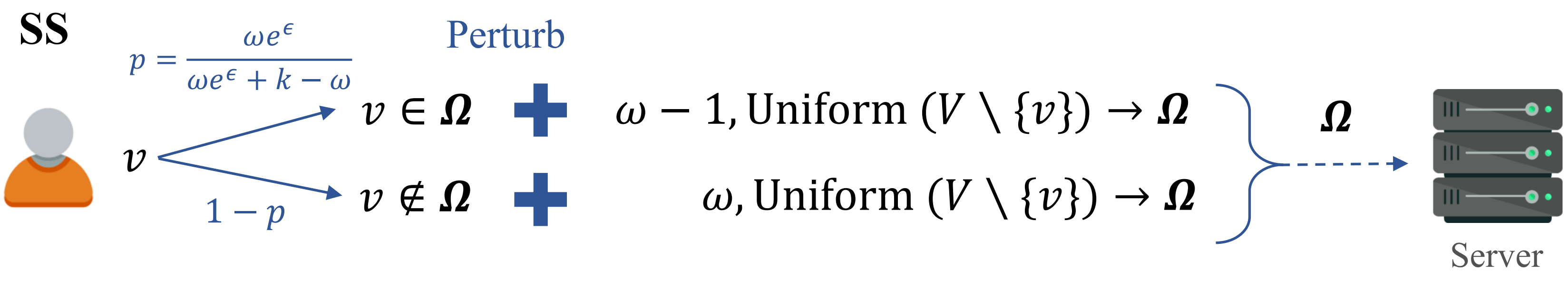


[WBLJ17] Locally Differentially Private Protocols for Frequency Estimation. USENIX Security 2017.
[BS15] Local, Private, Efficient Protocols for Succinct Histograms. STOC 2015.

Subset Selection (SS) [YB18]

Bayesian adversary $\mathcal{A}_{SS}$:

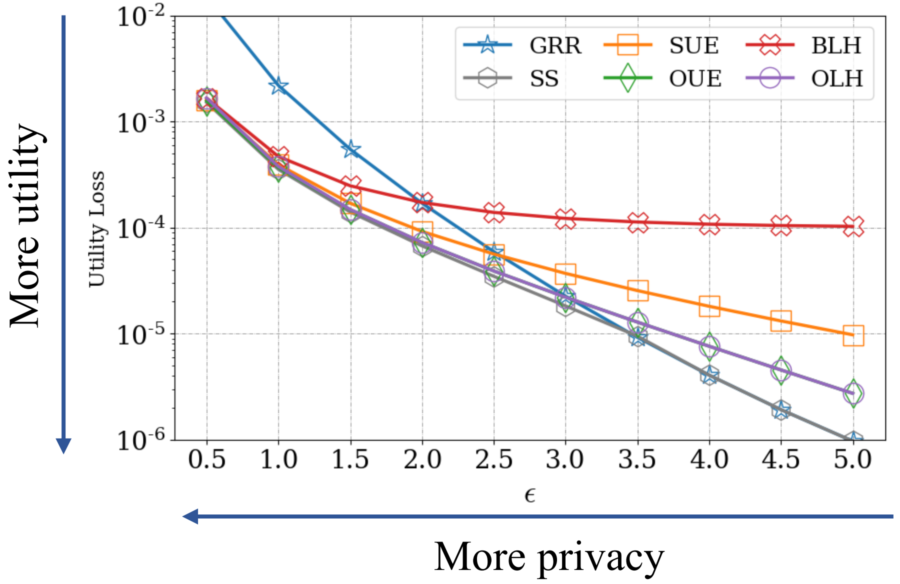
- Optimal prediction strategy is a random choice from the reported subset $\boldsymbol{\Omega}$.
- For any value $v \in V$, $\Pr[v \in \boldsymbol{\Omega}] > \Pr[v' \in \boldsymbol{\Omega}]$ for all $v' \in V \setminus \{v\}$.
- $\mathcal{A}_{SS}$: $\hat{v} = \text{Uniform}(\boldsymbol{\Omega})$.



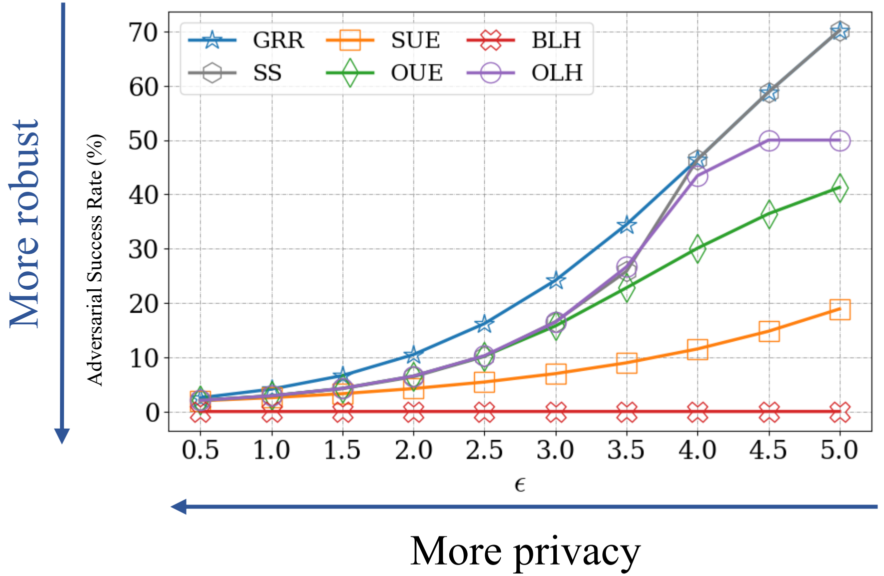
Privacy-Utility-Robustness Trade-Off [GLCTW22, AGCP23]

ϵ is not the unique parameter to measure privacy!

Usual approach: Privacy-Utility Trade-off



This approach: Privacy-Robustness Trade-off

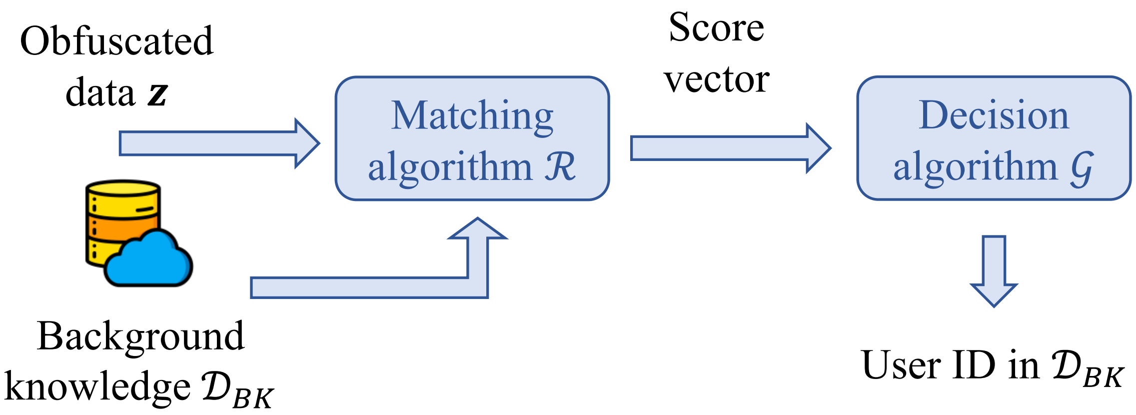


[GLCTW22] An Adversarial Approach to Protocol Analysis and Selection in LDP. TIFS 2022.
[AGCP23] On the Risks of Collecting Multidimensional Data Under LDP. VLDB 2023.

Other Privacy Threats to LDP Protocols

Re-identification risks [MT21, AGCP23]:

- Sequential data (*e.g.*, location traces) allows linking obfuscated data to users.
- Multiple collections lead to profiling and uniqueness through quasi-identifiers.

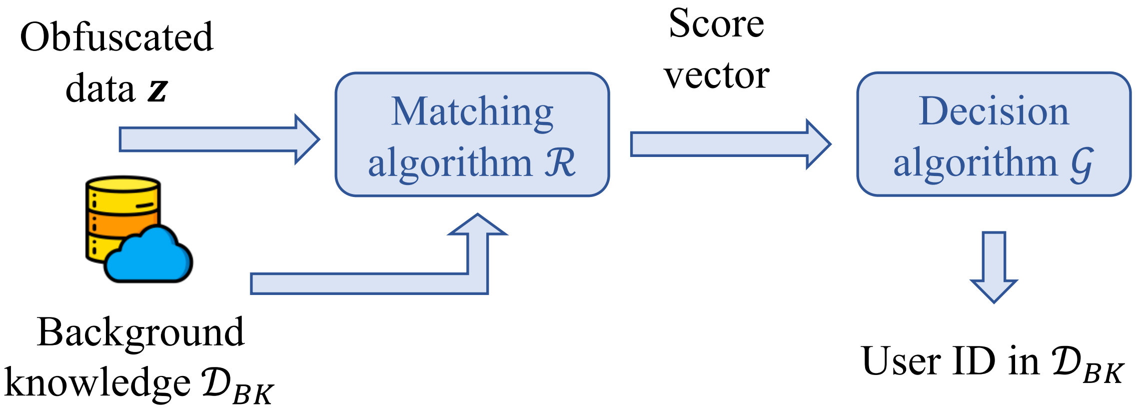


[MT21] Toward evaluating re-identification risks in the local privacy model. TDP 2020.
[AGCP23] On the Risks of Collecting Multidimensional Data Under LDP. VLDB 2023.

