Differential Privacy

Protecting Individual Contributions to Aggregated Results

Toni Mattis

Cloud Security Mechanisms
(Prof. Polze, Christian Neuhaus, SS2013)

16. May 2013

Structure

Privacy Examples

Goal

Avoid disclosure of an *individual's contribution* to an *aggregate result*.

What does *aggregated results* mean?

- Average income of a neighbourhood
- Frequency of certain diseases among a population
- Correlation between coffee and tobacco consumption

Sensitive individual attributes:

- Income (Census)
- Diseases (Clinical Reports)
- Habits (Surveys)
- Location and Movement (Mobile Phones)

Privacy Examples

Goal

Avoid disclosure of an *individual's contribution* to an *aggregate result*.

What does *aggregated results* mean?

- Average income of a neighbourhood
- Frequency of certain diseases among a population
- Correlation between coffee and tobacco consumption

Sensitive individual attributes:

- Income (Census)
- Diseases (Clinical Reports)
- Habits (Surveys)
- Location and Movement (Mobile Phones)

Privacy Examples

Goal

Avoid disclosure of an *individual's contribution* to an *aggregate result*.

What does *aggregated results* mean?

- Average income of a neighbourhood
- Frequency of certain diseases among a population
- Correlation between coffee and tobacco consumption

Sensitive individual attributes:

- Income (Census)
- Diseases (Clinical Reports)
- Habits (Surveys)
- Location and Movement (Mobile Phones)

May 12 2013: EE data sold to track customers?

NEWS

Switch on and you become a goldmine

Market researchers snooping on mobile phones tried to sell personal data to police to track criminals and protesters

Richard Kerbaj and Jon Ungoed-Thomas Published: 12 May 2013

Comment (0) Print

Official Response from *Ipsos*:

> *We do not have access to any names, personal address information, nor postcodes or phone numbers. [...]*
> *We only ever report on aggregated* **groups of 50** *or more customers. [...]*
> **We will never release any data that in any way allows an individual to be identified.**

May 12 2013: EE data sold to track customers?

NEWS

Switch on and you become a goldmine

Market researchers snooping on mobile phones tried to sell personal data to police to track criminals and protesters

