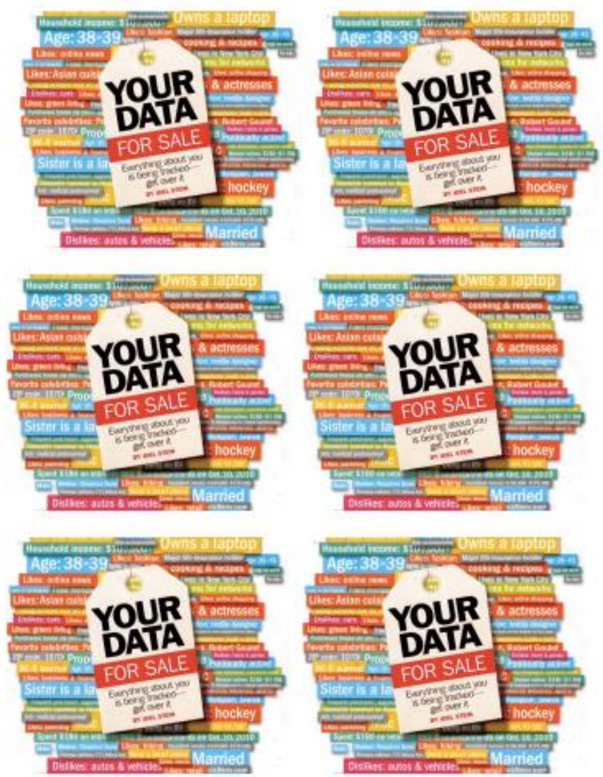
Data Anonymization and Differential Privacy

Dr. Chen Zhang

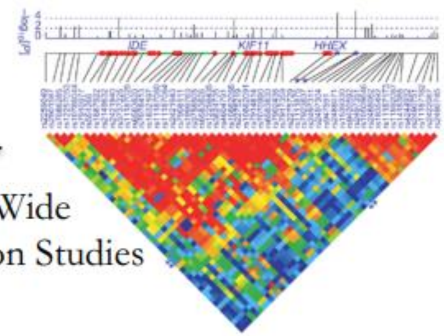
Department of Computer Science
The Hang Seng University of Hong Kong

Aggregated Personal Data is **Invaluable**



Advertising

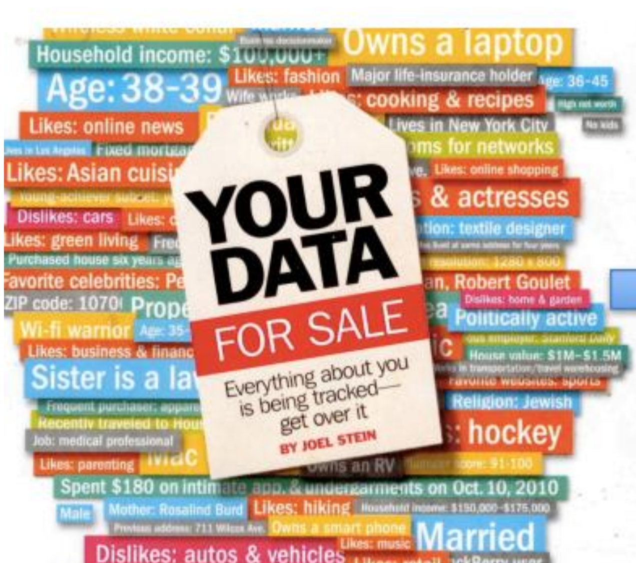
Source (esri.com)



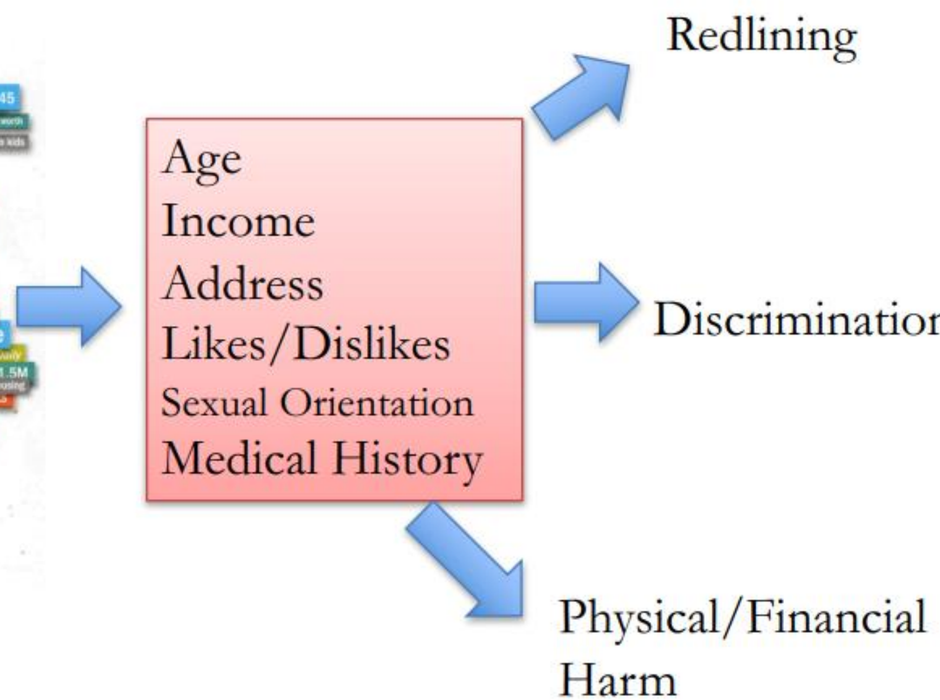
Genome Wide Association Studies

Human Mobility analysis

Personal Data is … Very … Personal!



Source (time.com)

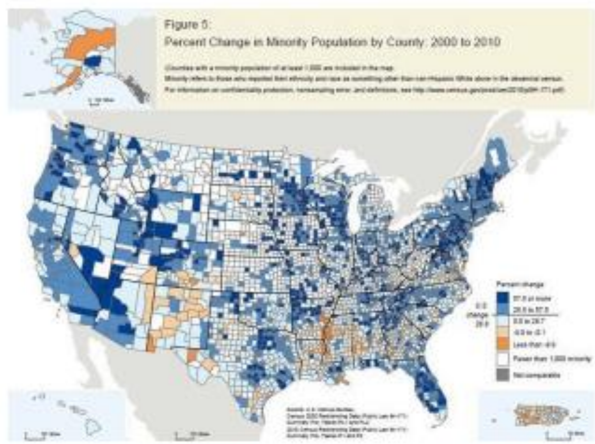


Aggregated Personal Data

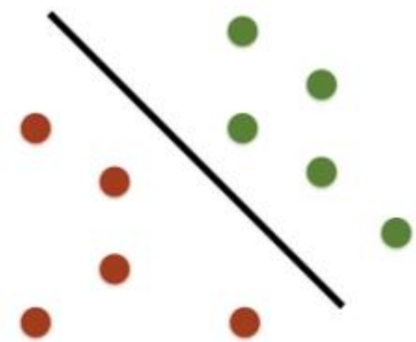
- … is made publicly available in many forms.

