Locally Differentially Private Protocols for Frequency Estimation

Tianhao Wang, Jeremiah Blocki, Ninghui Li, Somesh Jha

PURDUE UNIVERSITY

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

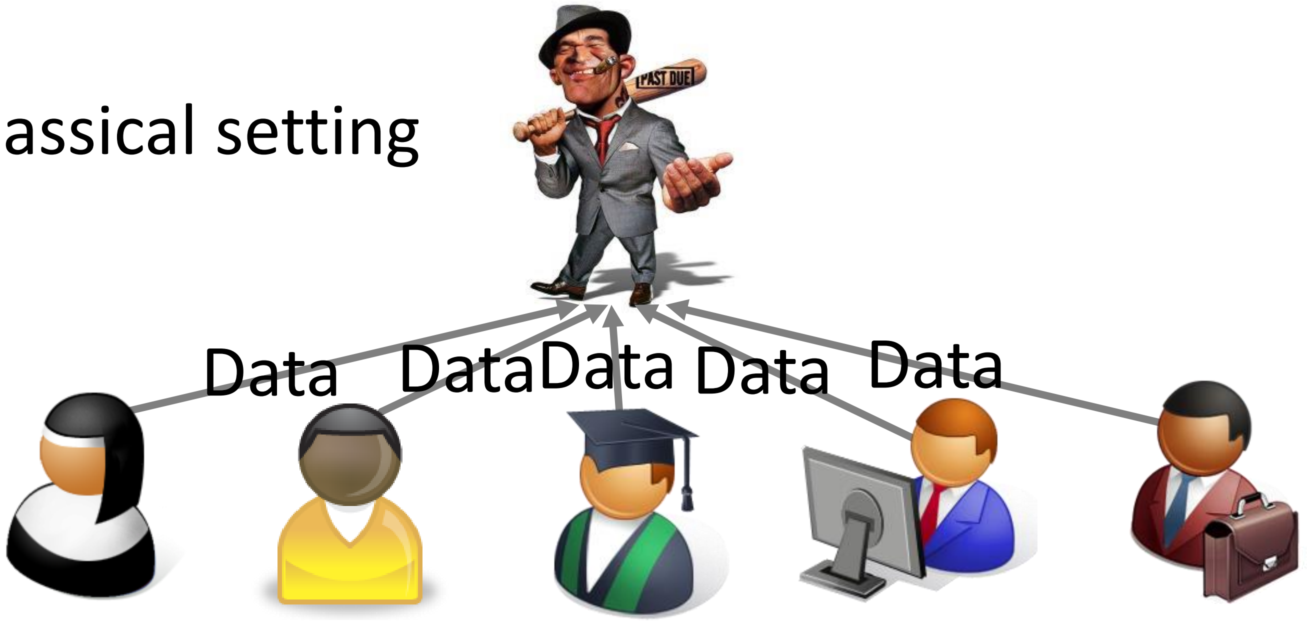
Differential Privacy

Differential Privacy

Classical setting