Other Privacy Threats to LDP Protocols

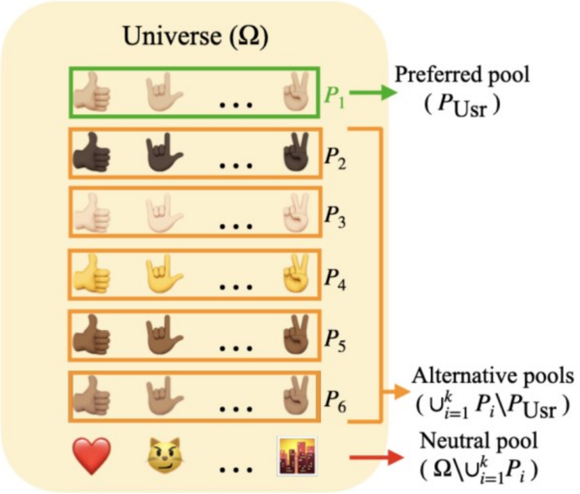
Re-identification risks [MT21, AGCP23]:

- Sequential data (*e.g.*, location traces) allows linking obfuscated data to users.
- Multiple collections lead to profiling and uniqueness through quasi-identifiers.



Pool inference attacks [GHAM22]:

- Multiple collections lead to profiling and pool inference.



[MT21] Toward evaluating re-identification risks in the local privacy model. TDP 2020.
[AGCP23] On the Risks of Collecting Multidimensional Data Under LDP. VLDB 2023.
[GHAM22] Pool Inference Attacks on Local Differential Privacy. USENIX Security 2022.

Using Privacy Attacks to Audit LDP

Statistically Measuring LDP [AG24]

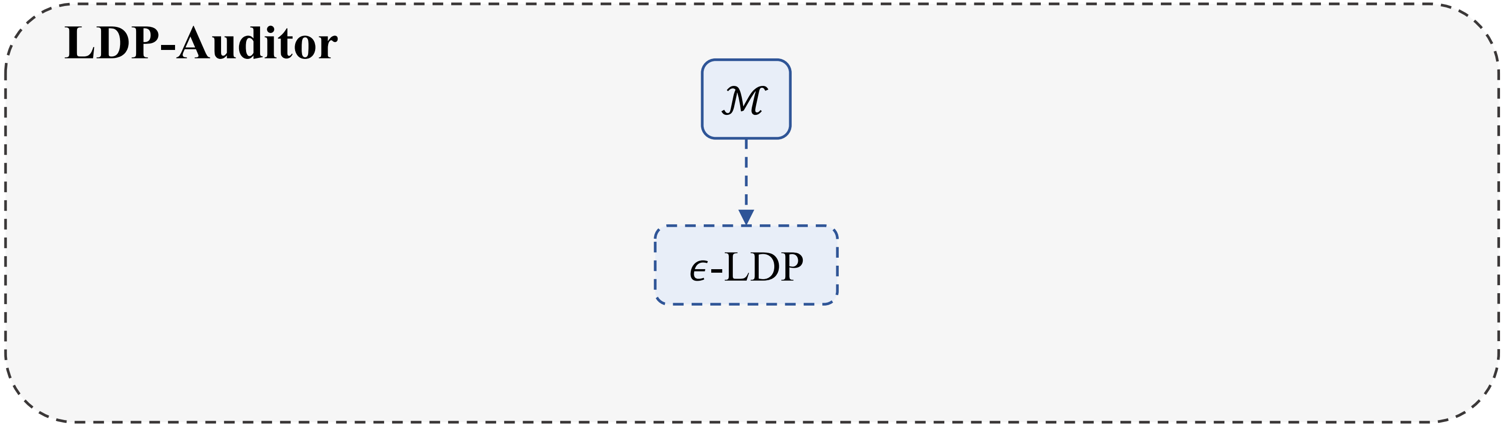
LDP as hypothesis testing → Attacker receives an output drawn either from $\mathcal{M}(v)$ or $\mathcal{M}(v')$:

- For every attacker: $\underbrace{\Pr[\mathcal{A}(\mathcal{M}(v)) = v]}_{\text{True Positive (TPR)}} \leq e^{\epsilon} \cdot \underbrace{\Pr[\mathcal{A}(\mathcal{M}(v')) = v]}_{\text{False Positive (FPR)}}$

Statistically Measuring LDP [AG24]

LDP as hypothesis testing → Attacker receives an output drawn either from $\mathcal{M}(v)$ or $\mathcal{M}(v')$:

- For every attacker: $\underbrace{\Pr[\mathcal{A}(\mathcal{M}(v)) = v]}_{\text{True Positive (TPR)}} \leq e^{\epsilon} \cdot \underbrace{\Pr[\mathcal{A}(\mathcal{M}(v')) = v]}_{\text{False Positive (FPR)}}$

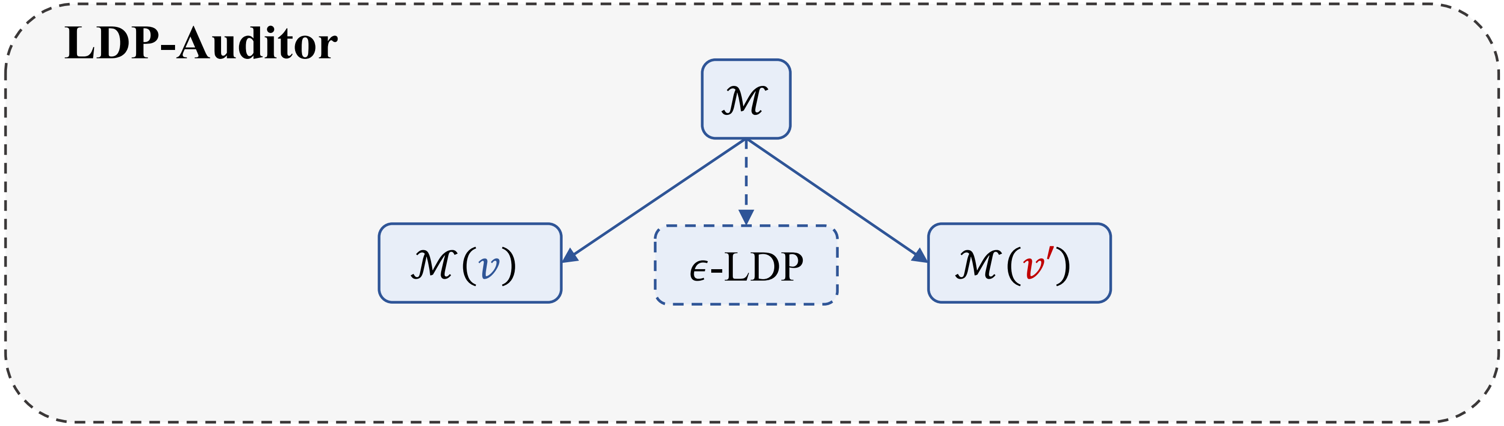


[AG24] Revealing the True Cost of Locally Differentially Private Protocols: An Auditing Perspective. PETS 2024.

Statistically Measuring LDP [AG24]

LDP as hypothesis testing → Attacker receives an output drawn either from $\mathcal{M}(v)$ or $\mathcal{M}(v')$:

- For every attacker: $\underbrace{\Pr[\mathcal{A}(\mathcal{M}(v)) = v]}_{\text{True Positive (TPR)}} \leq e^{\epsilon} \cdot \underbrace{\Pr[\mathcal{A}(\mathcal{M}(v')) = v]}_{\text{False Positive (FPR)}}$

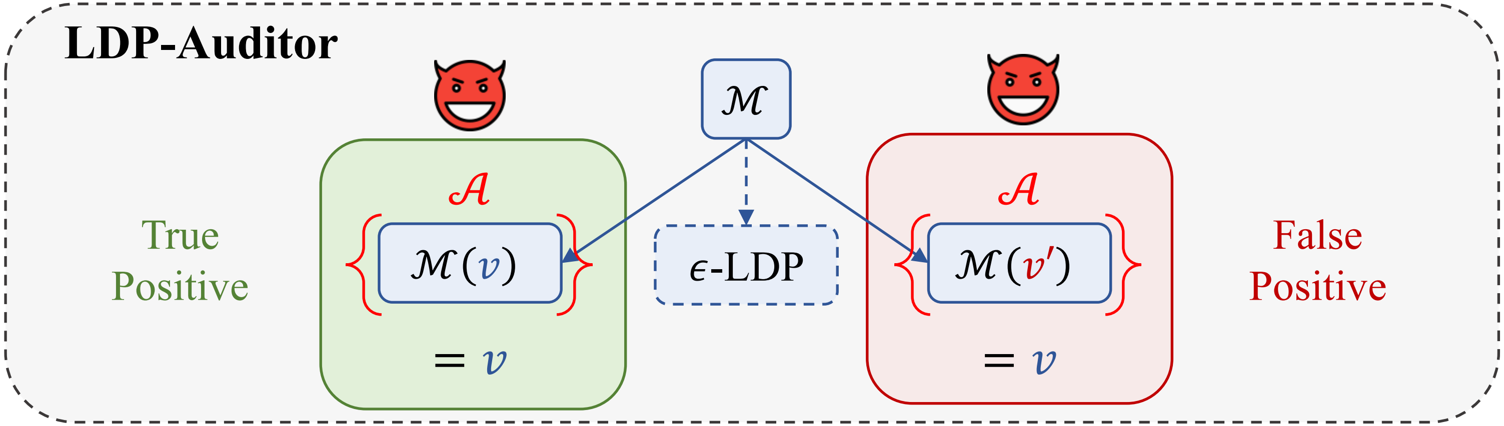


[AG24] Revealing the True Cost of Locally Differentially Private Protocols: An Auditing Perspective. PETS 2024.