De-identified records
(e.g., medical)

Statistics
(e.g., demographic)



Predictive models
(e.g., advertising)



Data “Anonymization”

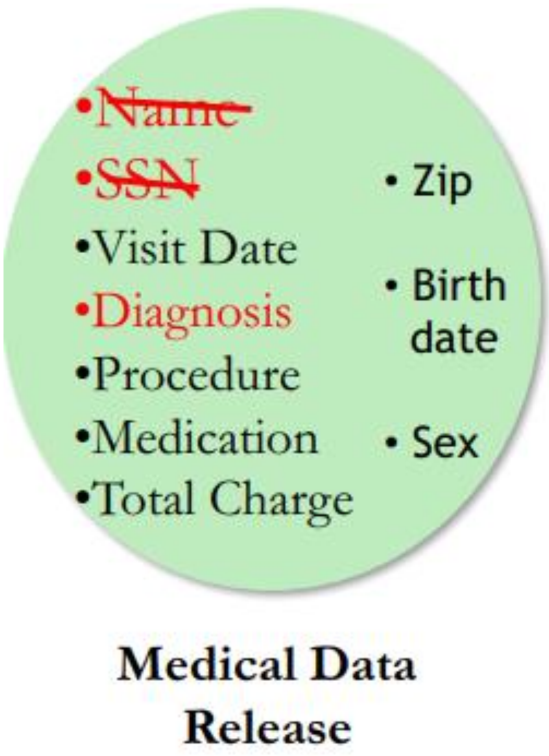
- Anonymity: the property that certain records or transactions not to be attributable to any individual.

- How?
  - Remove “personally identifying information” (PII)
    - Name, Social Security number, phone number, email, address… what else?

- Problem: PII has no technical meaning
  - In privacy breaches, any information can be personally identifying

The Massachusetts Governor Privacy Breach

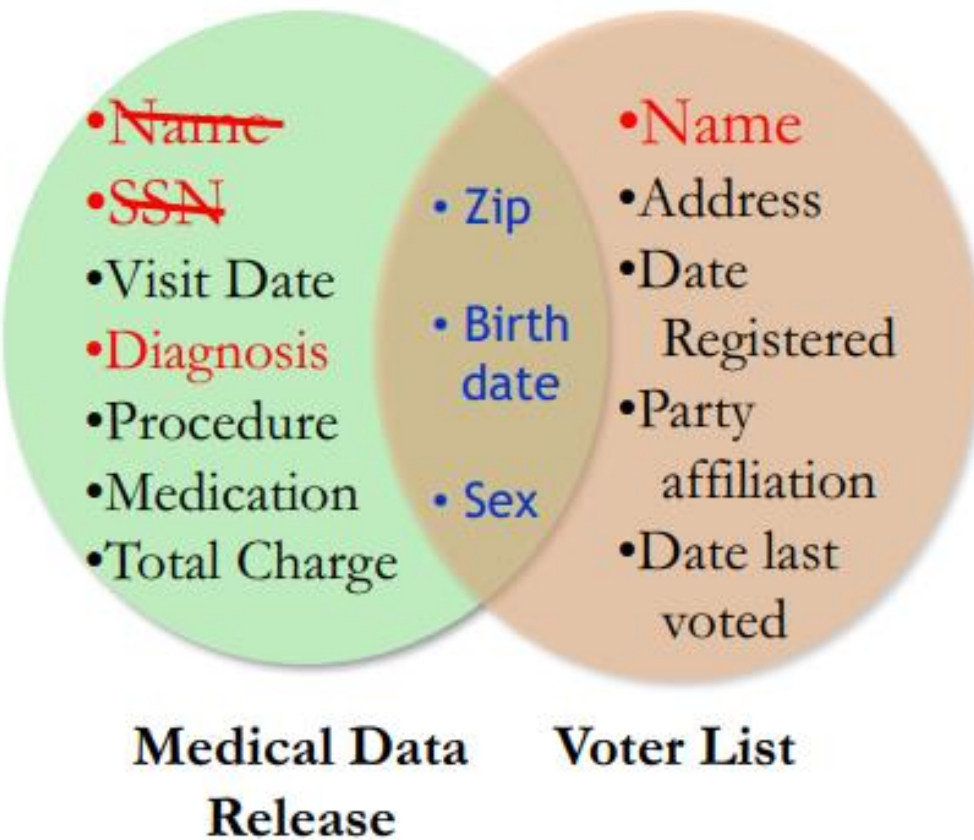
[Sweeney, 2010]



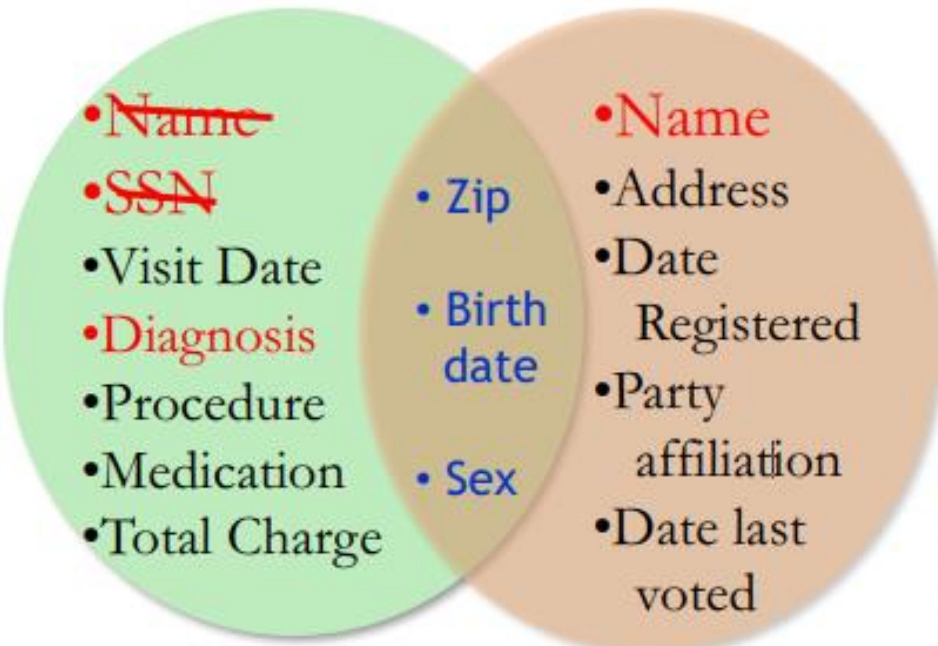
SSN: Social Security Number

The Massachusetts Governor Privacy Breach

[Sweeney, 2010]



Linkage Attack



Medical Data Release **Voter List**

[Sweeney, 2010]

- Governor of MA **uniquely identified** using ZipCode, Birth Date, and Sex.