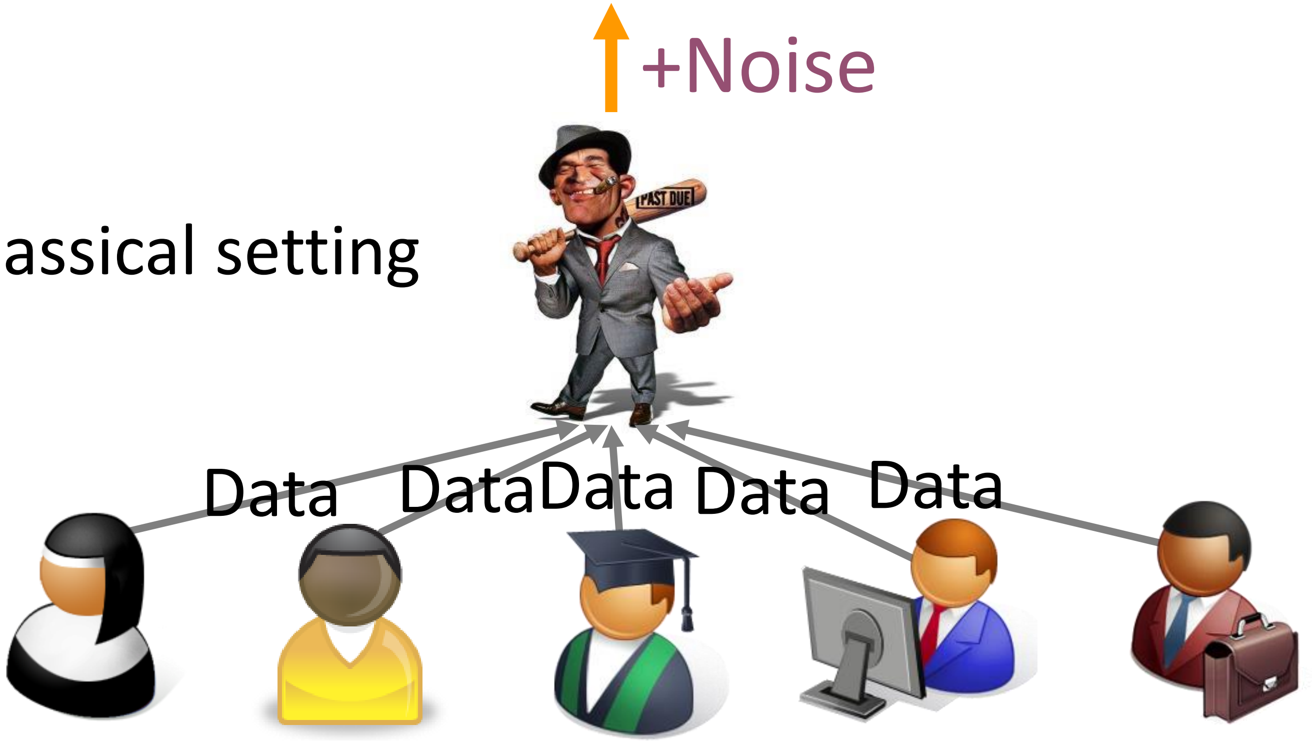
Differential Privacy

Classical setting



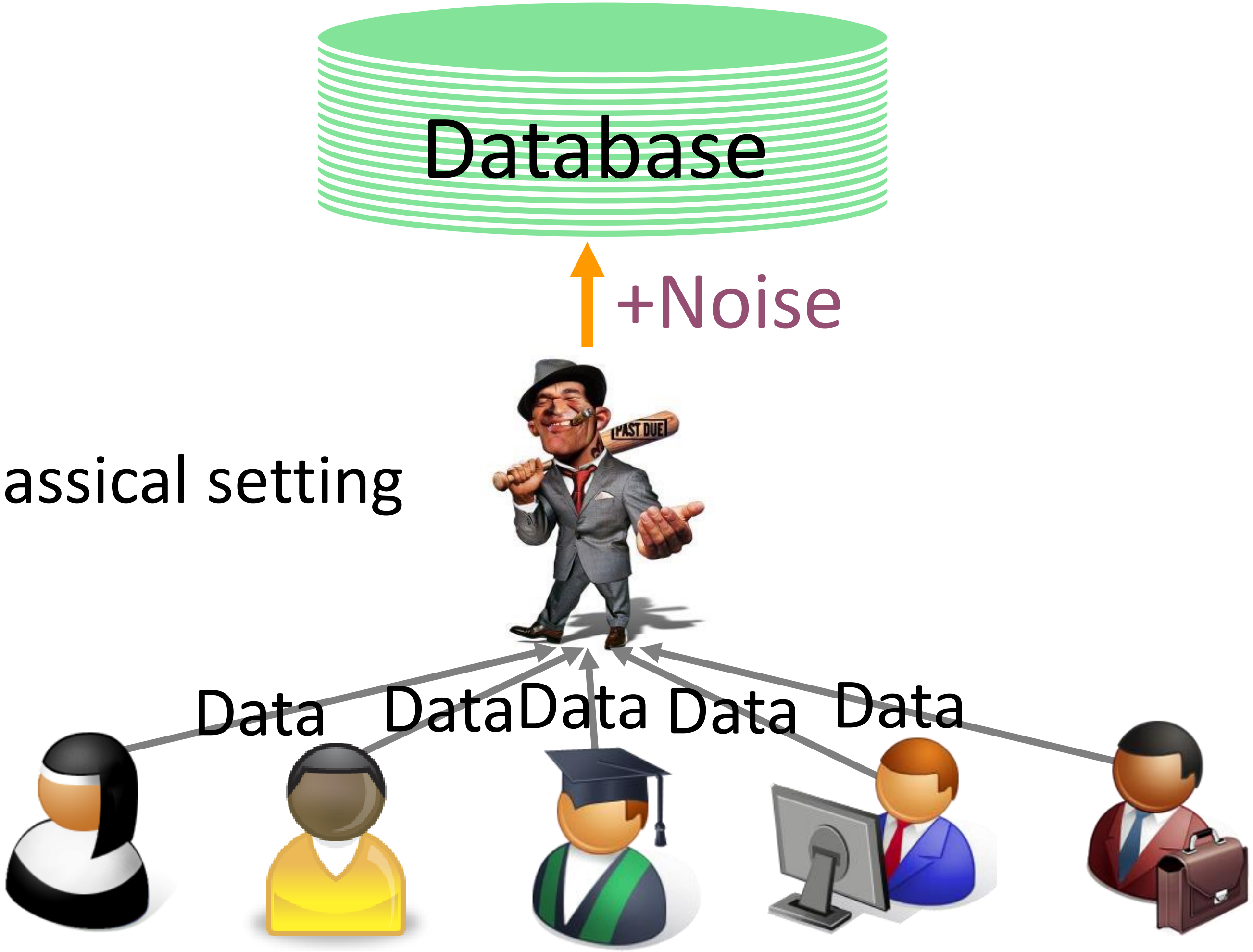
Differential Privacy

Classical setting

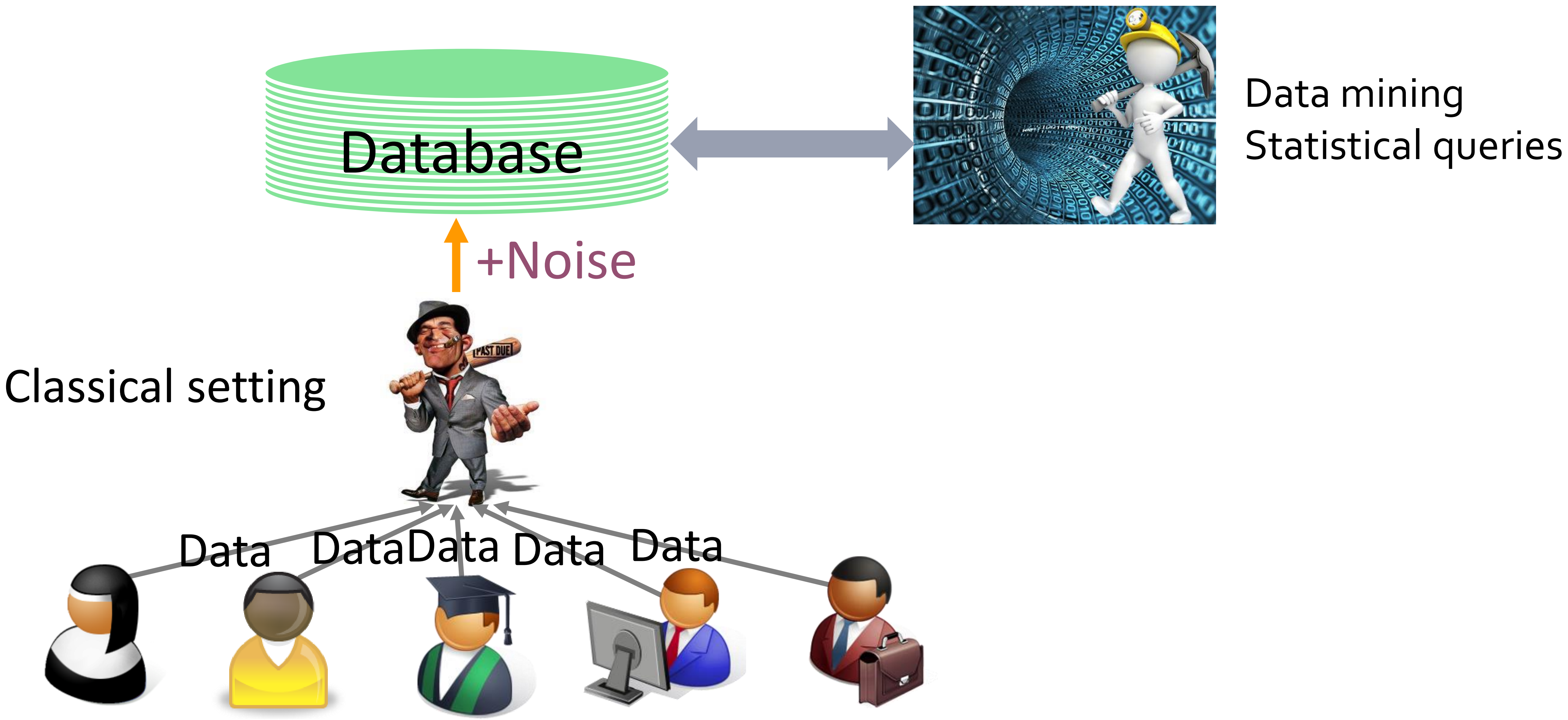


Differential Privacy

Classical setting



Differential Privacy



Differential Privacy

Database