Statistically Measuring LDP [AG24]

LDP as hypothesis testing → Attacker receives an output drawn either from $\mathcal{M}(v)$ or $\mathcal{M}(v')$:

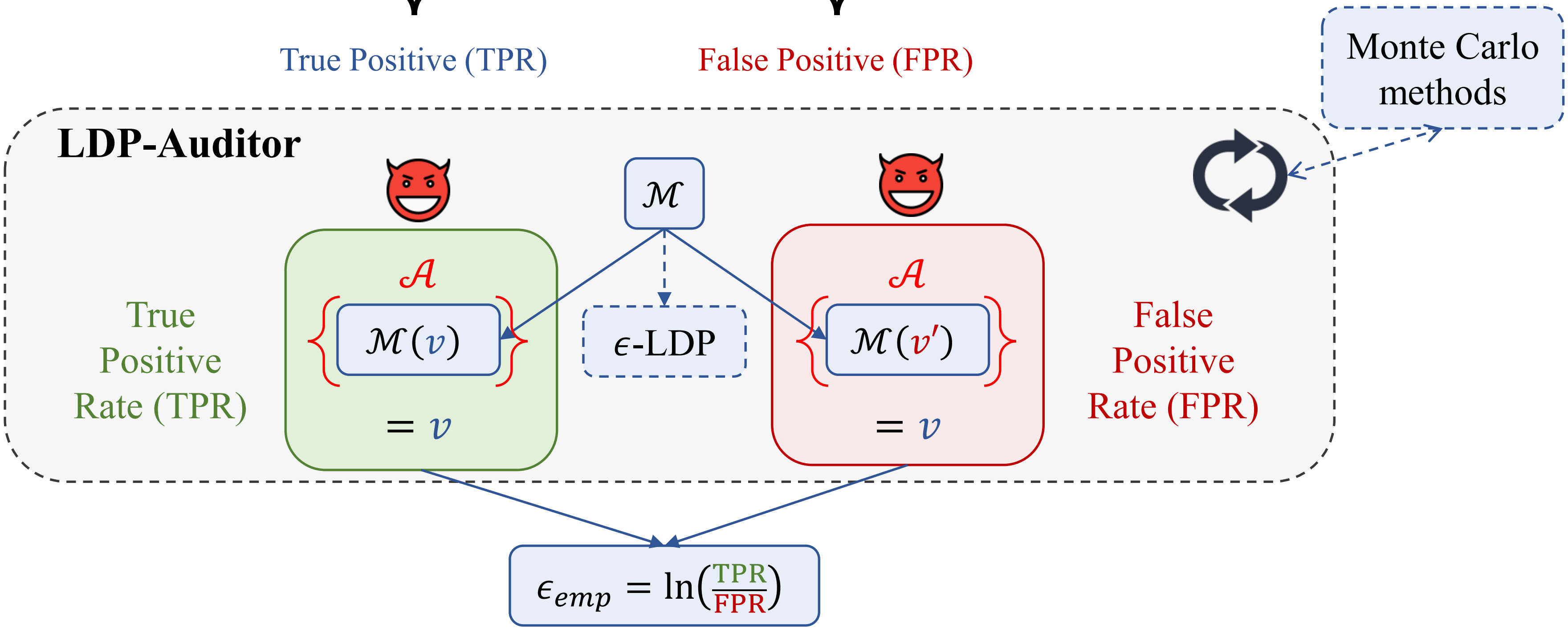
- For every attacker: $\underbrace{\Pr[\mathcal{A}(\mathcal{M}(v)) = v]}_{\text{True Positive (TPR)}} \leq e^{\epsilon} \cdot \underbrace{\Pr[\mathcal{A}(\mathcal{M}(v')) = v]}_{\text{False Positive (FPR)}}$



Statistically Measuring LDP [AG24]

LDP as hypothesis testing → Attacker receives an output drawn either from $\mathcal{M}(v)$ or $\mathcal{M}(v')$:

- For every attacker: $\underbrace{\Pr[\mathcal{A}(\mathcal{M}(v)) = v]}_{\text{True Positive (TPR)}} \leq e^{\epsilon} \cdot \underbrace{\Pr[\mathcal{A}(\mathcal{M}(v')) = v]}_{\text{False Positive (FPR)}}$

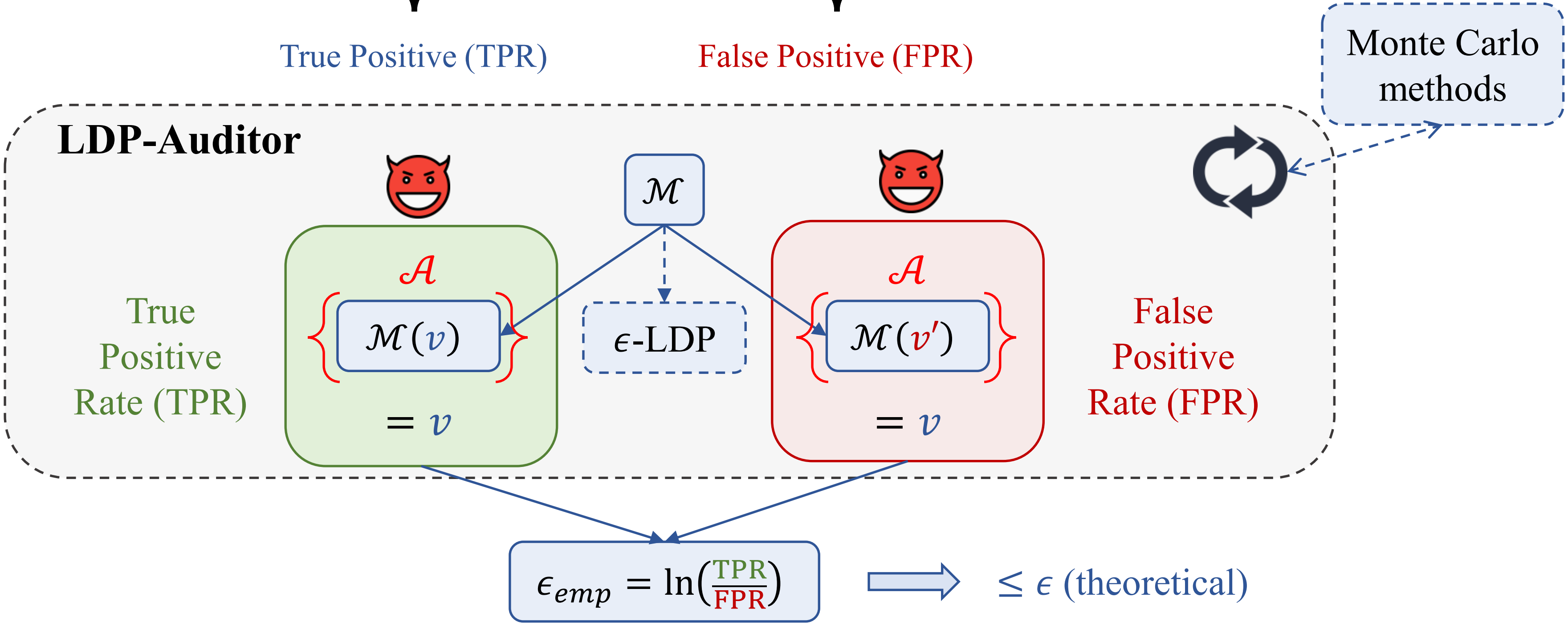


[AG24] Revealing the True Cost of Locally Differentially Private Protocols: An Auditing Perspective. PETS 2024.

Statistically Measuring LDP [AG24]

LDP as hypothesis testing → Attacker receives an output drawn either from $\mathcal{M}(v)$ or $\mathcal{M}(v')$:

- For every attacker: $\underbrace{\Pr[\mathcal{A}(\mathcal{M}(v)) = v]}_{\text{True Positive (TPR)}} \leq e^{\epsilon} \cdot \underbrace{\Pr[\mathcal{A}(\mathcal{M}(v')) = v]}_{\text{False Positive (FPR)}}$



[AG24] Revealing the True Cost of Locally Differentially Private Protocols: An Auditing Perspective. PETS 2024.

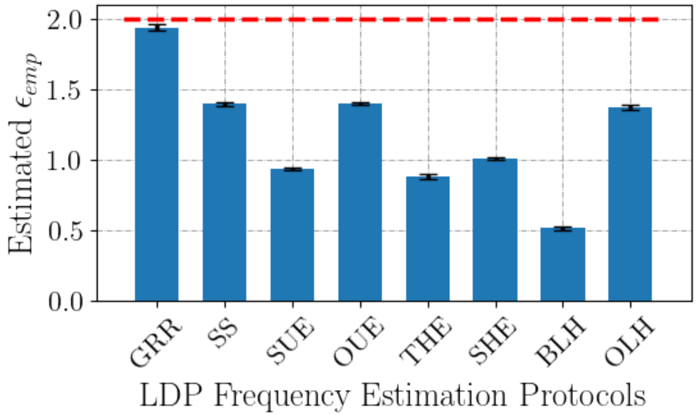
Instance of LDP Audit Results [AG24]

Setup:

- Eight fundamental LDP protocols.
- Theoretical $\epsilon = 2$ (red dashed line).