Name linked to Diagnosis

Observation #1: Dataset Joins

- Attacker learns sensitive data by joining two datasets on common attributes
  - Anonymized dataset with sensitive attributes
    - Example: age, race, symptoms
  - “Harmless” dataset with individual identifiers
    - Example: name, address, age, race

- Demographic attributes (age, ZIP code, race, etc.) are common in datasets with information about individuals

Observation #2: Quasi-Identifiers

- Quasi-identifiers are pieces of information that are not of themselves unique identifiers, but are sufficiently well correlated with an entity that they can be combined with other quasi-identifiers to create a unique identifier.
- Sweeney's observation: (birthdate, ZIP code, gender) uniquely identifies more than 60% of US population
- Publishing a record with a quasi-identifier is as bad as publishing it with an explicit identity
- Eliminating quasi-identifiers is not desirable
  - For example, users of the dataset may want to study distribution of diseases by age and ZIP code

quasi-identifiers

Race	Age	Symptoms	Blood type	Medical history
…	…	…	…	
…	…	…	…	…

sensitive attributes

Anonymization in a Nutshell

- Dataset is a relational table
- Attributes (columns) are divided into quasi-identifiers and sensitive attributes

Race	Age	Symptoms	Blood type	Medical history
…	…	…	…	
…	…	…	…	…

quasi-identifiers

sensitive attributes

- Generalize/suppress quasi-identifiers, don't touch sensitive attributes (keep them "truthful")

k-Anonymity

- Proposed by Samarati and Sweeney (1998)
- Definition: Each (transformed) quasi-identifier group must appear in at least k records in the anonymized dataset
  - k is chosen by the data owner
  - Example: any age-race combination from original DB must appear at least 10 times in anonymized DB
- Guarantees that any join on quasi-identifiers with the anonymized dataset will contain at least k records for each quasi-identifier

Achieving k-Anonymity

Most designs based on generalization and suppression

- Generalization
  - Individual values of attributes replaced by **broader category**
    - Area code instead of phone number: 3442 8765 -- >> 3442 xxxx
    - Value "23" of the age attribute is replaced by 20<Age<=30

- Suppression
  - Replace certain values of the attributes by an asterisk '*' (not releasing a value at all.)
    - Example: replace all the values in the 'Name' attribute with a '*'.

Example: 3-Anonymity

Caucas	78712	Flu
Asian	78705	Shingles
Caucas	78754	Flu
Asian	78705	Acne
AfrAm	78705	Acne
Caucas	78705	Flu

Caucas	787XX	Flu
Asian/AfrAm	78705	Shingles
Caucas	787XX	Flu
Asian/AfrAm	78705	Acne
Asian/AfrAm	78705	Acne
Caucas	787XX	Flu

This is 3-anonymous, right?

Problem of k-Anonymity

When joining with external database ,
adversary learns Rusty Shackleford has Flu

…	…	…
Rusty Shackleford	Caucas	78705
…	…	…

+

Caucas	787XX	Flu
Asian/AfrAm	78705	Shingles
Caucas	787XX	Flu
Asian/AfrAm	78705	Acne
Asian/AfrAm	78705	Acne
Caucas	787XX	Flu

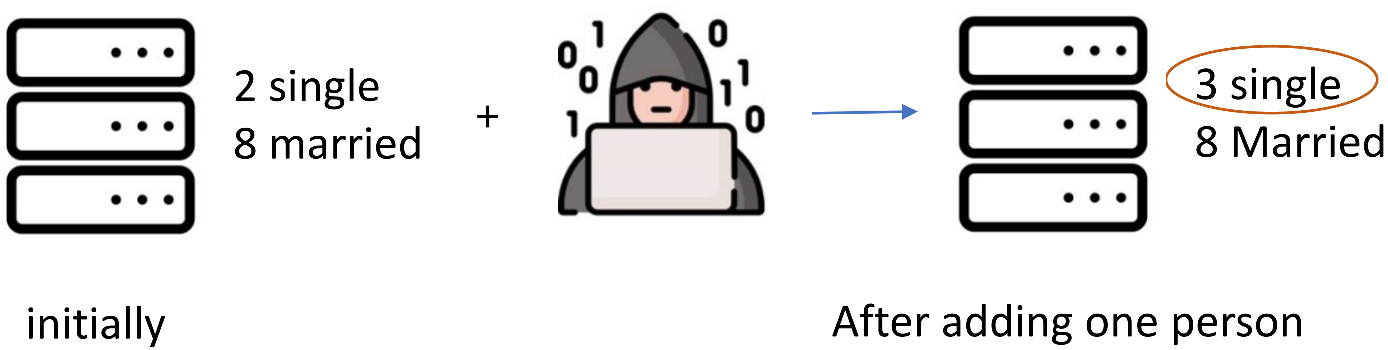
Problem: sensitive attributes are not "diverse"
within each quasi-identifier group

Other Attempts

- *L*-diversity
  - Entropy of sensitive attributes within each quasi-identifier group must be at least *L*
- *t*-closeness
  - Distribution of sensitive attributes within each quasi-identifier group should be “close” to their distribution in the entire original database

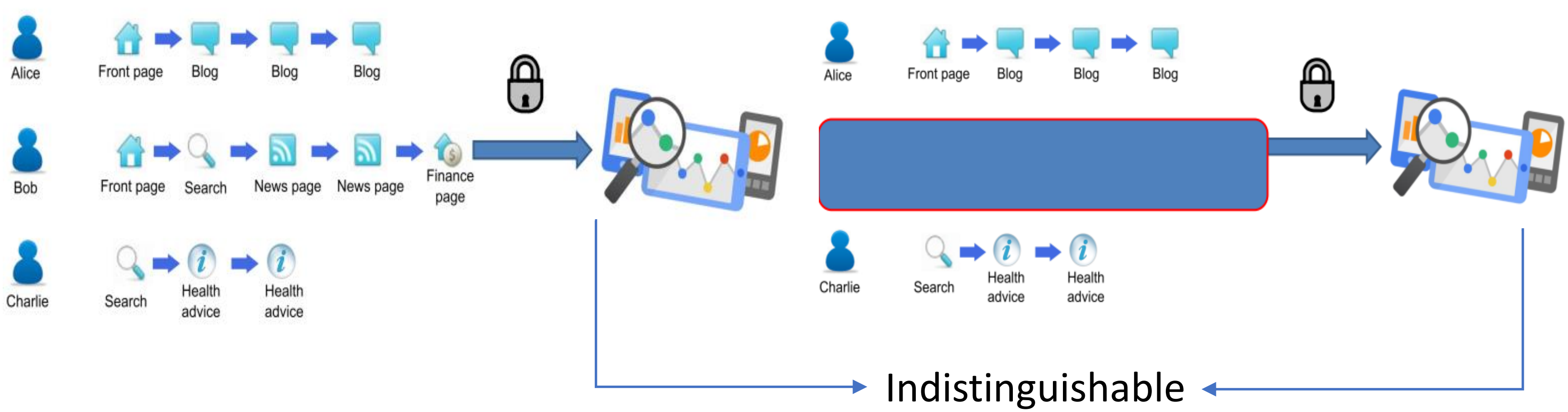
Differential Attacks

- Compares the variations in the input with variations in the encrypted output to find the desired key or plaintext message.
- Example:

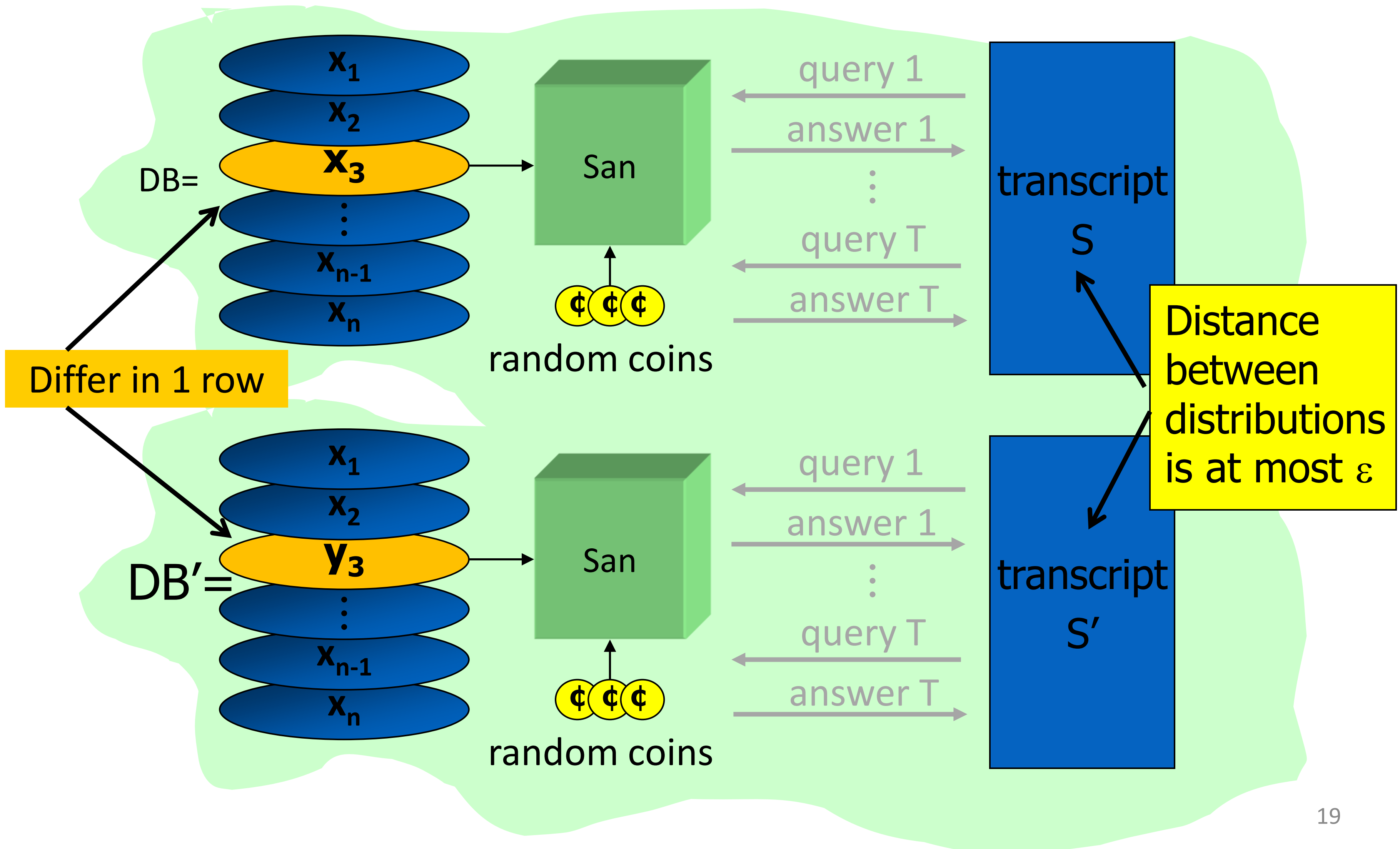


Differential Privacy

- Statistical outcome is indistinguishable regardless of **whether a particular user record is in the data or not**.
  - "Whatever is learned would be learned regardless of whether or not you participate".

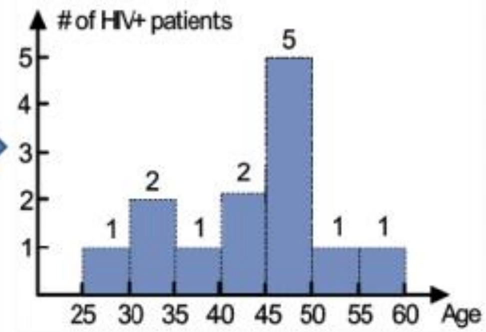


Indistinguishability

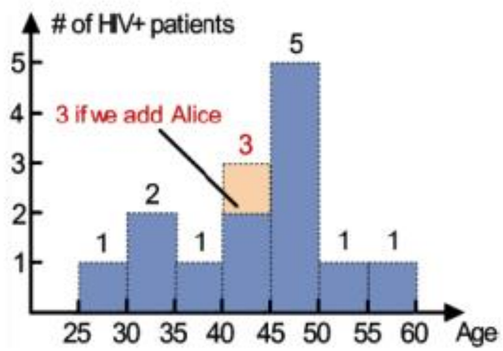


An Example: Statistical Data Release

Name	Age	HIV+
Frank	42	Y
Bob	31	Y
Mary	28	Y
Dave	43	N
...	...	...



Name	Age	HIV+
Alice	43	Y
Frank	42	Y
Bob	31	Y
Mary	28	Y
Dave	43	N
...	...	...