+Noise

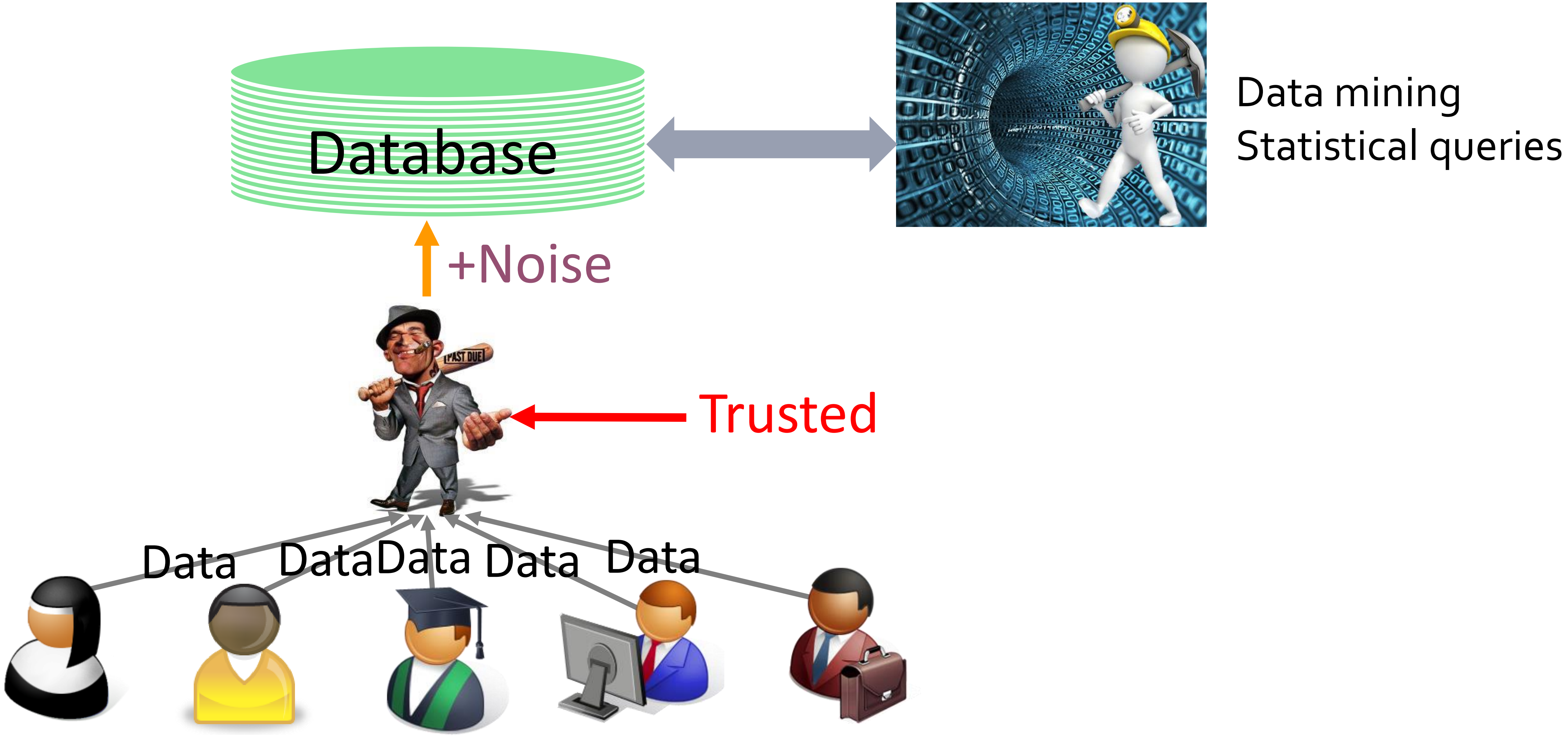
Data mining
Statistical queries

Classical setting

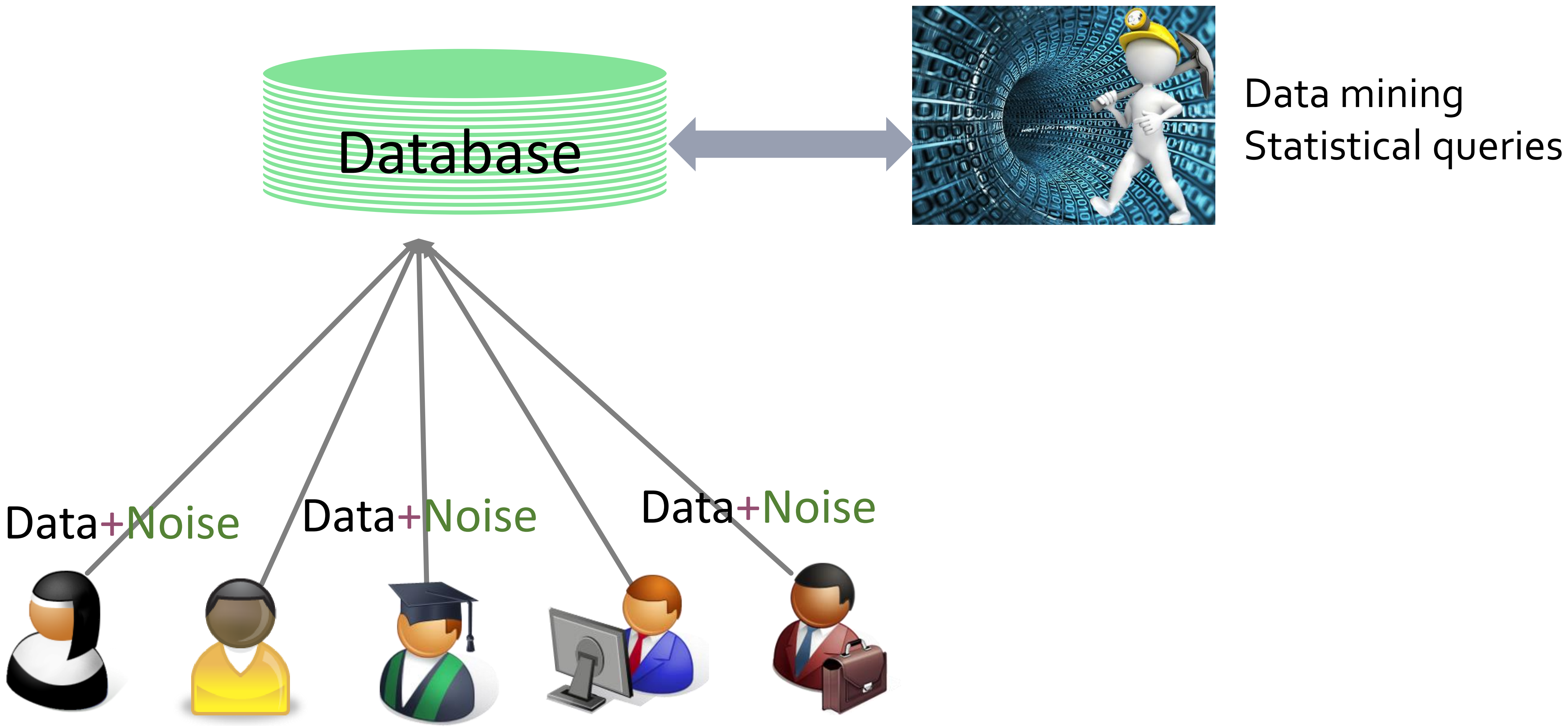
Differential Privacy Interpretation:
The decision to include/exclude individual’s record has minimal (ε) influence on the outcome.
Smaller ε ➔ Stronger Privacy