Main Insights:

- Distinct auditing results due to different encoding & perturbation LDP functions.



[AG24] Revealing the True Cost of Locally Differentially Private Protocols: An Auditing Perspective. PETS 2024.

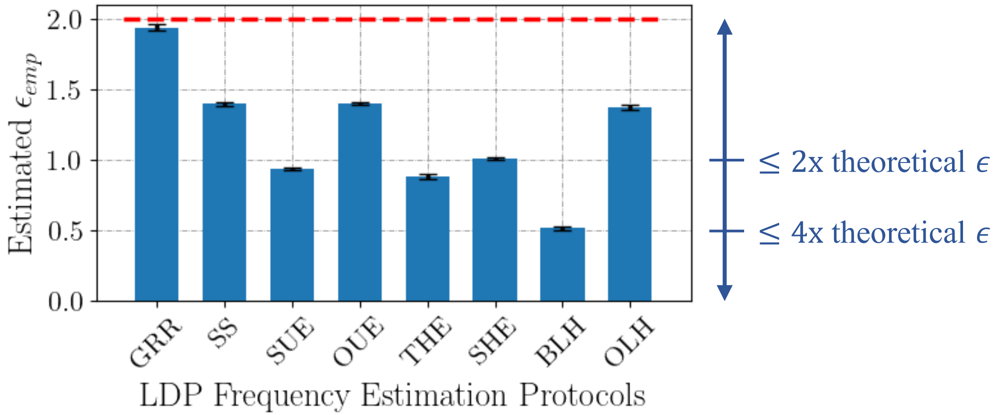
Instance of LDP Audit Results [AG24]

Setup:

- Eight fundamental LDP protocols.
- Theoretical $\epsilon = 2$ (red dashed line).

Main Insights:

- Distinct auditing results due to different encoding & perturbation LDP functions.
- GRR is the only LDP protocol that yields tight empirical privacy estimates ($i.e.$, $\epsilon_{emp} \approx \epsilon$).



Instance of LDP Audit Results [AG24]

Setup:

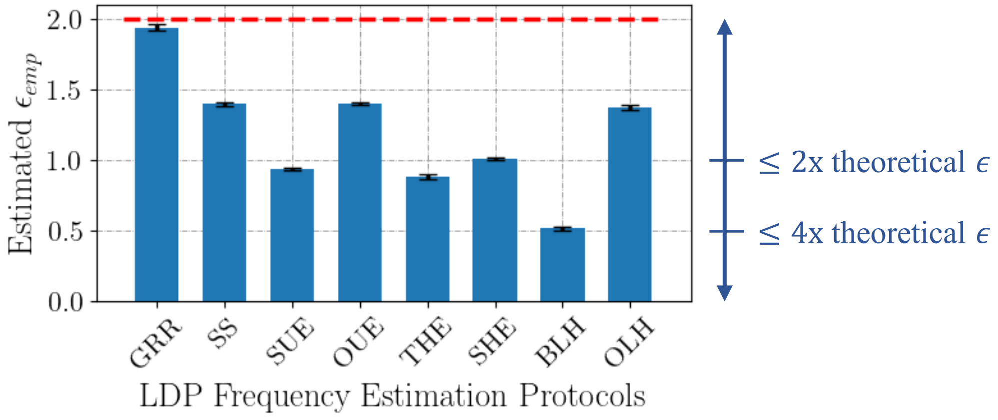
- Eight fundamental LDP protocols.
- Theoretical $\epsilon = 2$ (red dashed line).

Main Insights:

- Distinct auditing results due to different encoding & perturbation LDP functions.
- GRR is the only LDP protocol that yields tight empirical privacy estimates (*i.e.*, $\epsilon_{emp} \approx \epsilon$).

Hypotheses:

- State-of-the-art attacks are **not** strong enough...?
- Privacy gain in the encoding step (*e.g.*, LH)...?



Instance of LDP Audit Results [AG24]

Question → Can LDP-Auditor also help finding bugs in LDP implementations?

General Setup:

- LDP Python package: pure-ldp [M21, CMM21].
- LDP protocols: Symmetric UE (SUE) and Optimal UE (OUE).

[M21] pure-LDP: https://github.com/Samuel-Maddock/pure-LDP
[CMM21] Frequency Estimation under Local Differential Privacy. VLDB 2021.

Instance of LDP Audit Results [AG24]

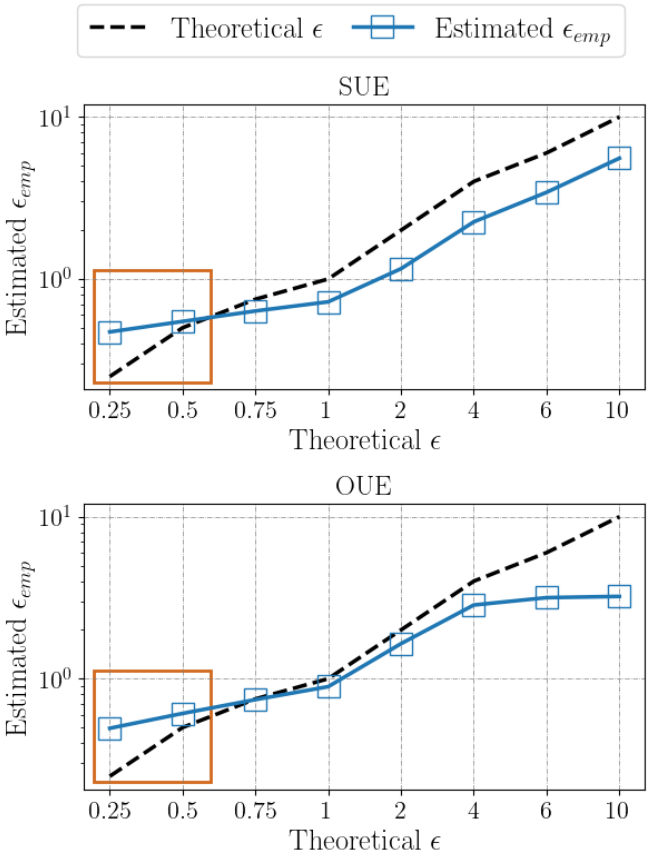
Question → Can LDP-Auditor also help finding bugs in LDP implementations?

General Setup:

- LDP Python package: pure-ldp [M21, CMM21].
- LDP protocols: Symmetric UE (SUE) and Optimal UE (OUE).

Main Insights:

- UE implementation with ϵ-LDP violation (*i.e.*, $\epsilon_{emp} > \epsilon$).
- Missing step in code reported to authors.
- Bug fixed with new pure-LDP version 1.2.0 [M21].



[M21] pure-LDP: https://github.com/Samuel-Maddock/pure-LDP
[CMM21] Frequency Estimation under Local Differential Privacy. VLDB 2021.

Outline

- Module 1 (Introduction):
  - Review of DP and preliminaries
  - LDP introduction
  - State-of-the-art deployments of LDP

- **Module 2 (Current research directions):**
  - Privacy attacks on LDP protocols
  - **Security attacks on LDP protocols**
  - Final remarks & open problems

Security Vulnerabilities of LDP Protocols [CJG21, CSU21]

Spoil the estimated statistic at the server side.

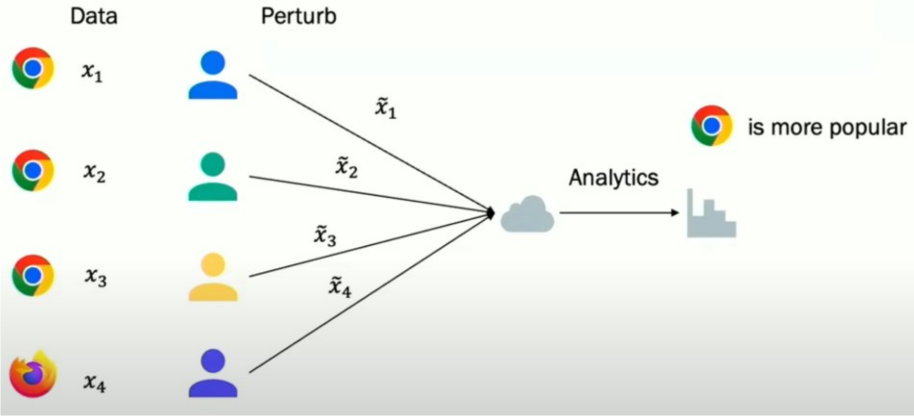
- Data poisoning attack: Target items.
- Manipulation attacks: No target items.

[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.
[CSU21] Manipulation Attacks in Local Differential Privacy. IEEE S&P 2021.

Security Vulnerabilities of LDP Protocols [CJG21, CSU21]

Spoil the estimated statistic at the server side.

- Data poisoning attack: Target items.
- Manipulation attacks: No target items.