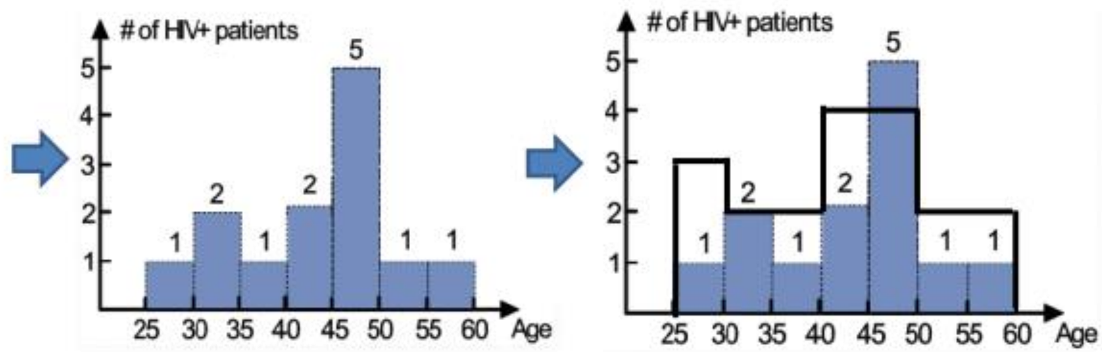


Original records

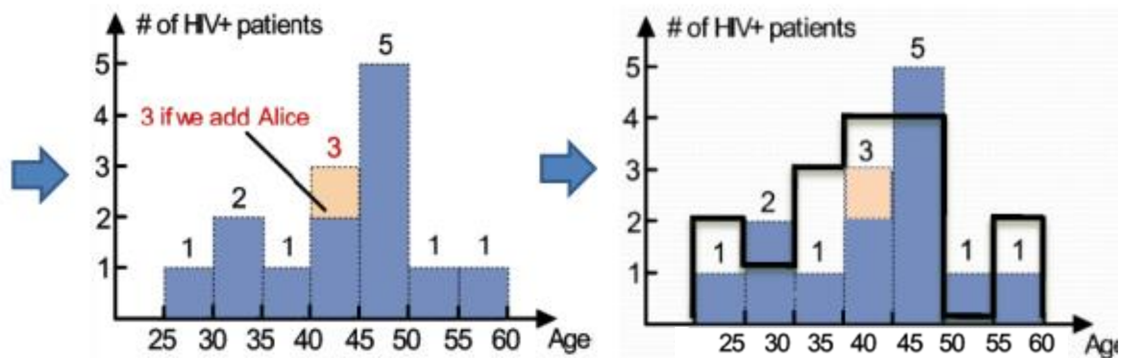
Original histogram

An Example: Statistical Data Release

Name	Age	HIV+
Frank	42	Y
Bob	31	Y
Mary	28	Y
Dave	43	N
…	…	…



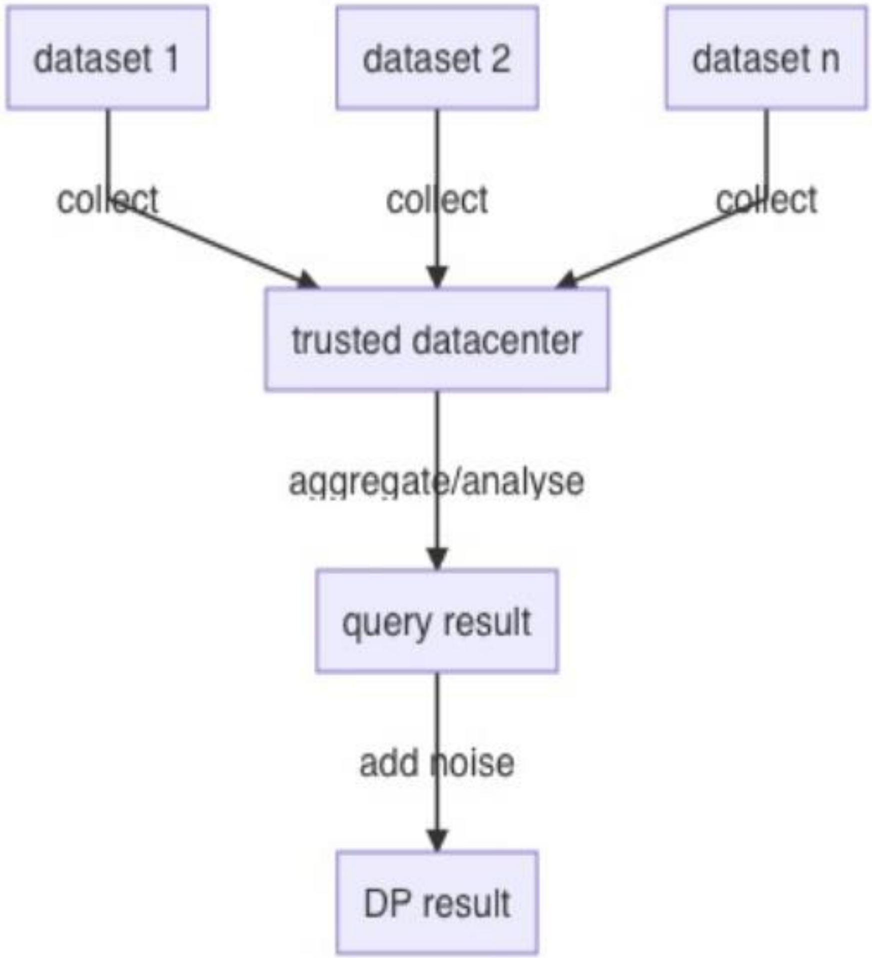
Name	Age	HIV+
Alice	43	Y
Frank	42	Y
Bob	31	Y
Mary	28	Y
Dave	43	N
…	…	…



Original records **Original histogram** **Perturbed histogram with differential privacy**

Framework of DP

users



Formalizing Indistinguishability

[Dwork, ICALP'06]

For every pair of **neighboring databases** that differ in only one record

For every output

If algorithm A satisfies differential privacy then

$$\frac{\Pr[A(D_1) = O]}{\Pr[A(D_2) = O]} < \exp(\varepsilon) \quad (\varepsilon > 0)$$

Intuition: adversary should not be able to use output O to distinguish between any D_1 and D_2

- A is a <u>randomized algorithm</u> that takes a dataset as input (representing the actions of the trusted party holding the data).