Data

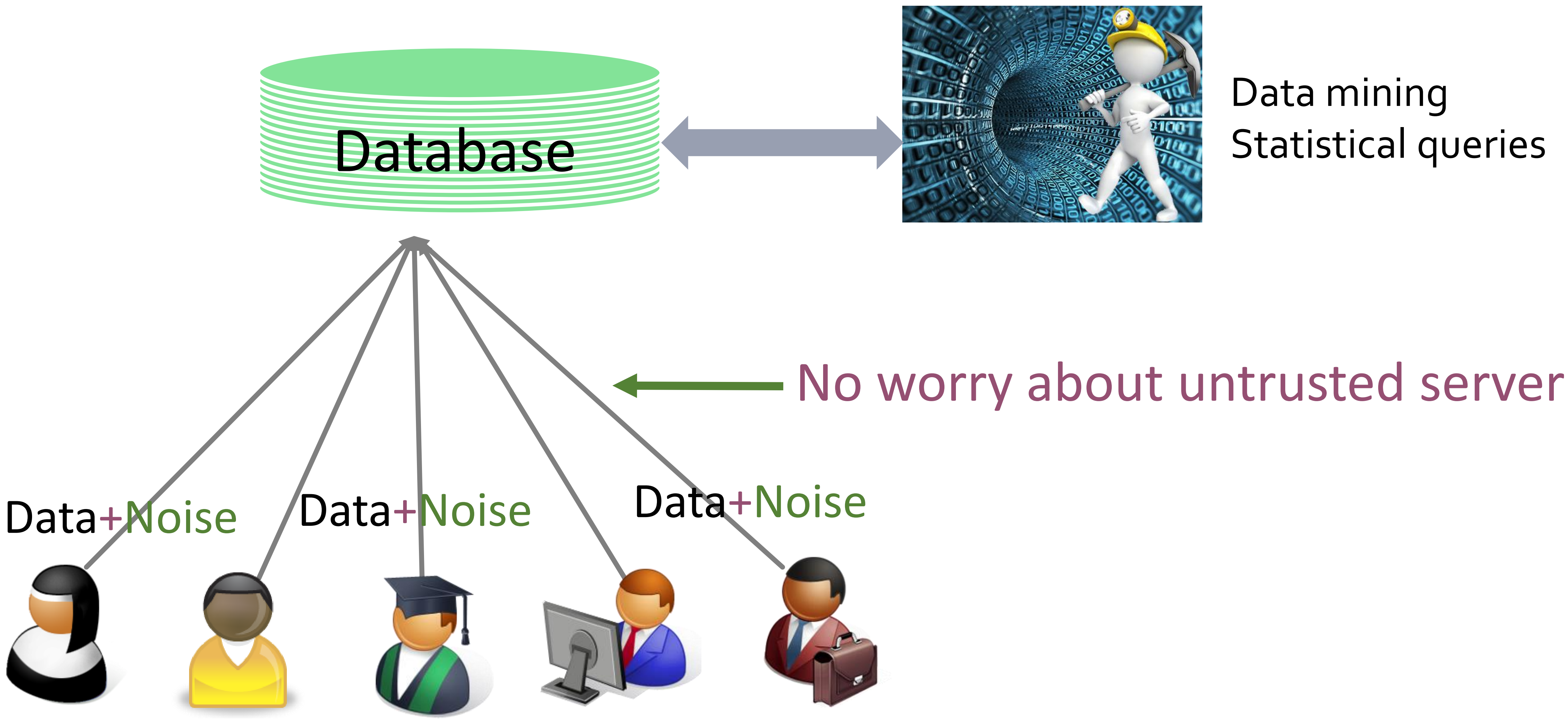
Differential Privacy



Local Differential Privacy



Local Differential Privacy



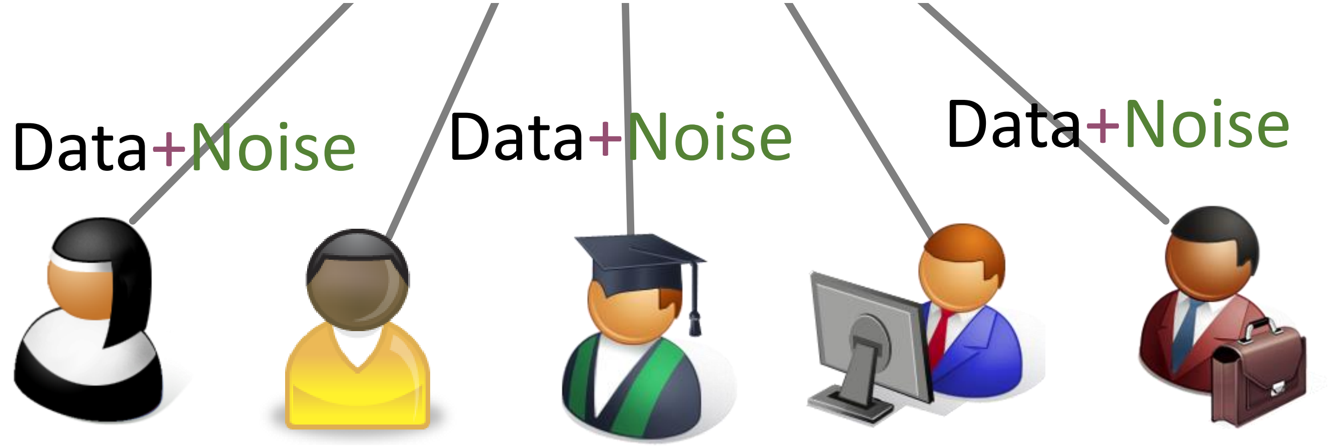
Local Differential Privacy

RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response

Úlfar Erlingsson
Google, Inc.
ulfar@google.com

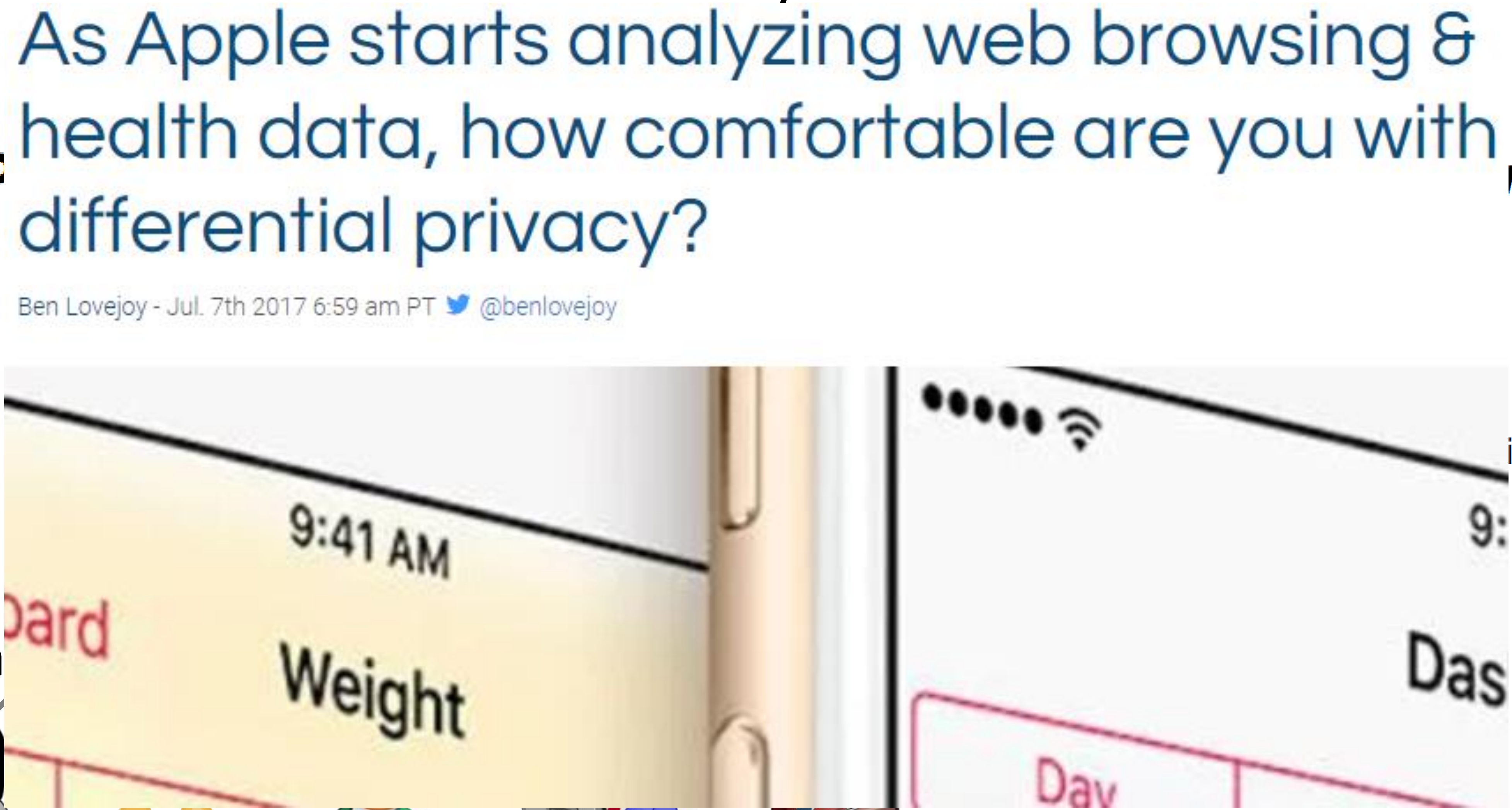
Vasyl Pihur
Google, Inc.
vpihur@google.com

Aleksandra Korolova
University of Southern California
korolova@usc.edu



Local Differential Privacy

RAP ving

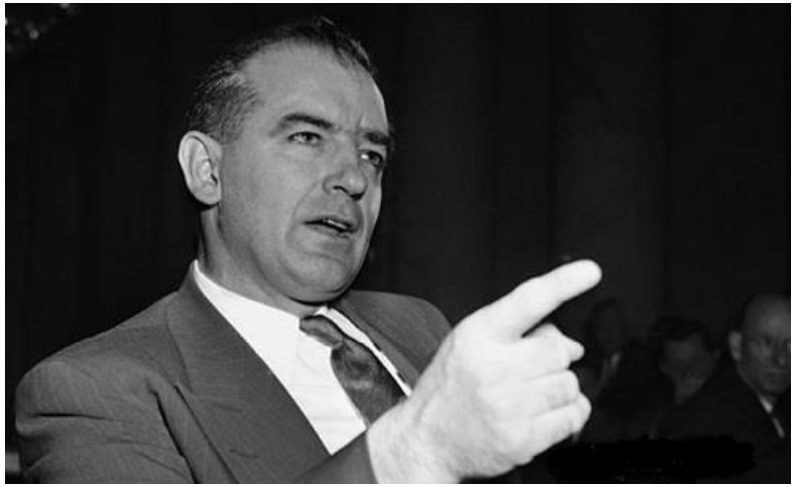


ia

er

Data

The Warner Model (1965)



- Survey technique for private questions
- Survey people:
  - "Are you communist party?"
- Each person:
  - Flip a secret coin
  - Answer truth if head (w/p 0.5)
  - Answer randomly if tail
  - E.g., a communist will answer "yes" w/p 75%, and "no" w/p 25%

Provide deniability:
Seeing answer, not certain about the secret.

- To get unbiased estimation of the distribution:
  - If n_v out of n people are communist, we expect to see $E[\,I_v] = 0.75n_v + 0.25(n - n_v)$ "yes" answers