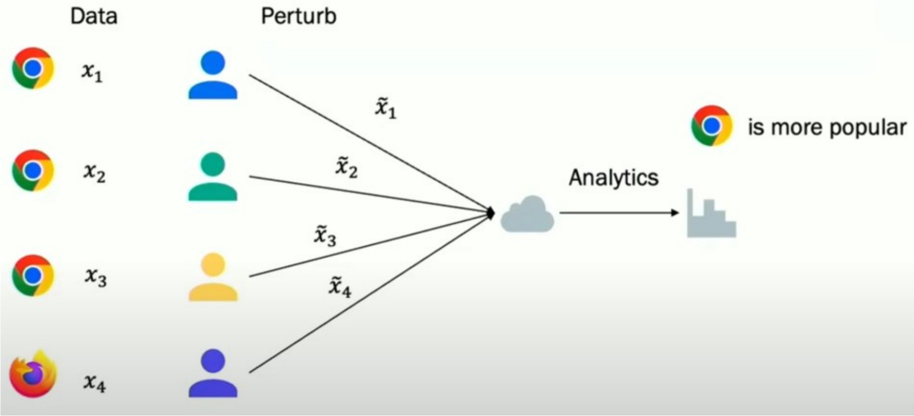


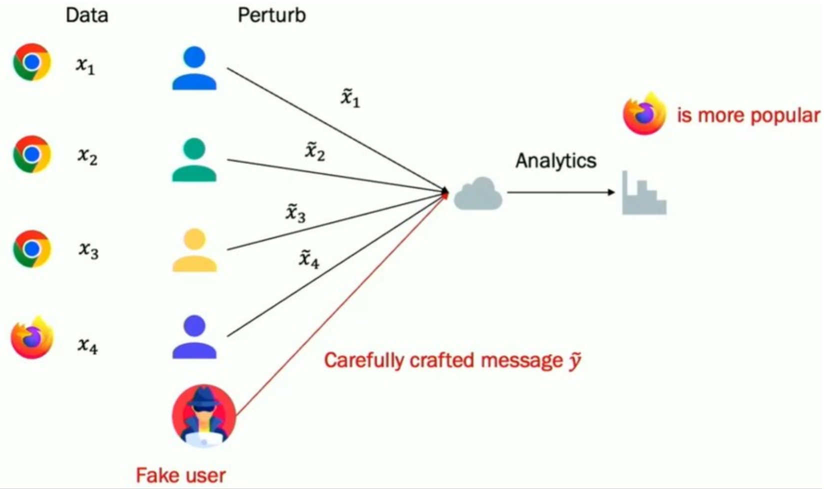
[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.
[CSU21] Manipulation Attacks in Local Differential Privacy. IEEE S&P 2021.

Security Vulnerabilities of LDP Protocols [CJG21, CSU21]

Spoil the estimated statistic at the server side.

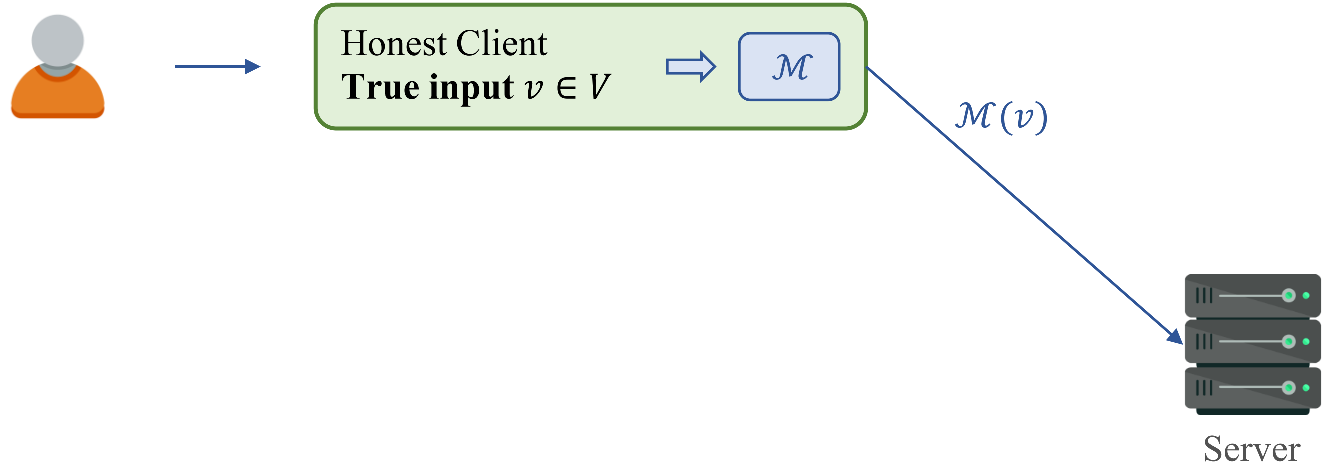
- Data poisoning attack: Target items.
- Manipulation attacks: No target items.





[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.
[CSU21] Manipulation Attacks in Local Differential Privacy. IEEE S&P 2021.

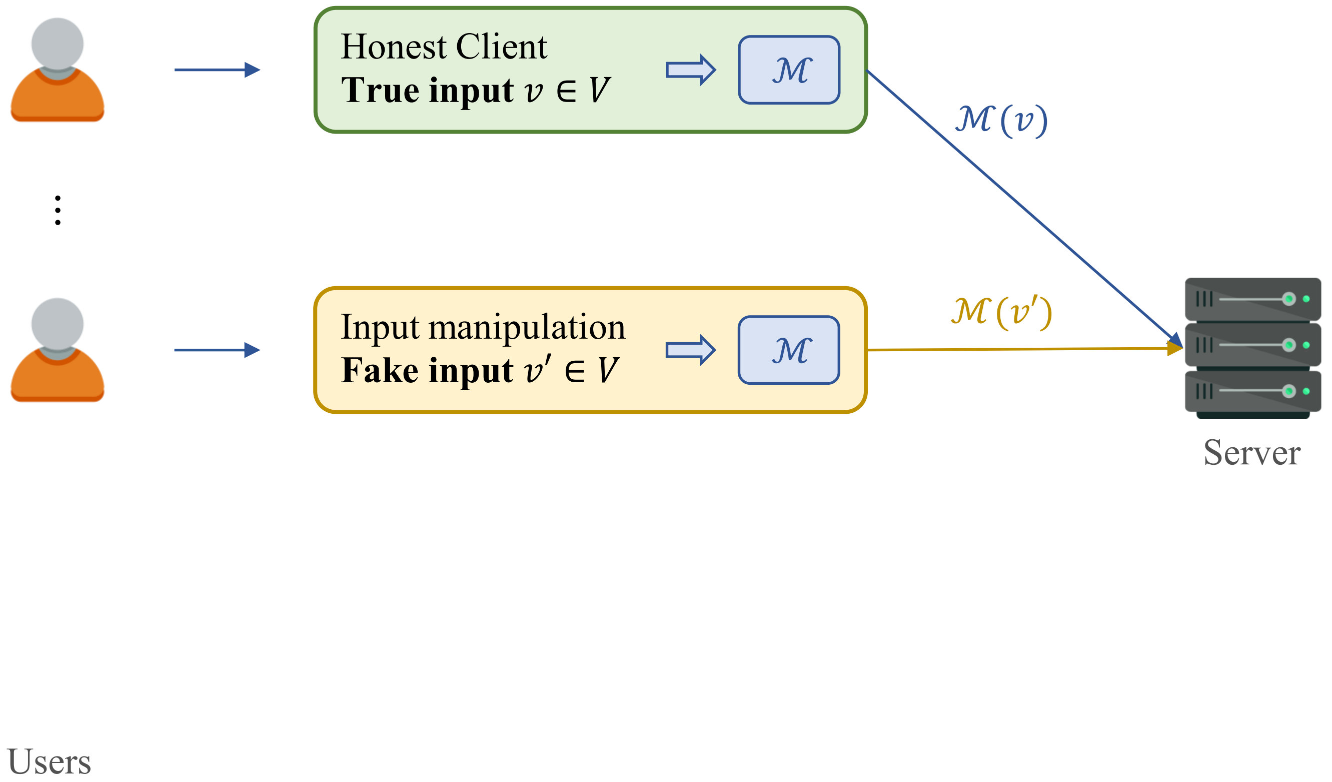
Security Vulnerabilities of LDP Protocols [CJG21, CSU21]