Richard Kerbaj and Jon Ungoed-Thomas Published: 12 May 2013

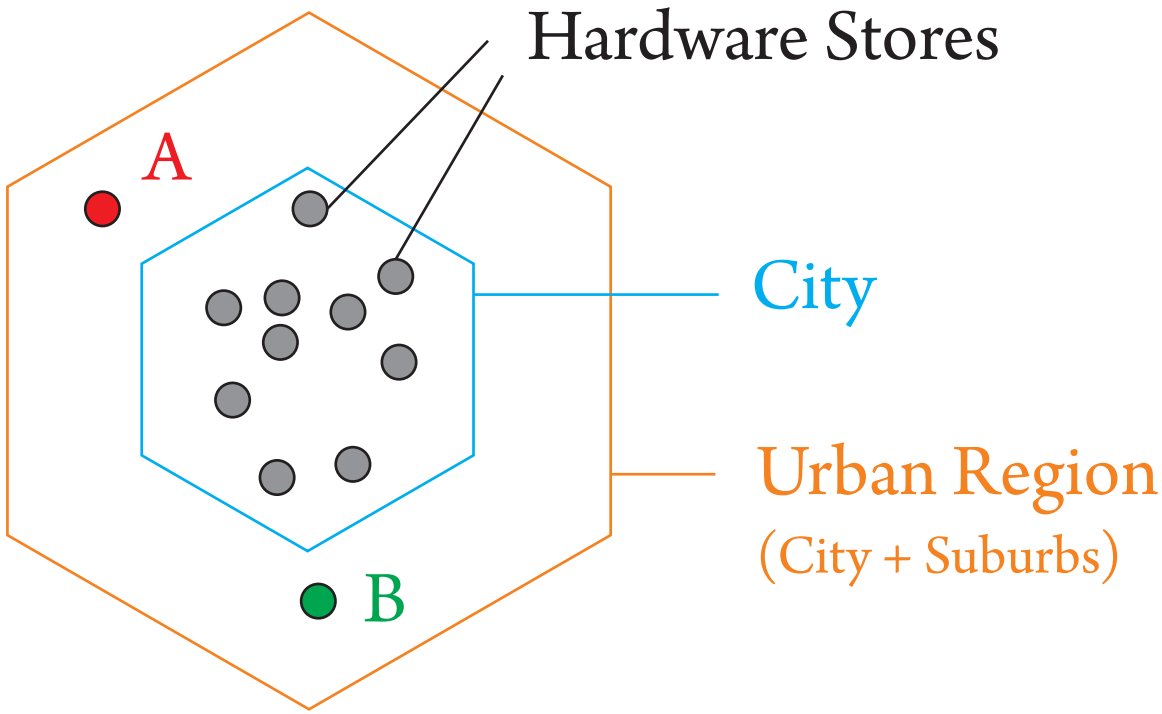
Comment (0) Print

Official Response from *Ipsos*:

> *We do not have access to any names, personal address information, nor postcodes or phone numbers. [...]*
> *We only ever report on aggregated* **groups of 50** *or more customers. [...]*
> **We will never release any data that in any way allows an individual to be identified**.

The Inference Problem

A tax officer requests the **total revenue** of the *City* at the Census Bureau. May the Census Bureau release the total revenue of the *Region* afterwards?[Han71]

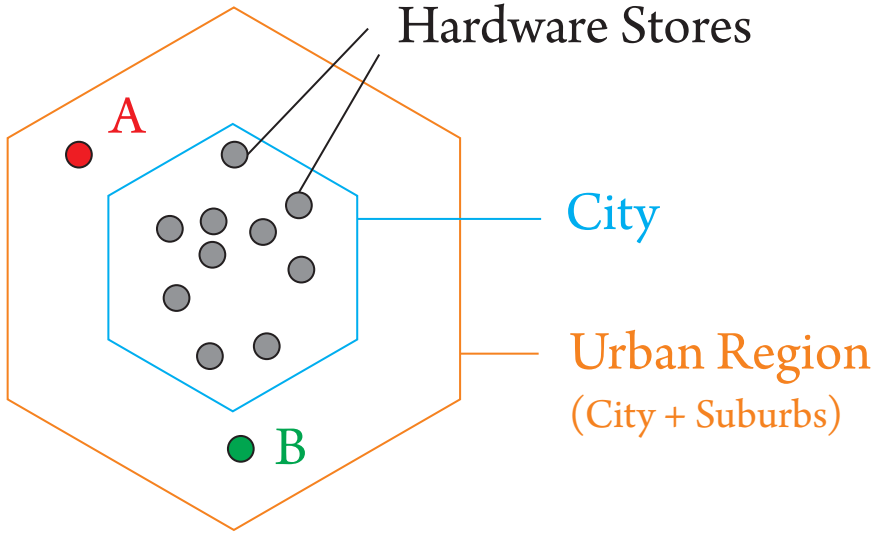


The Inference Problem

Privacy Breach

Store A may provably obtain B's contribution to the sum:

$$B = \sum Region - \sum City - A \tag{1}$$



The Inference Problem

Privacy Breach

Store A may provably obtain B's contribution to the sum:

$$B = \sum Region - \sum City - A \tag{1}$$

The Inference Problem

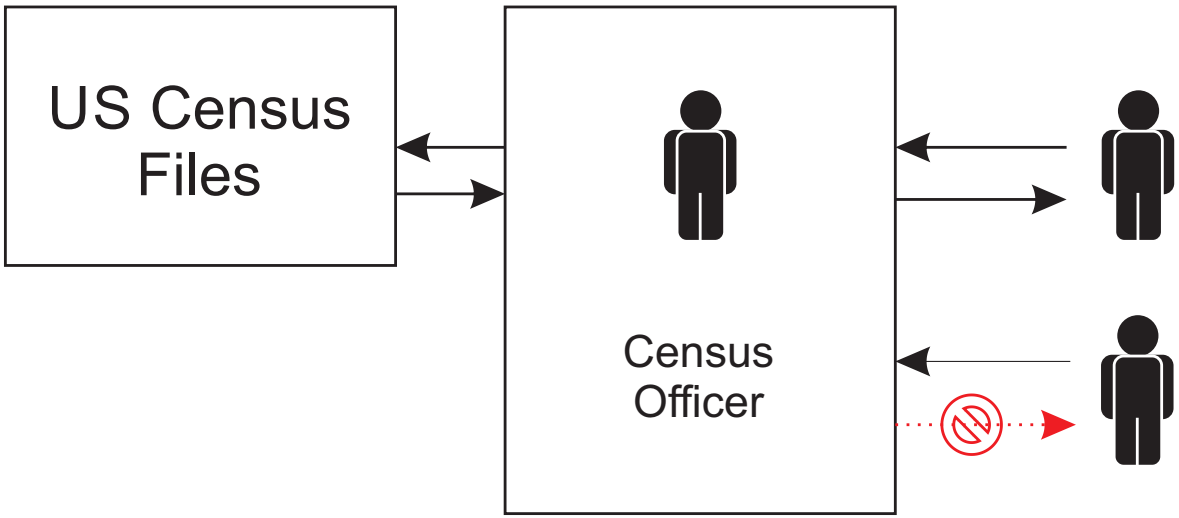
Background knowledge of stakeholders (i.e. Store A) used to **infer** more information than actually published.

Structure

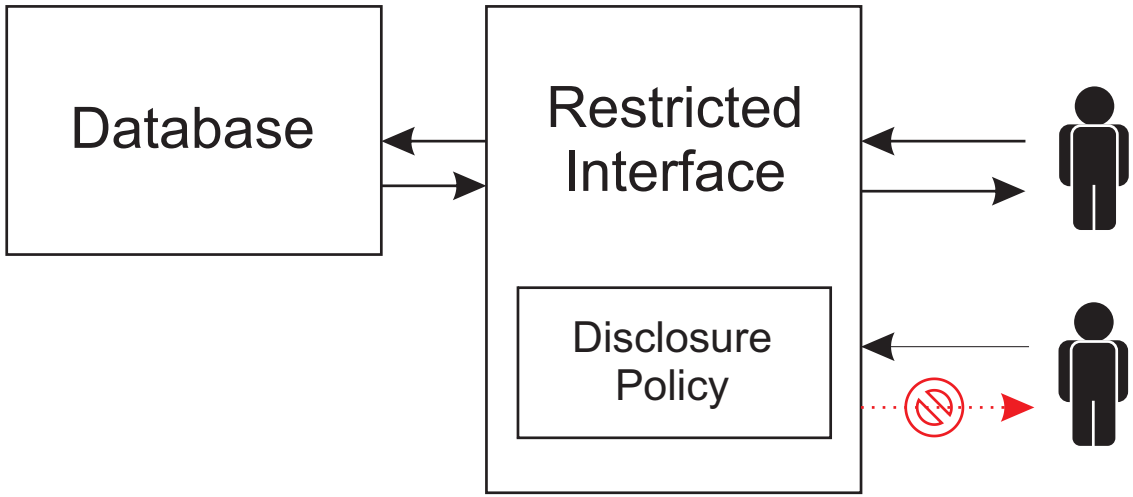
1. Privacy Goals and Examples
2. Early Means of Privacy Control
   - Query Restriction
   - Query Set Approximation
   - Summary
3. Modern Data Anonymization
   - Anonymity Concepts
   - Resampling and Permutation
   - Summary
4. Differential Privacy
   - Epsilon-Differential Privacy
   - Summary
5. Challenges and Current Research Topics