  - $c(n_v) = \frac{I_v - 0.25n}{0.5}$ is the unbiased estimation of number of communists
  - Since $E[c(n_v)] = \frac{E[I_v] - 0.25n}{0.5} = n_v$

The Warner Model (1965)

- Survey technique for private questions
- Survey people:
  - "Are you comm
- Each person:
  - Flip a secret co
  - Answer truth i
  - Answer randomly if tail
  - E.g., a commu
- To get unbiased
  - If n_v out of n p ... $.75n_v + 0.25(n - n_v)$ "yes" answ
  - $c(n_v) = \frac{I_v - 0.25n}{0.5}$ is the unbiased estimation of number of communists
  - Since $E[c(n_v)] = \frac{E[I_v] - 0.25n}{0.5} = n_v$

...tain about the secret.

We say the protocol is ε -LDP iff for any $\boldsymbol{v}$ and $\boldsymbol{v}'$ from "yes" and "no",

$$\frac{\Pr[P(\boldsymbol{v}) = \boldsymbol{v}]}{\Pr[P(\boldsymbol{v}') = \boldsymbol{v}]} \le e^{\varepsilon}$$

This only handles binary attribute. We want to handle the more general setting.

Abstract LDP Protocol

- $x \coloneqq E(v)$
takes input value v from domain D and outputs an encoded value x
- $y \coloneqq P(x)$
takes an encoded value x and outputs y.