Users

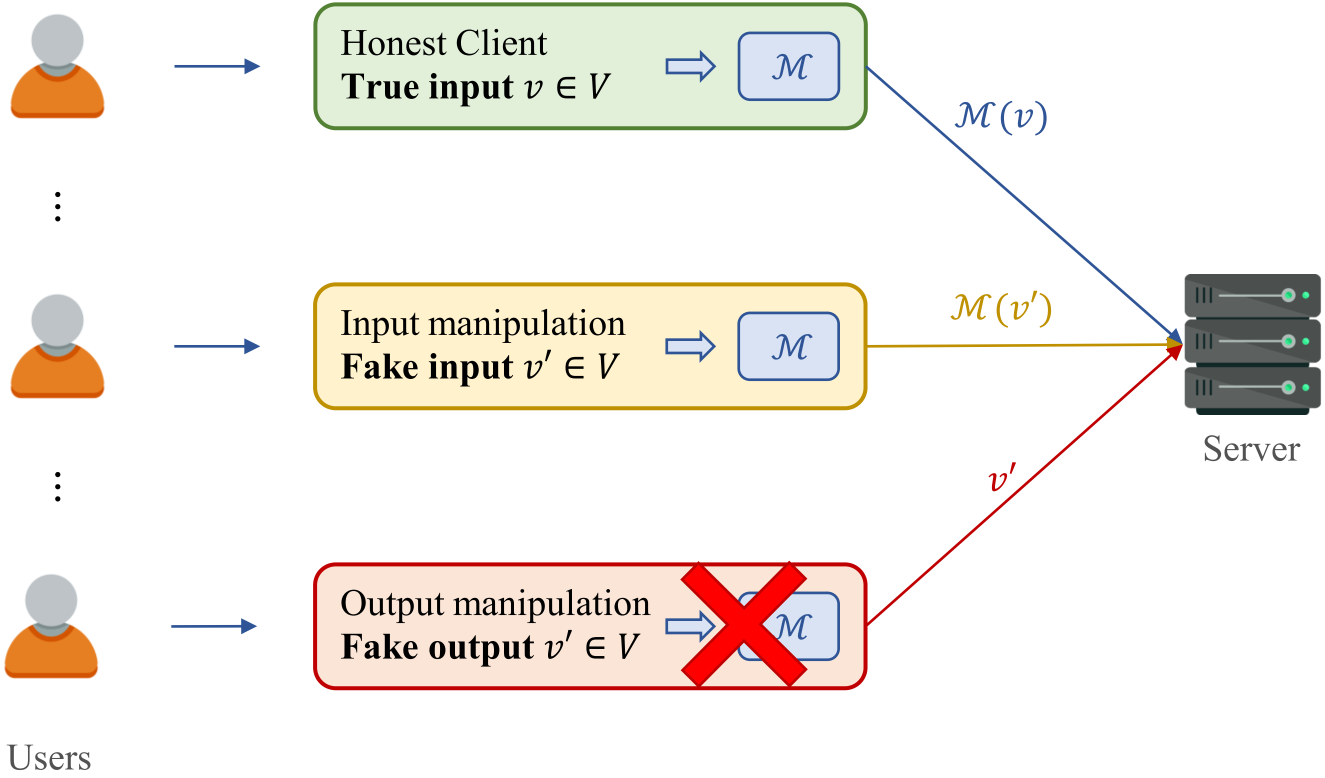
[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.
[CSU21] Manipulation Attacks in Local Differential Privacy. IEEE S&P 2021.

Security Vulnerabilities of LDP Protocols [CJG21, CSU21]



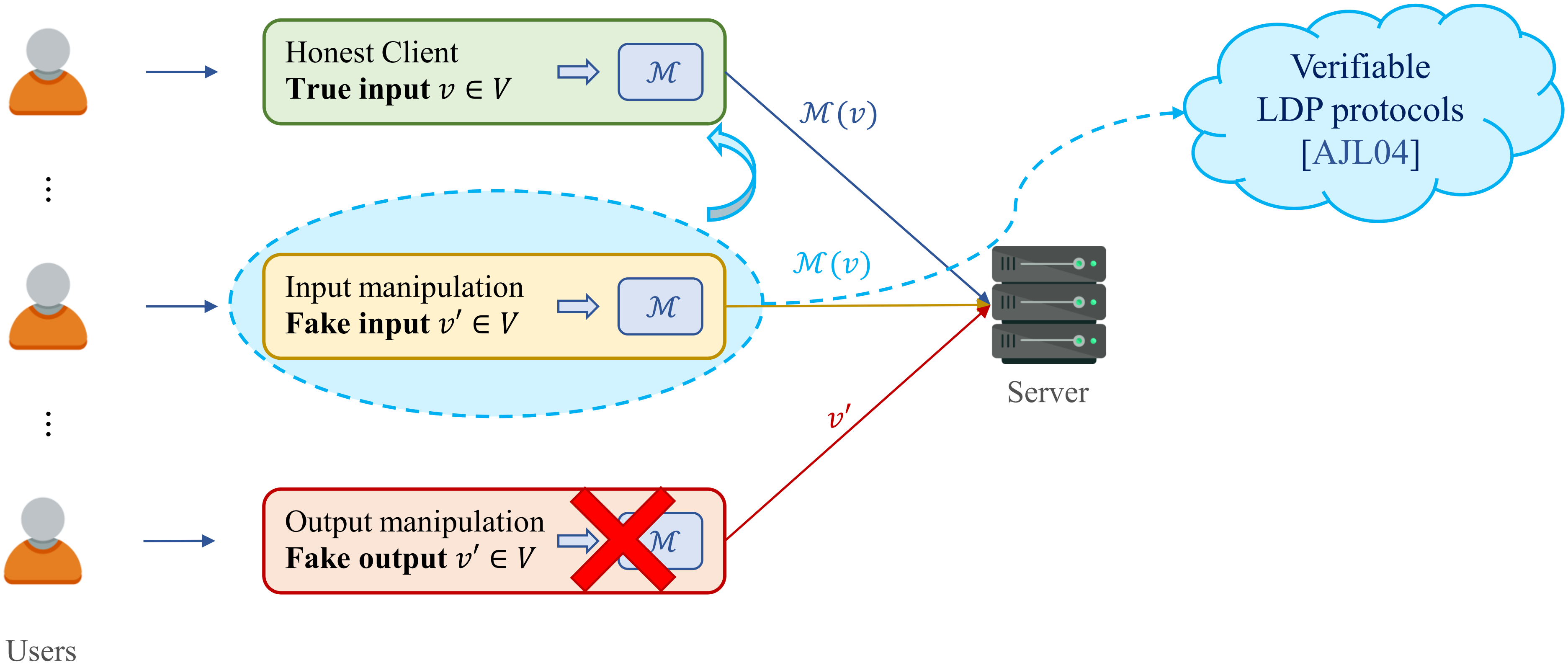
[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.
[CSU21] Manipulation Attacks in Local Differential Privacy. IEEE S&P 2021.

Security Vulnerabilities of LDP Protocols [CJG21, CSU21]



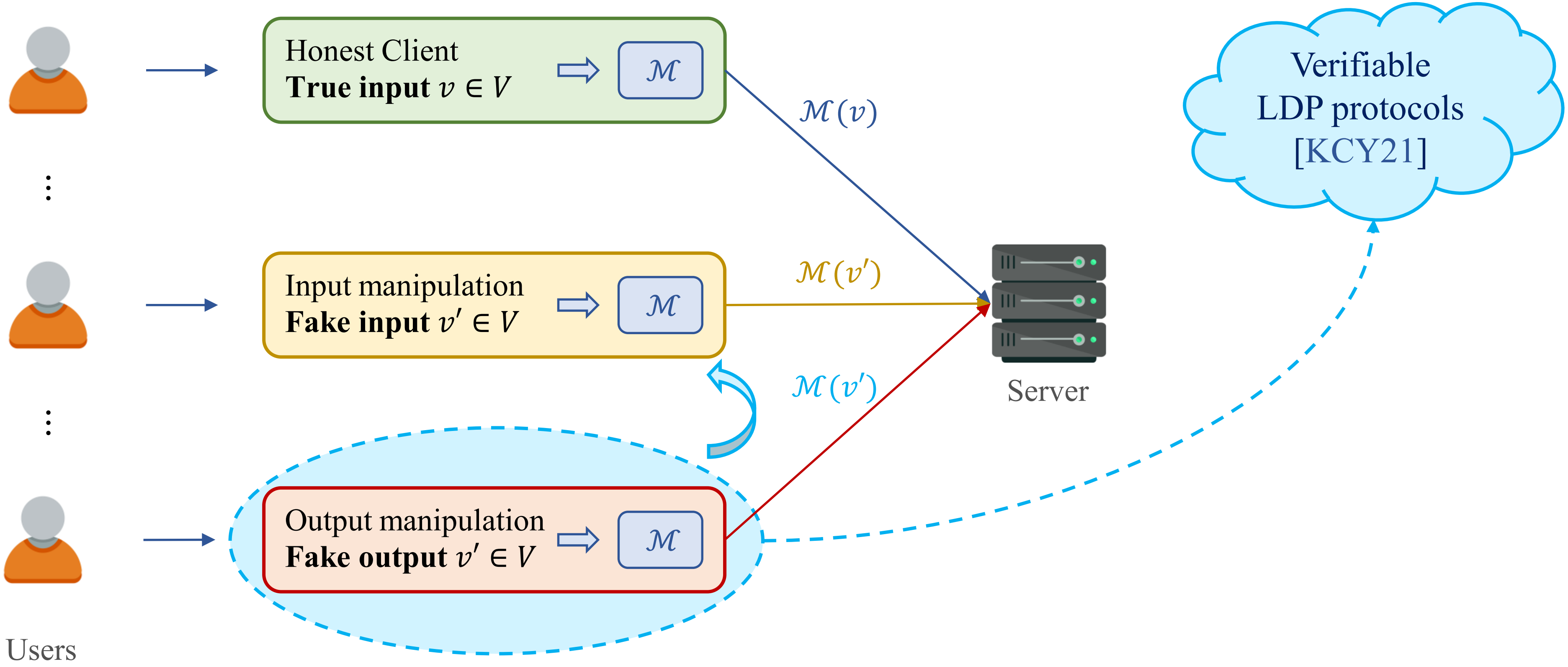
[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.
[CSU21] Manipulation Attacks in Local Differential Privacy. IEEE S&P 2021.