Privacy Budget ε

For every pair of neighboring databases that differ in only one record

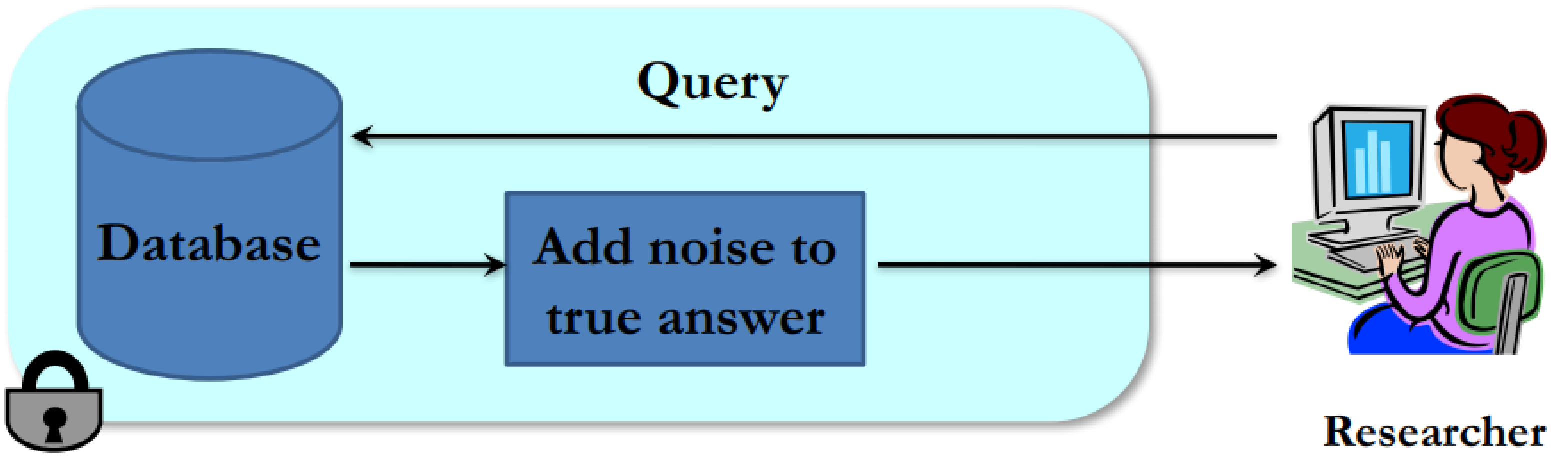
For every output

D_1 D_2 O

$$\Pr[A(D_1) = O] \le e^{\varepsilon} \Pr[A(D_2) = O]$$

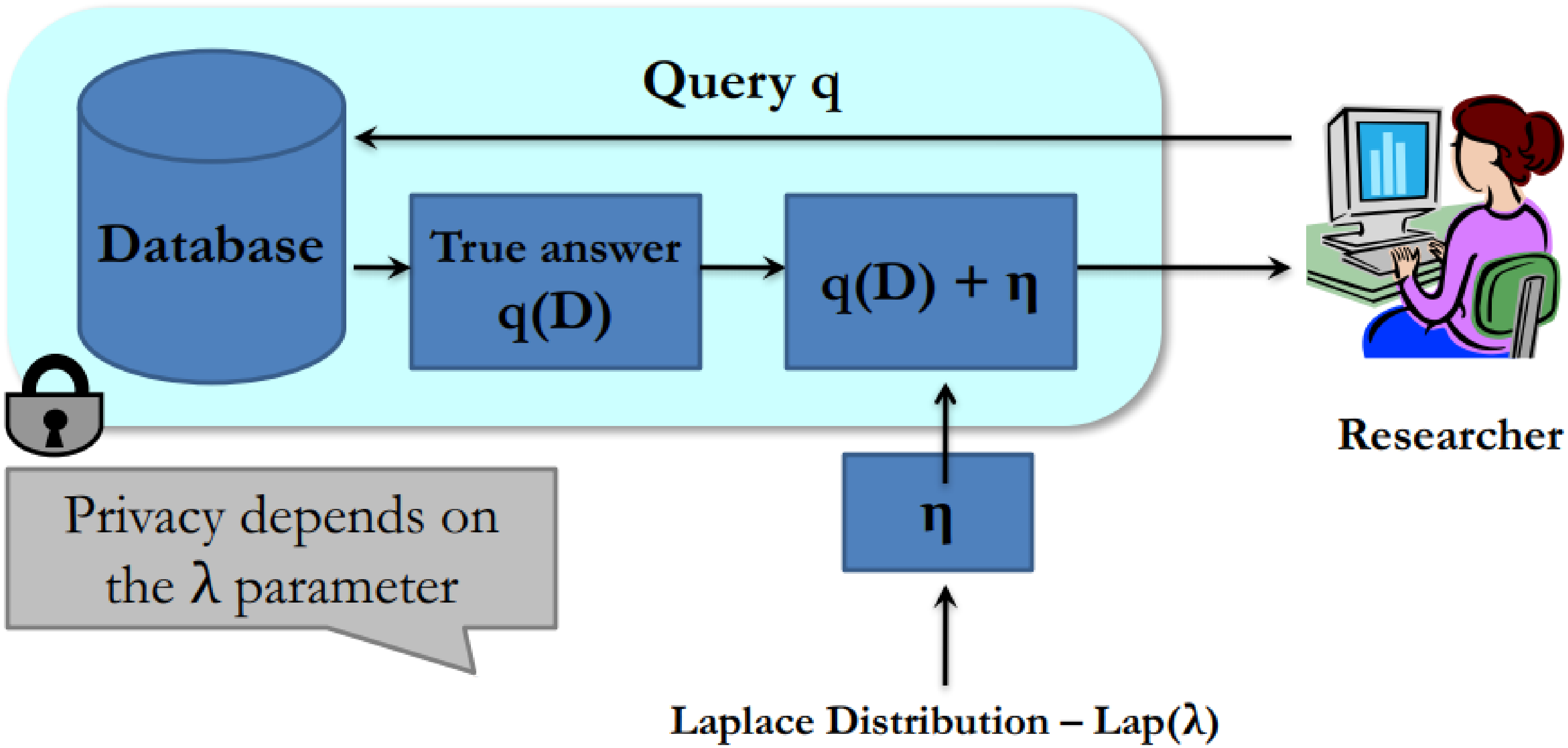
Controls the degree to which D_1 and D_2 can be distinguished.
Smaller ε gives more privacy (and worse utility)

Output Randomization



- Add noise to answers such that:
  - Each answer does not leak too much information about the database.
  - Noisy answers are close to the original answers.

Laplace Mechanism



$\lambda = \frac{s}{\varepsilon}$, where ε is privacy budget and s is sensitivity

The sensitivity of a function reflects the amount the function's output will change when its input changes.

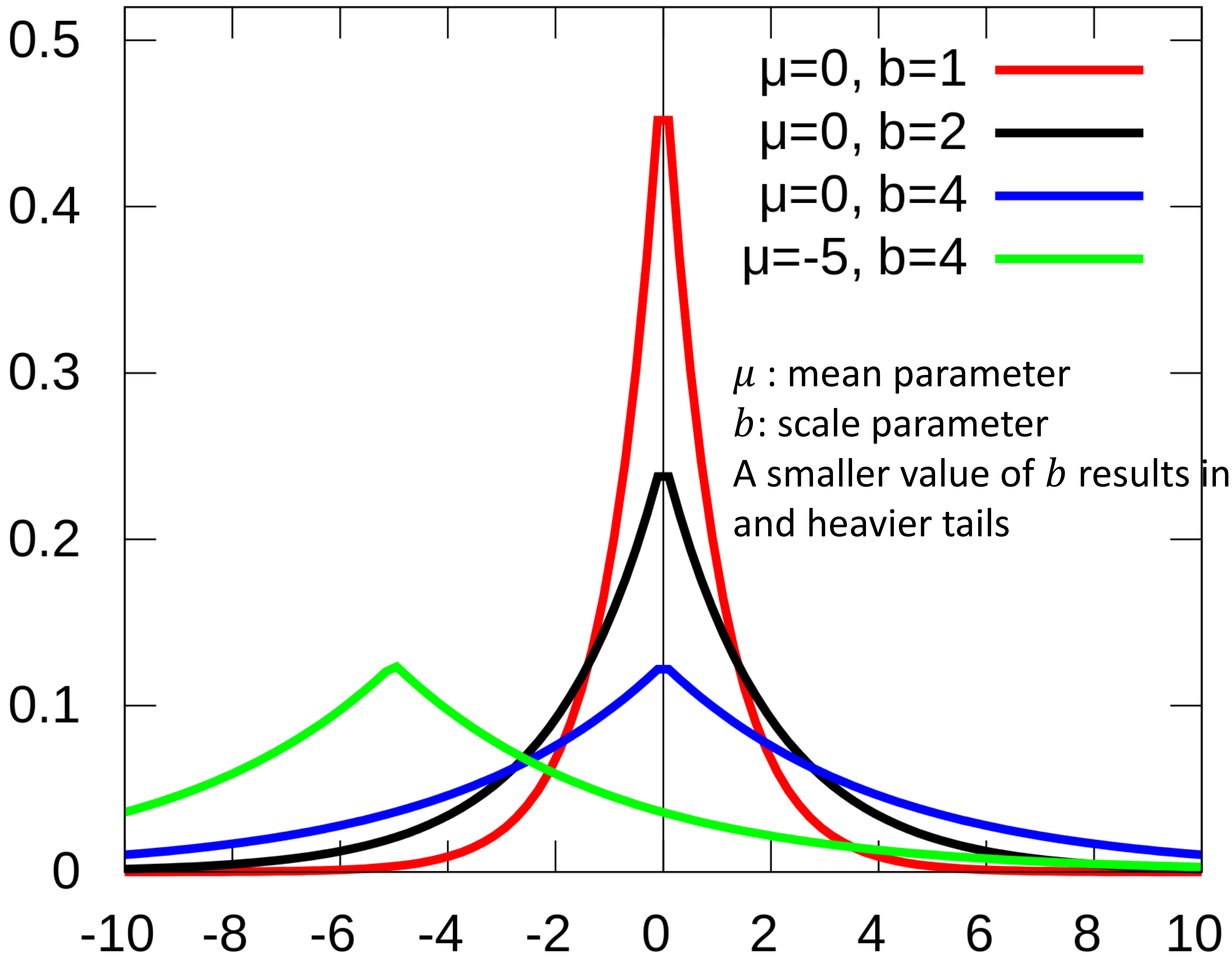
A random variable has a $\text{Laplace}(\mu, b)$ distribution if its probability density function is

$$f(x \mid \mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right)$$

$$= \frac{1}{2b} \begin{cases} \exp\left(-\frac{\mu-x}{b}\right) & \text{if } x < \mu \\ \exp\left(-\frac{x-\mu}{b}\right) & \text{if } x \geq \mu \end{cases}$$

Here, μ is a location parameter and $b > 0$, which is sometimes referred to as the diversity, is a scale parameter. If $\mu = 0$ and $b = 1$, the positive half-line is exactly an exponential distribution scaled by 1/2.

https://en.wikipedia.org/wiki/Laplace_distribution



μ : mean parameter

b: scale parameter

A smaller value of b results in a sharper peak and heavier tails

Typical Laplace probability density function

https://en.wikipedia.org/wiki/Laplace_distribution

Composition Theorems

Why composition?