P satisfies ε-LDP

$\xrightarrow{y}$

- $c \coloneqq Est(\{y\})$
takes reports $\{y\}$ from all users and outputs estimations $c(v)$ for any value v in domain D

We focus on frequency estimation

Frequency Estimation Protocols

- Direct Encoding (Generalized Random Response) [Warner'65]
  - Generalize binary attribute to arbitrary domain
- Unary Encoding (Basic One-time RAPPOR) [Erlingsson et al'14]
  - Encode into a bit-vector and perturb each bit
- Binary Local Hash [Bassily and Smith'15]
  - Encode by hashing and then perturb

Direct Encoding (Random Response)

- User:
  - Encode $x = v$ (suppose v from $D = \{1,2, \dots, d\}$)
  - Toss a coin with bias p
  - If it is head, report the true value $y = x$
  - Otherwise, report any other value with probability $q = \frac{1-p}{d-1}$ (uniformly at random)
    - $p = \frac{e^{\varepsilon}}{e^{\varepsilon}+d-1}, q = \frac{1}{e^{\varepsilon}+d-1} \Rightarrow \frac{\Pr[P(v)=v]}{\Pr[P(v\prime)=v]} = \frac{p}{q} = e^{\varepsilon}$
- Aggregator:
  - Suppose n_v users possess value v, I_v is the number of reports on v.
  - $E[I_v] = n_v \cdot p + (n - n_v) \cdot q$
  - Unbiased Estimation: $c(v) = \frac{I_v - n \cdot q}{p-q}$

Direct Encoding (Random Response)

- User:
  - Encode $x = v$ (suppose v from $D = \{1,2,\ldots,d\}$)
  - Toss a coin with bias p
  - If it is head, report the true value $y = x$
  - Otherwise, … ly at random)

Intuitively, the higher p, the more accurate

    - $p = \frac{e^{\varepsilon}}{e^{\varepsilon}+d-1}, q = \frac{1}{e^{\varepsilon}+d-1} \Rightarrow \frac{\Pr[P(v)=v]}{\Pr[P(v\prime)=v]} = \frac{p}{q} = e^{\varepsilon}$

- Aggregator:

However, when d is large, p becomes small

  - Suppose n_v users possess value v, I_v is the number of reports on v.
  - $E[I_v] = n_v \cdot p + (n - n_v) \cdot q$
  - Unbiased Estimation: $c(v) = \frac{I_v - n \cdot q}{p-q}$

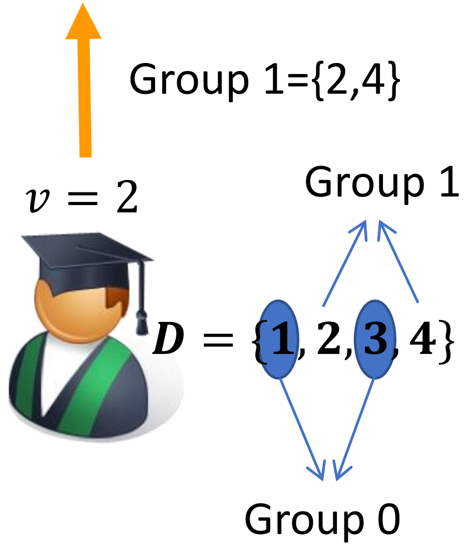
Unary Encoding (Basic RAPPOR)

- Encode the value v into a bit string $\boldsymbol{x} \coloneqq \vec{0}, \boldsymbol{x}[v] \coloneqq 1$
  - e.g., $D = \{1,2,3,4\}, v = 3,$ then $\boldsymbol{x} =$ [0,0,1,0]
- Perturb each bit independently
  - $p = \frac{e^{\varepsilon/2}}{e^{\varepsilon/2}+1}, q = \frac{1}{e^{\varepsilon/2}+1} \Rightarrow \frac{\Pr[P(E(v))=\boldsymbol{x}]}{\Pr[P(E(v'))=\boldsymbol{x}]} = \frac{\prod_i \Pr[\boldsymbol{x}[i]|v]}{\prod_i \Pr[\boldsymbol{x}[i]|v\prime]} = \frac{p\cdot(1-q)}{q\cdot(1-p)} = e^{\varepsilon}$
    - Since $\boldsymbol{x}$ is unary encoding of v, $\boldsymbol{x}$ and $\boldsymbol{x}'$ differ in two locations
- Intuition:
  - By unary encoding, each location can only be 0 or 1, effectively reducing d in each location to 2.
  - When d is large, UE is better.
- To estimate frequency of each value, do it for each bit.

Binary Local Hash

- The protocol description itself is more complicated
- Now we describe a simpler equivalent
- Each user uses a random hash function from D to $\{0,1\}$
- The user then perturbs the bit with probabilities
  - $p = \frac{e^{\varepsilon}}{e^{\varepsilon}+g-1} = \frac{e^{\varepsilon}}{e^{\varepsilon}+1}, q = \frac{1}{e^{\varepsilon}+g-1} = \frac{1}{e^{\varepsilon}+1} \Rightarrow \frac{\Pr[P(E(\boldsymbol{v}))=b]}{\Pr[P(E(\boldsymbol{v'}))=b]} = \frac{p}{q} = e^{\varepsilon}$
- The user then reports the bit and the hash function
- The aggregator increments the reported group
- $E[I_v] = n_v \cdot p + (n - n_v) \cdot (\frac{1}{2}q + \frac{1}{2}p)$
- Unbiased Estimation: $c(v) = \frac{I_v - n \cdot \frac{1}{2}}{p - \frac{1}{2}}$

$+1 \quad +1$
$[\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}]$



Takeaway

- Key Question:
  - Maximize utility of frequency estimation under LDP
- Key Idea:
  - A framework to generalize and optimize these protocols
- Results:
  - Optimized Unary Encoding and Local Hash
  - By improving the frequency estimator, results in other more complicated settings that use LDP can be improved, e.g., private learning, frequent itemset mining, etc.