Security Vulnerabilities of LDP Protocols [CJG21, CSU21]



[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.
[CSU21] Manipulation Attacks in Local Differential Privacy. IEEE S&P 2021.
[AJL04] Cryptographic randomized response techniques. PKC 2004.

Security Vulnerabilities of LDP Protocols [CJG21, CSU21]



[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.
[CSU21] Manipulation Attacks in Local Differential Privacy. IEEE S&P 2021.
[KCY21] Preventing Output-Manipulation in LDP using Verifiable Randomization Mechanism. DBSec 2021.

Data Poisoning Attacks to LDP Protocols [CJG21]

Goal:

- Promote a set of target items T.
- Increasing their estimated frequency.

Background knowledge:

- LDP protocol.

Capability:

- Inject fake accounts.

Fake accounts are cheap!

Prices through the course of our analysis range from $0.01 to $0.20 per Twitter account, with a median cost of $0.04 for all merchants. Despite the large overall span,

Yahoo Yahoo accounts, like Hotmail, are widely available, with prices ranging from $0.006 – 0.015 per account.

[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.

Data Poisoning Attacks to LDP Protocols [CJG21]

Metrics:

- Frequency gain: $\Delta\widetilde{f_t} = \widetilde{f_{t,a}} - \widetilde{f_{t,b}}$, $f_{t,a}$: after attack, $f_{t,b}$: before attack.
- Overall gain: $G = \sum_{t \in T} \mathbb{E}(\Delta\widetilde{f_t})$.
- G depends on the set of attacker-crafted perturbed values $\boldsymbol{Z}$.
- Attacker manipulates Encode/Perturb to craft $\boldsymbol{Z}$ that maximizes G.
- Attacker controls m fake users.
- Fraction of fake users: $\beta = \frac{m}{n+m}$.

[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.

Data Poisoning Attacks to LDP Protocols [CJG21]

Attacks:

- Random perturbed-value attack (RPA):
  - Each fake user randomly selects $z \in V$.

Non-targeted
"output manipulation"

[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.

Data Poisoning Attacks to LDP Protocols [CJG21]

Attacks:

- Random perturbed-value attack (RPA):
    - Each fake user randomly selects $z \in V$.
- Random item attack (RIA):
    - Each fake user randomly selects a target item $t \in T$.
    - Follow the LDP protocol to generate z.

Non-targeted
"output manipulation"

"input manipulation"

[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.

Data Poisoning Attacks to LDP Protocols [CJG21]

Attacks:

- Random perturbed-value attack (RPA):
  - Each fake user randomly selects $z \in V$.
- Random item attack (RIA):
  - Each fake user randomly selects a target item $t \in T$.
  - Follow the LDP protocol to generate z.
- Maximal gain attack (MGA):
  - Find $\boldsymbol{Z}$ by solving $\max_{\boldsymbol{Z}} G(\boldsymbol{Z})$.
  - Maximize the number of items that z supports.
  - Randomly sets other bits such that number of 1's seems normal.

Non-targeted
"output manipulation"

"input manipulation"

Targeted
"output manipulation"

[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.

Data Poisoning Attacks to LDP Protocols [CJG21]

Attacks:

- Random perturbed-value attack (RPA):
  - Each fake user randomly selects $z \in V$.
- Random item attack (RIA):
  - Each fake user randomly selects a target item $t \in T$.
  - Follow the LDP protocol to generate z.
- Maximal gain attack (MGA):
  - Find $\boldsymbol{Z}$ by solving $\max_{\boldsymbol{Z}} G(\boldsymbol{Z})$.
  - Maximize the number of items that z supports.
  - Randomly sets other bits such that number of 1's seems normal.

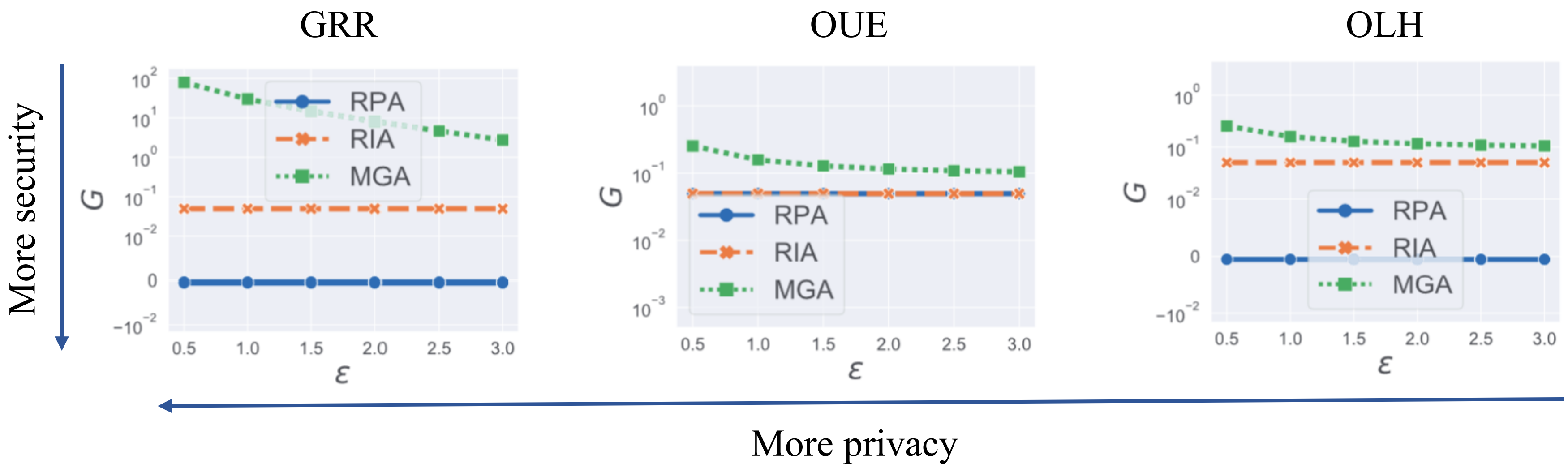
Ex. MGA with OUE

$T = \{2,4\}$

0	1	0	1	0	0	0	0 1

[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.

Data Poisoning Attacks to LDP Protocols [CJG21]

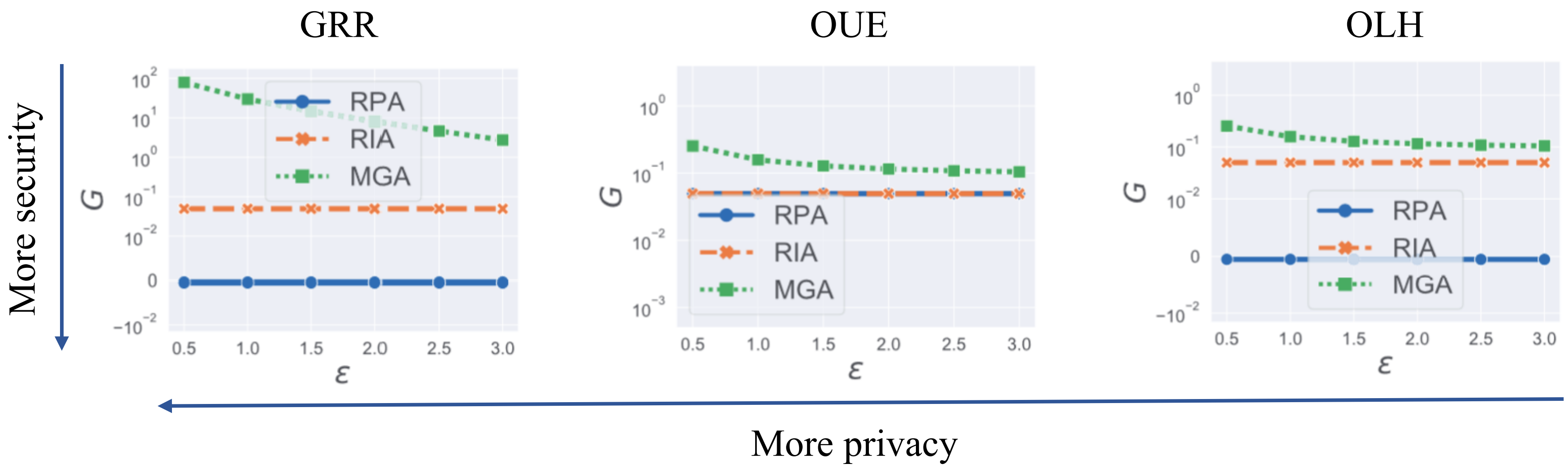


[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.

Data Poisoning Attacks to LDP Protocols [CJG21]

There is a security-privacy trade-off for the LDP protocols!

Smaller $\epsilon \rightarrow$ stronger privacy and weaker security!



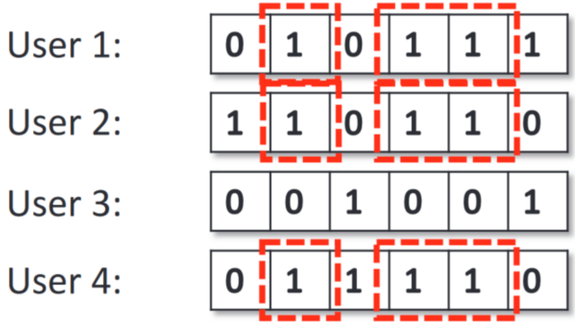
[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.