- Reasoning about privacy of a complex algorithm is hard.

- Helps software design
  - If building blocks are proven to be private, it would be easy to reason about privacy of a complex algorithm built entirely using these building blocks.

Sequential Composition

- If M_1, M_2, ..., M_k are algorithms that access a private database D such that each M_i satisfies ε_i -differential privacy,

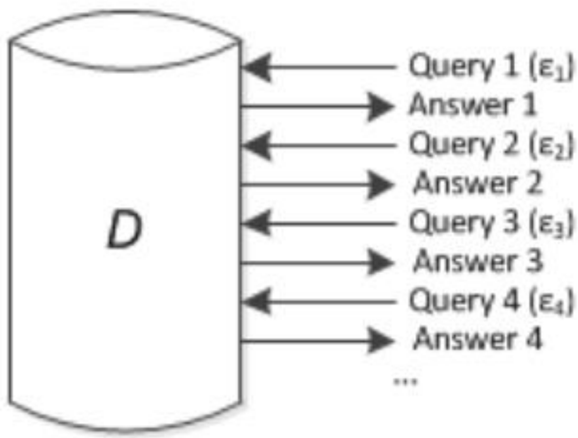
  then running all k algorithms sequentially satisfies ε-differential privacy with $\varepsilon=\varepsilon_1+\ldots+\varepsilon_k$

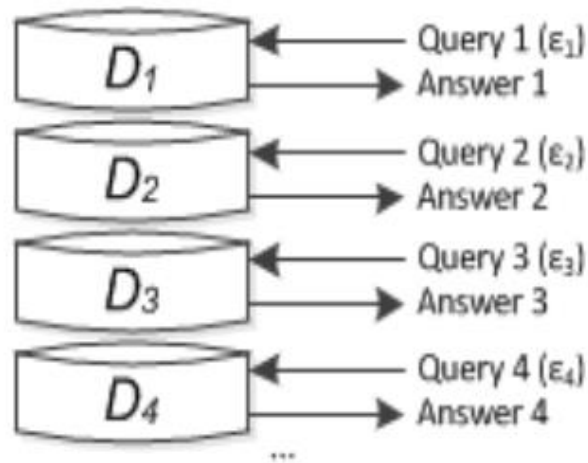
Parallel Composition

- If $M_1, M_2, \ldots, M_k$ are algorithms that access disjoint databases $D_1, D_2, \ldots, D_k$ such that each M_i satisfies ε_i -differential privacy,

  then running all k algorithms in "parallel" satisfies ε-differential privacy with $\varepsilon = \max\{\varepsilon_1,\ldots,\varepsilon_k\}$

Composition theorems





Sequential composition

$\sum_i \varepsilon_i$ –differential privacy

Parallel composition

$\max(\varepsilon_i)$–differential privacy

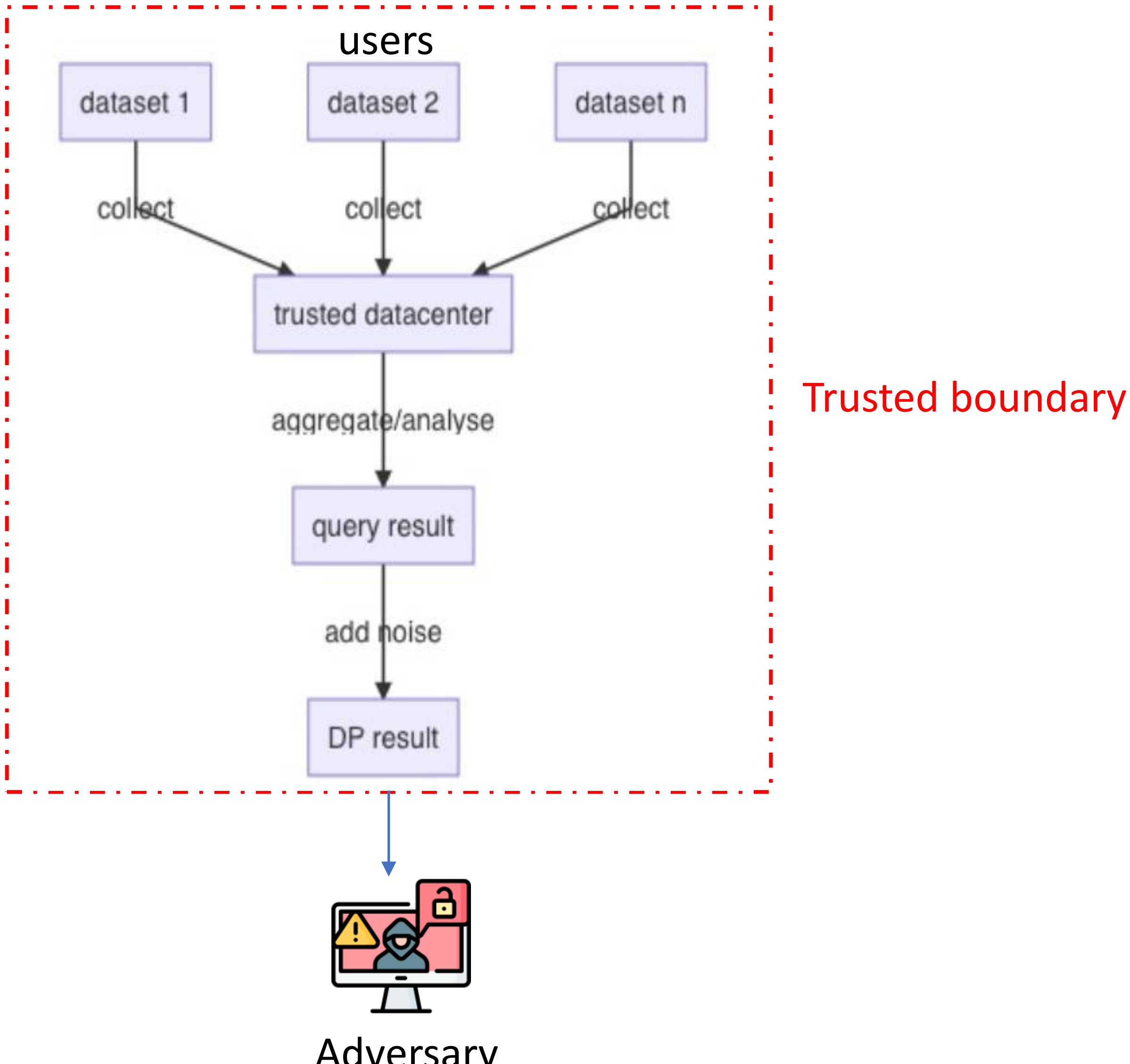
Summary

- Differential privacy ensures that an attacker can't infer the presence or absence of a single record in the input based on any output