Method

- We measure utility of a mechanism by its variance
  - E.g., in Random Response, $Var[c(v)] = Var\left[\frac{I_v - n\cdot q}{p-q}\right] = \frac{Var[I_v]}{(p-q)^2} \approx \frac{n\cdot q\cdot(1-q)}{(p-q)^2}$
- We propose a framework called 'pure' and cast existing mechanisms into the framework.
  - For each output y, define a set of input v called Support
    - Intuition: each output votes for a set of input
    - After perturbation, output y will support input from Support(y)
    - E.g., In BLH, Support(y) = $\{v|H(v) = y\}$
  - Define p' and q' such that $P(E(v))$ support v w/p p' and don't w/p q'
    - E.g., In Random Response, $p' = p$, $q' = q$
- Pure means this holds for all input-output pairs

Method

- We measure utility of a mechanism by its variance
  - E.g., in Random Response, $Var[c(v)] = Var\left[\frac{I_v - n\cdot q}{p-q}\right] = \frac{Var[I_v]}{(p-q)^2} \approx \frac{n\cdot q\cdot(1-q)}{(p-q)^2}$
- We propose ... anisms into the fra...
  - For each ...
    - Intuitio...
    - After p...
    - E.g., In ...
  - Define p' and q' such that $P(E(v))$ support v w/p p' and don't w/p q'
    - E.g., In Random Response, $p' = p, q' = q$
- Pure means this holds for all input-output pairs

$$min_{q'} Var[c(v)]$$

or $min_{q'} \frac{n\cdot q'\cdot(1-q')}{(p'-q)'^2}$

where p', q' satisfy ε-LDP

Optimized UE

- In the original UE, each bit is perturbed independently
  - $p = \frac{e^{\varepsilon/2}}{e^{\varepsilon/2}+1}, q = \frac{1}{e^{\varepsilon/2}+1}$
- We want to make p higher.
- **Key Insight:** We perturb 0 and 1 differently!
- There are more 0, so we perturb with greater p; there is a single 1, so we perturb with smaller p
  - For bit 0: $p_0 = \frac{e^{\varepsilon}}{e^{\varepsilon}+1}, q_0 = \frac{1}{e^{\varepsilon}+1}$
  - For bit 1: $p_1 = \frac{1}{2}, q_1 = \frac{1}{2}$
    - $\Rightarrow \frac{\Pr[P(E(v))=\boldsymbol{x}]}{\Pr[P(E(v'))=\boldsymbol{x}]} = \frac{\prod_i \Pr[\boldsymbol{x}[i]|v]}{\prod_i \Pr[\boldsymbol{x}[i]|v\prime]} = \frac{p_0 \cdot (1-q_0)}{q_0 \cdot (1-p_0)} = e^{\varepsilon}$

Optimized Local Hash (OLH)

- In original BLH, secret is **compressed** into a bit, **perturbed** and transmitted.
  - $p = \frac{e^{\varepsilon}}{e^{\varepsilon}+g-1}, q = \frac{1}{e^{\varepsilon}+g-1} \Rightarrow \frac{\Pr[P(E(v))=\boldsymbol{x}]}{\Pr[P(E(v'))=\boldsymbol{x}]} = \frac{p}{q} = e^{\varepsilon}$ ($g = 2$ groups)
- Two steps that cause information loss:
  - Compressing: loses much
  - Perturbation: pretty accurate
- **Key Insight**: We want to make a balance between the two steps:
  - By compressing into more groups, the first step carries more information
- Variance is optimized when $g = e^{\varepsilon} + 1$
- Read our paper for details!

Comparison of Different Mechanisms

	DE	SHE	THE ($\theta = 1$)	SUE	OUE	BLH	OLH
Communication Cost	$O(\log d)$	$O(d)$	$O(d)$	$O(d)$	$O(d)$	$O(\log n)$	$O(\log n)$
$\mathrm{Var}[\tilde{c}(i)]/n$	$\frac{d-2+e^{\varepsilon}}{(e^{\varepsilon}-1)^2}$	$\frac{8}{\varepsilon^2}$	$\frac{2e^{\varepsilon/2}-1}{(e^{\varepsilon/2}-1)^2}$	$\frac{e^{\varepsilon/2}}{(e^{\varepsilon/2}-1)^2}$	$\frac{4e^{\varepsilon}}{(e^{\varepsilon}-1)^2}$	$\frac{(e^{\varepsilon}+1)^2}{(e^{\varepsilon}-1)^2}$	$\frac{4e^{\varepsilon}}{(e^{\varepsilon}-1)^2}$

Table 1: Com … d Variances for different metho…

Direct Encoding has greater variance with larger d

OUE and OLH have the same variance
But OLH has smaller communication cost

Conclusion

- We survey existing LDP protocols on frequency estimation
- We propose a pure framework and cast existing protocols into it
- We optimize UE and BLH and come up with OUE and OLH

Limitations

- Variance is linear in n, which seems inevitable
  - Therefore, requires large number of users
- Cannot handle large domains