Structure

Query Restriction



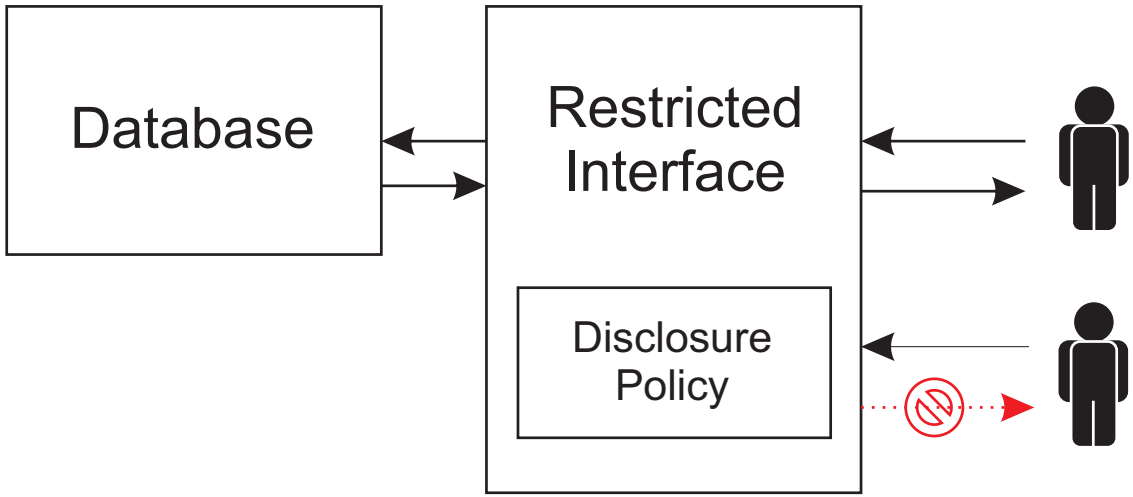
Query Restriction



Disclosure Policies according to [Den80]:

- Reject too small sets (minimum query set control)
- Reject too similar queries (minimum overlap control)
- Deny results which lead to a solvable system of equations (Auditing, only theoretical due to complexity)

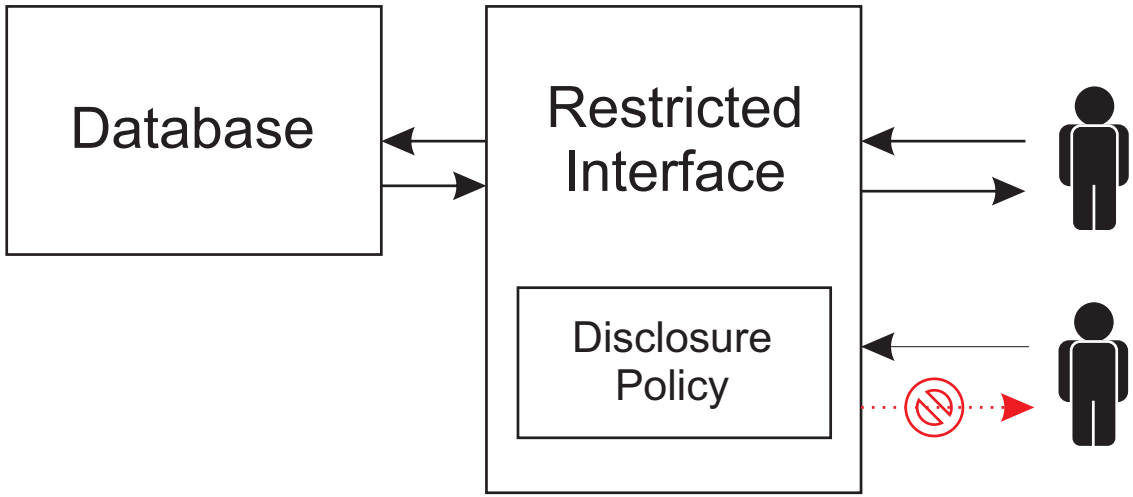
Query Restriction



Disclosure Policies according to [Den80]:

- Reject too small sets (minimum query set control)
- Reject too similar queries (minimum overlap control)
- Deny results which lead to a solvable system of equations (Auditing, only theoretical due to complexity)