- Basic algorithm with random perturbation
  - Laplacian mechanism

- Composition rules help build complex algorithms using building blocks

The problem of DP: Need A Trust Data Center



Trying to Reduce Trust

- Most work on differential privacy assumes a trusted party
  - Data aggregator (e.g., organizations) that sees the true, raw data
  - Can compute exact query answers, then perturb for privacy
- A reasonable question: can we reduce the amount of trust?
  - Can we remove the trusted party from the equation?
  - Users produce locally private output, aggregate to answer queries

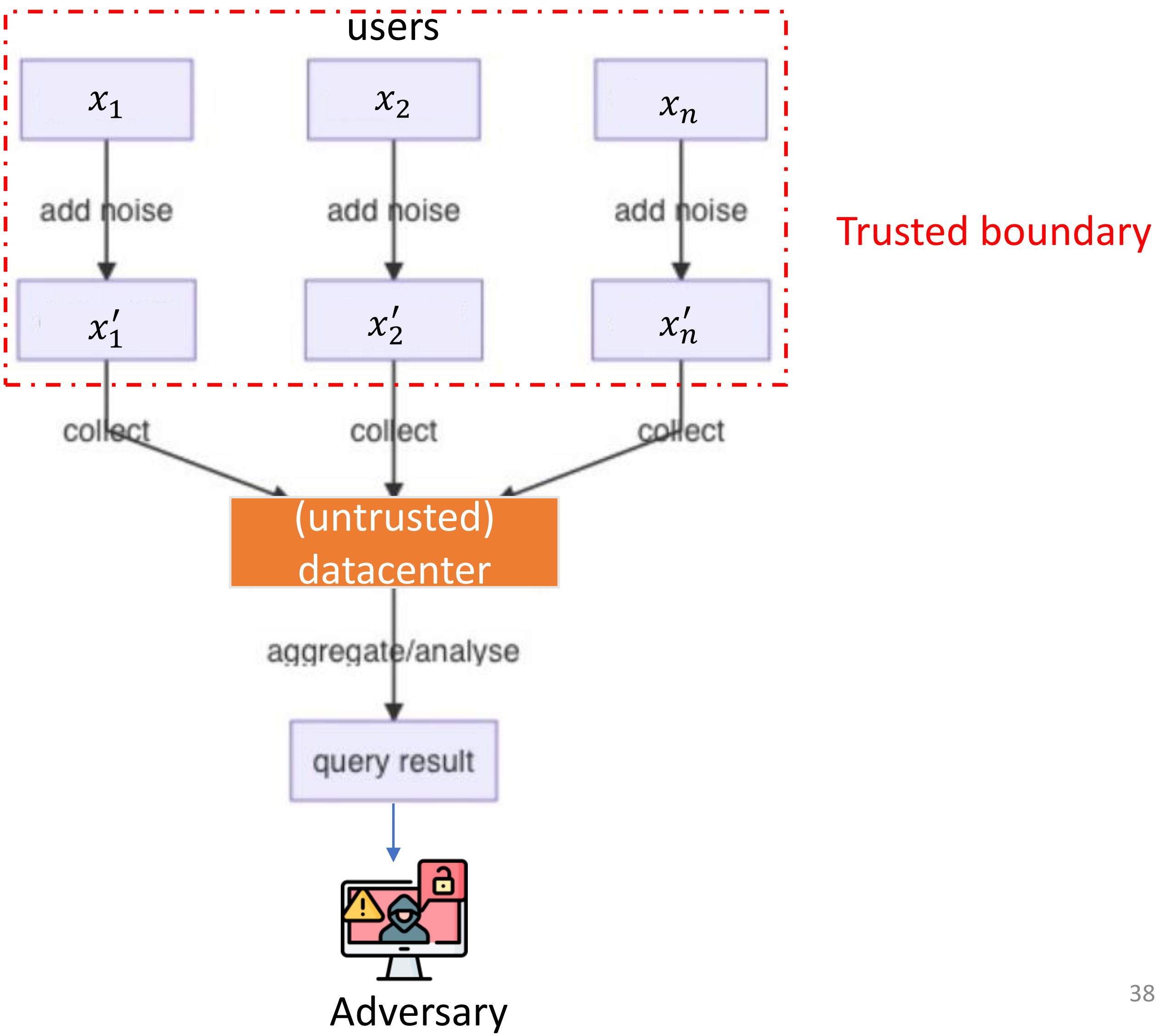
Local Differential Privacy

- How about having each user run a DP algorithm on their data?
  - Then combine all the results to get a final answer

- On first glance, this idea seems crazy
  - Each user adds noise to mask their own input
  - So surely the noise will always overwhelm the signal?

- But … noise can cancel out or be subtracted out
  - We end up with the true answer, plus noise which can be smaller

Framework of Local Differential Privacy



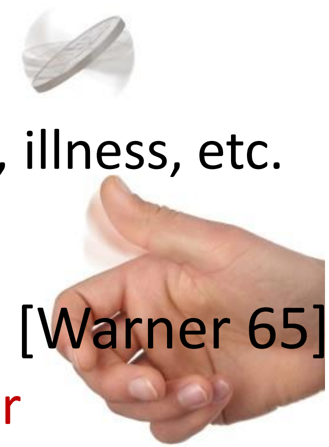
Local Differential Privacy

- We can achieve LDP, and obtain reasonable accuracy (for large N)

- Generic approach: apply centralized DP algorithm to local data
  - But error might be quite large
  - Unclear how to merge private outputs (e.g. private clustering)

- So we seek to design new LDP algorithms
  - Maximize the accuracy of the results
  - Minimize the costs to the users (space, time, communication)
  - Ensure that there is an accurate algorithm for aggregation

Privacy with A Coin Toss: Randomized Response



- Each user has a single bit of private information
  - Encoding e.g. political/sexual/religious preference, illness, etc.

- Randomize Response (RR): toss an unbiased coin [Warner 65]
  - If Heads (probability $p = ½$), report the true answer
  - Else, toss unbiased coin again: if Heads, report True, else False

- Collect responses from N users, subtract noise
  - See Differential Privacy Tutorial.ipynb
  - Generalization: allow biased coins ($p \neq ½$)

Key difference between DP and LDP

- DP concerns two neighboring datasets
- LDP concerns any two values
- As a result, the amount of noise is different: In aggregated result for counting queries
  - Noise in DP is $\Omega(1)$ (sensitivity is constant)
  - But in LDP, even noise for each user is constant, the aggregated result is $\Omega(\sqrt{n})$ [1]

[1] Optimal lower bound for differentially private multi-party aggregation by T.-H. H. Chan, E. Shi, and D. Song

The Use of Differential Privacy

- Google:
  - Chrome
  - Google Maps
  - Google assistant
  - BigQuery
  - differential privacy library developed by Google: https://github.com/google/differential-privacy
- Apple:
  - iOS e.g., Learning iconic scenes, discovering new words
  - Safari e.g., Auto-play intent analysis
- Microsoft:
  - Windows e.g., understand overall app usage
  - Advertiser queries on LinkedIn
  - Machine learning
  - https://blogs.microsoft.com/ai-for-business/differential-privacy/

Reflecting on LDP

- Local Differential Privacy is a big success for privacy research
  - Adopted by Google, Apple, Microsoft and more for deployment
  - Deployments affecting (hundreds of) millions of users
  - In contrast, centralized DP has smaller success
- However, there are reasons to pause and reflect:
  - LDP only works when you can rely on millions of active participants
  - Privacy settings are not very tight: deployed ε ranges from 0.5 to 8+