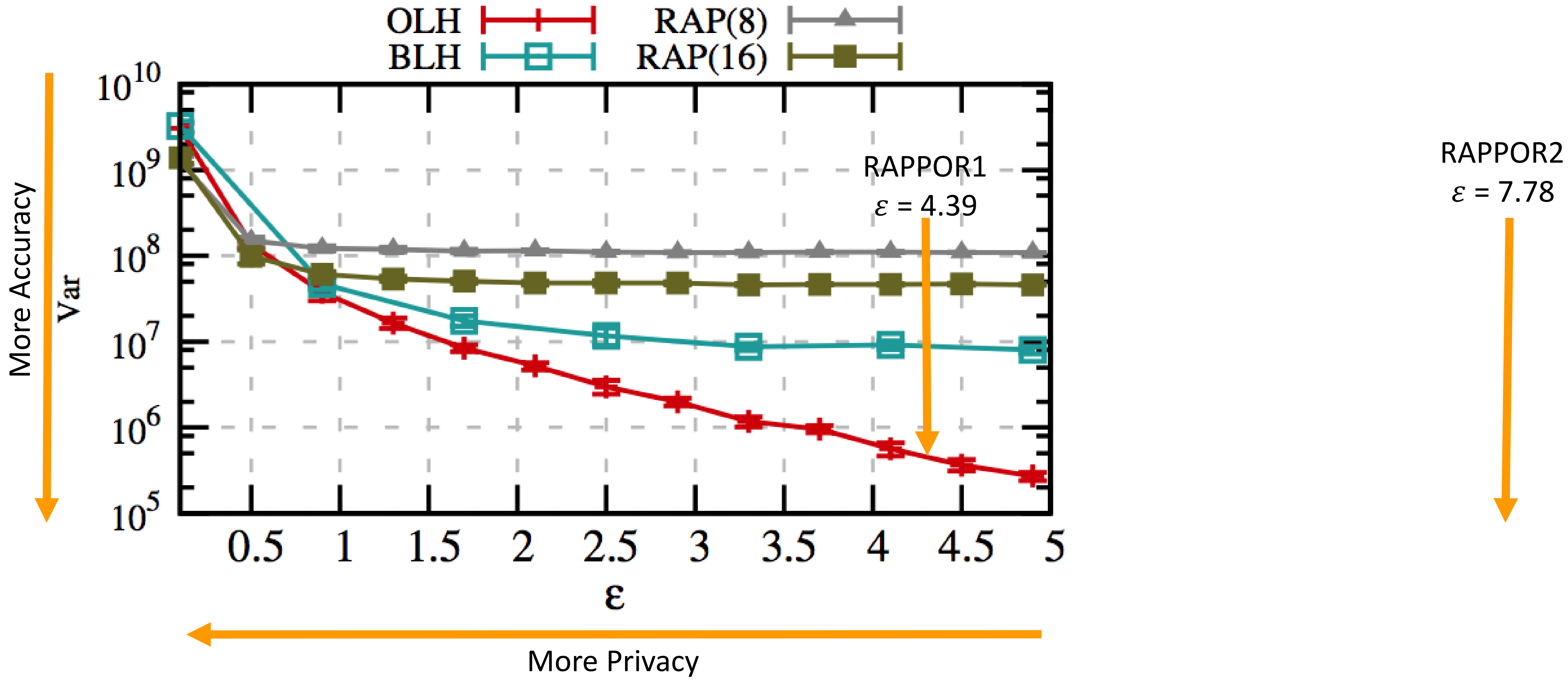
Future Work

- Handling large domains
- Handling set-values

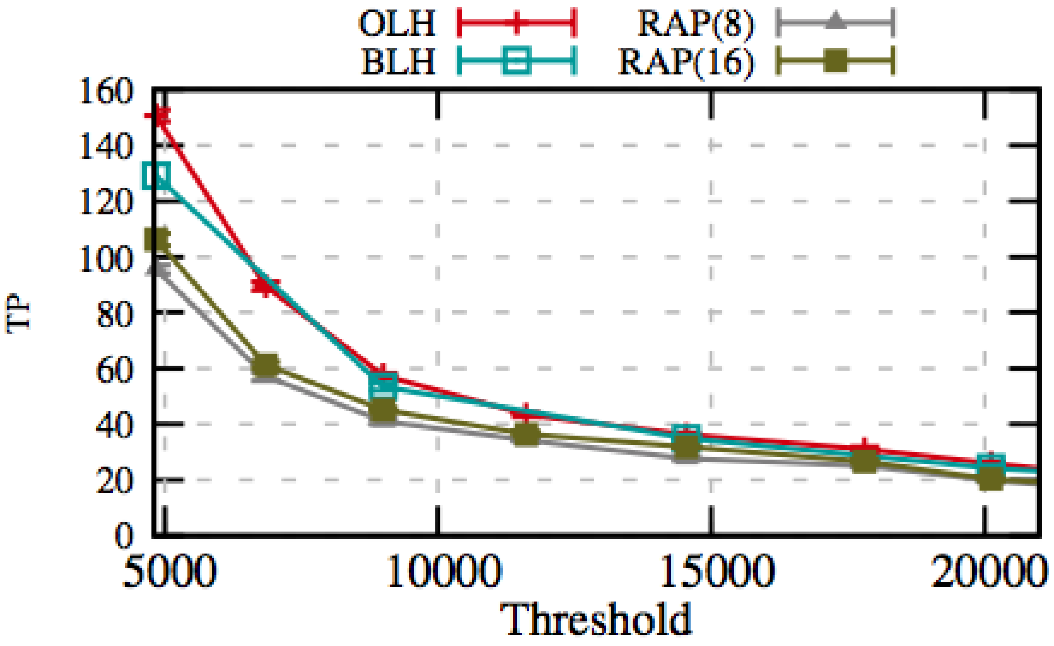
Backup: Experiments Highlights

- Dataset: Kosarak dataset
  - (also on Rockyou dataset and a Synthetic dataset)
- Competitors: RAPPOR, BLH, OLH
  - Randomized Response is not compared because the domain is large
- Key Results:
  - OLH performs magnitudes better, especially when ε is large
  - This also confirms our analytical conclusion

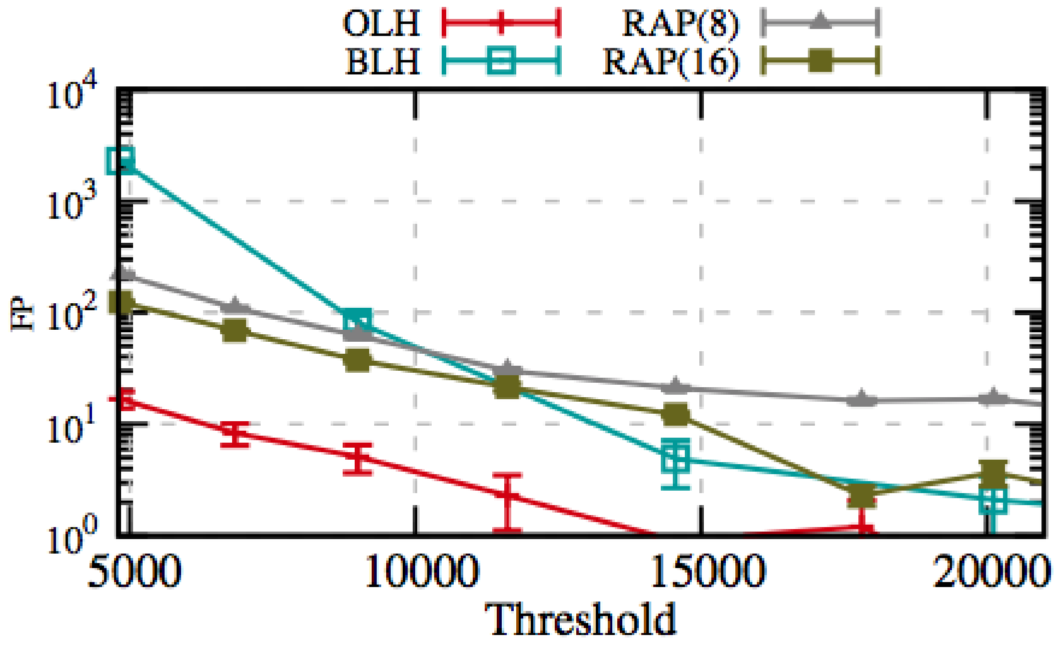
Backup: Accuracy on Frequent Values



Backup: On Information Quality



(a) Number of True Positives



(b) Number of False Positives

Backup: On answering multiple questions

- Previously works (including centralized DP) suggest splitting privacy budget
- For example, when a user answers two questions, privacy budgets are $\varepsilon/2$ and $\varepsilon/2$ (assuming the two questions are of equal importance)
- In the centralized setting, there are sequential composition and parallel composition
  - By partitioning users, one uses to parallel composition
  - By split privacy budget, one uses sequential composition
  - The two can basically produce equivalent results
- What about the local setting?

Backup: On answering multiple questions

- Measure the frequency accuracy (normalize since two approach have different number of users)
  - Assuming OLH is used: $Var[c(v)/n] = \frac{q \cdot (1-q)}{n \cdot (p-q)^2} = \frac{4e^{\varepsilon}}{n \cdot (e^{\varepsilon}-1)^2}$
- Two settings:
  - Split privacy budget: $Var[c(v)/n] = \frac{4e^{\varepsilon/2}}{n \cdot \left(e^{\varepsilon/2}-1\right)^2}$
  - Partition users: $Var\left[c(v)/\frac{1}{2}n\right] = \frac{8e^{\varepsilon}}{n \cdot (e^{\varepsilon}-1)^2}$ ✓
- Algebra shows that it is better to partition users

Thanks to my coauthors

// | ?