Data Poisoning Attacks to LDP Protocols [CJG21]

Countermeasures:

- Normalization:
    - Normalize estimated frequencies to form a distribution.

- Detecting fake users:
    - MGA max the gain with ***Z*** supporting all target items.
    - Common pattern in z of fake users.
    - Detect via frequent itemset mining.



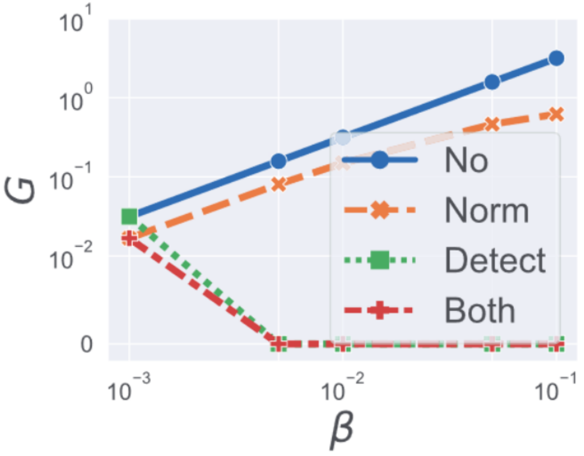
[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.

Data Poisoning Attacks to LDP Protocols [CJG21]

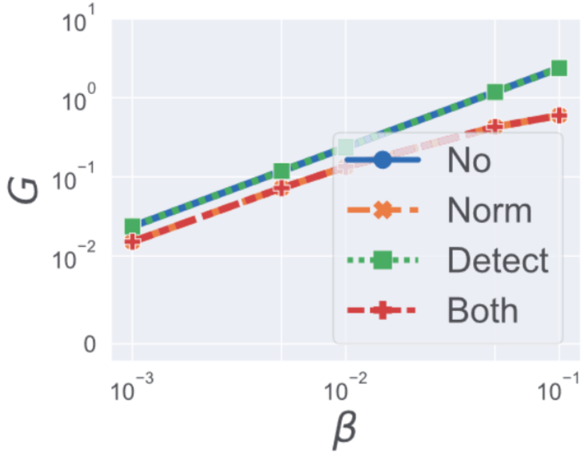
Detecting and removing fake users:

- Privacy parameter: $\epsilon = 1$.
- Fraction of fake users: $\beta = \frac{m}{n+m}$.

OUE



OLH



[CJG21] Data poisoning attacks to local differential privacy protocols. USENIX Security 2021.

Recent Advances on Security Vulnerabilities of LDP Protocols

LDP protocols are highly vulnerable to manipulation/poisoning attacks:

- Data poisoning attacks can effectively promote target items.
- There is an inherently security-privacy trade-off in LDP protocols.

New attacks/countermeasures:

- Poisoning attacks on different data types (or tasks) [WCJG22, LLSGL23, TCNZ24].
- Preventing output-manipulation attacks via verifiable LDP [KCY21, HKY23, SXZ23].
- Neutralizing data poisoning attacks [HOYHZZZZ24, SYHDWXY24].

[WCJG22] Poisoning attacks to local differential privacy protocols for Key-Value data. USENIX Security 2022.
[LLSGL23] Fine-grained poisoning attack to LDP protocols for mean and variance estimation. USENIX Security 2023.
[TCNZ24] Data Poisoning Attacks to Locally Differentially Private Frequent Itemset Mining Protocols. CCS 2024.
[KCY21] Preventing Output-Manipulation in LDP using Verifiable Randomization Mechanism. DBSec 2021.
[HKY23] Local differential privacy protocol for making key-value data robust against poisoning attacks. MDAI 2024.
[SXZ23] Efficient Defenses Against Output Poisoning Attacks on Local Differential Privacy. IEEE TIFS.
[HOYHZZZZ24] LDPGuard: Defenses against data poisoning attacks to LDP protocols. IEEE TKDE.
[SYHDWXY24] LDPRecover: Recovering frequencies from poisoning attacks against LDP. ICDE 2024.

Outline

- Module 1 (Introduction):
  - Review of DP and preliminaries
  - LDP introduction
  - State-of-the-art deployments of LDP

- **Module 2 (Current research directions):**
  - Privacy attacks on LDP protocols
  - Security attacks on LDP protocols
  - **Final remarks & open problems**

Final Remarks

Recap of key insights:

- **Trust models of DP:** Central, local, and shuffle DP.
- **Core principles of LDP:** Minimal trust assumptions, data obfuscated at the user side.

Final Remarks

Recap of key insights:

- **Trust models of DP:** Central, local, and shuffle DP.
- **Core principles of LDP:** Minimal trust assumptions, data obfuscated at the user side.
- **Practical applications:** LDP is a big success for privacy research:
  - Adopted by Google, Apple, Microsoft for gathering statistics (*e.g.*, frequency).
  - LDP comes at a cost → Need many more users than central DP.
  - Privacy settings are 'not very tight' → deployed ϵ ranges from 0.5 to 16.

Final Remarks

Recap of key insights:

- **Trust models of DP:** Central, local, and shuffle DP.
- **Core principles of LDP:** Minimal trust assumptions, data obfuscated at the user side.
- **Practical applications:** LDP is a big success for privacy research:
  - Adopted by Google, Apple, Microsoft for gathering statistics (*e.g.*, frequency).
  - LDP comes at a cost → Need many more users than central DP.
  - Privacy settings are 'not very tight' → deployed ϵ ranges from 0.5 to 16.
- **Adversarial Considerations:** Yet, the LDP model is vulnerable to:
  - Privacy attacks → Bayesian adversary can infer the user's true value.
  - Security attacks → Data poisoning and manipulation attacks spoil statistical utility.

Final Remarks

Reflecting on LDP:

- **Opening private data:** LDP offers a decentralized approach that ensures privacy at the point of data collection, before any data leaves the user's device.
  - However, deployments of LDP are still tightly controlled by the server (*e.g.*, Google).
  - Could there be a more "open" implementation of LDP?

Final Remarks

Reflecting on LDP:

- **Opening private data:** LDP offers a decentralized approach that ensures privacy at the point of data collection, before any data leaves the user's device.
    - However, deployments of LDP are still tightly controlled by the server (*e.g.*, Google).
    - Could there be a more "open" implementation of LDP?
- **Balancing privacy, utility, robustness, communication cost:** A four-way optimization issue:
    - Enhancing one often comes at the expense of another(s)…

Final Remarks

Reflecting on LDP:

- **Opening private data:** LDP offers a decentralized approach that ensures privacy at the point of data collection, before any data leaves the user's device.
    - However, deployments of LDP are still tightly controlled by the server (*e.g.*, Google).
    - Could there be a more "open" implementation of LDP?
- **Balancing privacy, utility, robustness, communication cost:** A four-way optimization issue:
    - Enhancing one often comes at the expense of another(s)…
- **Closing encouragement:**
    - Think of LDP not just as a set of tools, but as a mindset that prioritizes privacy at every step of data handling.
    - LDP is not a one-size-fits-all solution → tailor LDP protocols to fit specific needs.

Final Remarks

Lots of open challenges:

- Take any data analysis/mining task and ask → *"Can we handle this under LDP?"*.
    - Sentiment analysis for (private) reviews → "LDP"-IMDB?
    - Trajectory analysis of GPS movements → "LDP"-Strava?

Final Remarks

Lots of open challenges:

- Take any data analysis/mining task and ask → *"Can we handle this under LDP?"*.
  - Sentiment analysis for (private) reviews → "LDP"-IMDB?
  - Trajectory analysis of GPS movements → "LDP"-Strava?
- Designing optimal LDP protocols for:
  - Evolving data, graph data, trajectory data, unstructured data (*e.g.*, text, video?), ...
  - Learning tasks (*i.e.*, machine learning, federated learning, gossip learning)…

Final Remarks

Lots of open challenges:

- Take any data analysis/mining task and ask → *“Can we handle this under LDP?”*.
    - Sentiment analysis for (private) reviews → “LDP”-IMDB?
    - Trajectory analysis of GPS movements → “LDP”-Strava?
- Designing optimal LDP protocols for:
    - Evolving data, graph data, trajectory data, unstructured data (*e.g.*, text, video?), ...
    - Learning tasks (*i.e.*, machine learning, federated learning, gossip learning)…
- Make LDP widely available → RAPPOR, pure-ldp, multi-freq-ldpy but just the beginning…

Final Remarks

Lots of open challenges:

- Take any data analysis/mining task and ask → *"Can we handle this under LDP?"*.
  - Sentiment analysis for (private) reviews → "LDP"-IMDB?
  - Trajectory analysis of GPS movements → "LDP"-Strava?
- Designing optimal LDP protocols for:
  - Evolving data, graph data, trajectory data, unstructured data (*e.g.*, text, video?), ...
  - Learning tasks (*i.e.*, machine learning, federated learning, gossip learning)…
- Make LDP widely available → RAPPOR, pure-ldp, multi-freq-ldpy but just the beginning…
- What are other emerging attack vectors in the context of LDP, and how can they be mitigated?
- How can we combine LDP with cryptographic techniques to provide stronger guarantees against sophisticated adversaries?

Thank You for Your Attention!

Questions?

CONTACT

Héber H. Arcolezi
Research Scientist
Inria Grenoble, France

heber.hwang-arcolezi@inria.fr
hharcolezi.github.io
@hharcolezi