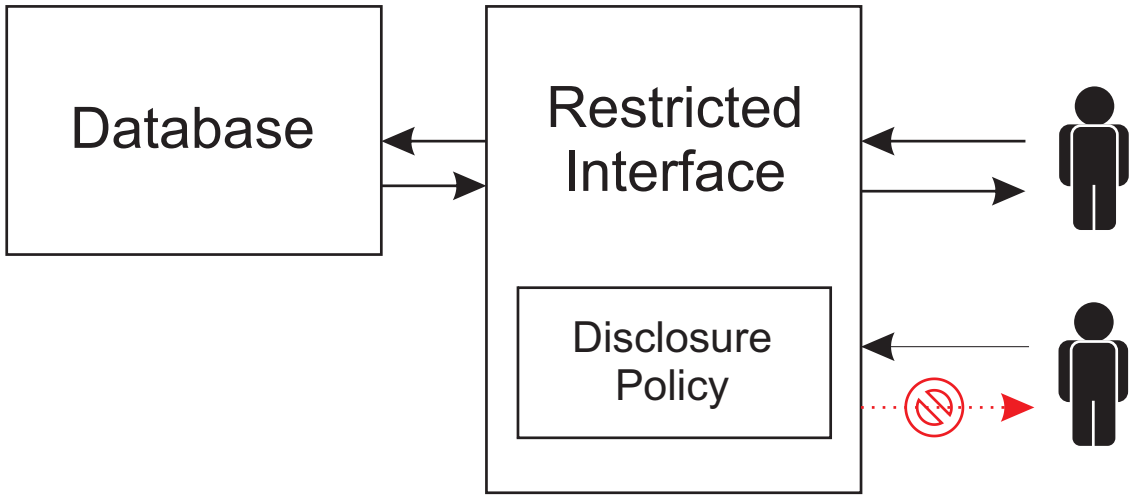
Query Restriction



Disclosure Policies according to [Den80]:

- Reject too small sets (minimum query set control)
- Reject too similar queries (minimum overlap control)
- Deny results which lead to a solvable system of equations (Auditing, only theoretical due to complexity)

Query Restriction



Disclosure Policies according to [Den80]:

- Reject too small sets (minimum query set control)
- Reject too similar queries (minimum overlap control)
- Deny results which lead to a solvable system of equations (Auditing, only theoretical due to complexity)

Query Restriction

Example

Name	Age	Income p.a.
Person A	30	$40.000
Person B	35	$60.000
Person C	40	$30.000
Person D	45	$80.000

Minimum Query Set Control $|Q| \geq 3$

Attack Vector

$$Q_1 = SUM(Income|Age < 42) = \$130.000 \quad (2)$$

$$Q_2 = SUM(Income|Age < 50) = \$210.000 \quad (3)$$

$$Income_D = Q_2 - Q_1 = \$80.000 \quad (4)$$

Query Restriction

Example

Name	Age	Income p.a.
Person A	30	$40.000
Person B	35	$60.000
Person C	40	$30.000
Person D	45	$80.000

Minimum Query Set Control $|Q| \geq 3$

Attack Vector

$$Q_1 = SUM(Income|Age < 42) = \$130.000 \quad (2)$$

$$Q_2 = SUM(Income|Age < 50) = \$210.000 \quad (3)$$

$$Income_D = Q_2 - Q_1 = \$80.000 \quad (4)$$

Query Restriction

Example

Name	Age	Income p.a.
Person A	30	$40.000
Person B	35	$60.000
Person C	40	$30.000
Person D	45	$80.000

Minimum Query Set Control $|Q| \geq 3$

Attack Vector

$$Q_1 = SUM(Income|Age < 42) = \$130.000 \quad (2)$$

$$Q_2 = SUM(Income|Age < 50) = \$210.000 \quad (3)$$

$$Income_D = Q_2 - Q_1 = \$80.000 \quad (4)$$

Query Restriction

Example

Name	Age	Income p.a.
Person A	30	$40.000
Person B	35	$60.000
Person C	40	$30.000
Person D	45	$80.000

Minimum Query Set Control $|Q| \geq 3$

Attack Vector

$$Q_1 = SUM(Income|Age < 42) = \$130.000 \quad (2)$$

$$Q_2 = SUM(Income|Age < 50) = \$210.000 \quad (3)$$

$$Income_D = Q_2 - Q_1 = \$80.000 \quad (4)$$

Example

Name	Age	Income p.a.
Person A	30	$40.000
Person B	35	$60.000
Person C	40	$30.000
Person D	45	$80.000

Minimum Overlap Control

Attack similar to Minimum Query Set Control with more equations to be solved.

Partitioning (a.k.a. Microaggregation)

Raw Data

Name	Age	Income p.a.
Person A	30	$40.000
Person B	35	$60.000
Person C	40	$30.000
Person D	45	$80.000

Partitioned Data

Name	Age	Income p.a.
Person AB	30 - 39	$50.000 ($\pm$20.000)
Person CD	40 - 49	$55.000 ($\pm$25.000)

1960 US Census data available with partitions of size 1000 [Han71]

Partitioning (a.k.a. Microaggregation)

Raw Data

Name	Age	Income p.a.
Person A	30	$40.000
Person B	35	$60.000
Person C	40	$30.000
Person D	45	$80.000

Partitioned Data

Name	Age	Income p.a.
Person AB	30 - 39	$50.000 (±20.000)
Person CD	40 - 49	$55.000 (±25.000)

1960 US Census data available with partitions of size 1000 [Han71]

Partitioning (a.k.a. Microaggregation)

Raw Data

Name	Age	Income p.a.
Person A	30	$40.000
Person B	35	$60.000
Person C	40	$30.000
Person D	45	$80.000

Partitioned Data

Name	Age	Income p.a.
Person AB	30 - 39	$50.000 ($\pm$20.000)
Person CD	40 - 49	$55.000 ($\pm$25.000)

1960 US Census data available with partitions of size 1000 [Han71]

Partitioning (a.k.a. Microaggregation)

Partitioned Data

Name	Age	Income p.a.
Person AB	30 - 39	\$50.000 ($\pm$20.000)
Person CD	40 - 49	\$55.000 ($\pm$25.000)

Consider **adding** Person E (Age 47, \$40.000):

Attack Vector

Given: Knowledge about E's age and the previous database

$$Income_E = NewSize * NewIncome - OldSize * OldIncome \quad (2)$$
$$= 3 * 50.000 - 2 * 55.000 \quad (3)$$
$$= 40.000 \quad (4)$$

[AW89]

Partitioning (a.k.a. Microaggregation)

Partitioned Data

Name	Age	Income p.a.
Person AB	30 - 39	\$50.000 ($\pm$20.000)
~~Person CD~~	~~40 - 49~~	~~\$55.000 ($\pm$25.000)~~
Person CDE	40 - 49	\$50.000 ($\pm$25.000)

Consider **adding** Person E (Age 47, \$40.000):

Attack Vector

Given: Knowledge about E's age and the previous database

$$Income_E = NewSize * NewIncome - OldSize * OldIncome \quad (2)$$
$$= 3 * 50.000 - 2 * 55.000 \quad (3)$$
$$= 40.000 \quad (4)$$

[AW89]

Partitioning (a.k.a. Microaggregation)

Partitioned Data

Name	Age	Income p.a.
Person AB	30 - 39	\$50.000 ($\pm$20.000)
~~Person CD~~	~~40 - 49~~	~~\$55.000 ($\pm$25.000)~~
Person CDE	40 - 49	\$50.000 ($\pm$25.000)

Consider **adding** Person E (Age 47, \$40.000):

Attack Vector

Given: Knowledge about E's age and the previous database

$$Income_E = NewSize * NewIncome - OldSize * OldIncome \quad (2)$$
$$= 3 * 50.000 - 2 * 55.000 \quad (3)$$
$$= 40.000 \quad (4)$$

[AW89]

Random Sampling

- **Idea** Pseudo-randomly select candidates for a given query
- same queries must operate on the same subset

Simplified Algorithm[Den80]

1. Define elimination probability $P_e = 2^{-k}$.
2. Preprocessing: For each record r_i compute a hash $h_i \in \{0,1\}^k$.
3. Query processing: Compute hash $H_q \in \{0,1\}^k$ from the query.
4. Eliminate all r_i where $h_i = H_q$ from query set.

Announce k, but keep hash function secret.

Random Sampling

- **Idea** Pseudo-randomly select candidates for a given query
- same queries must operate on the same subset

Simplified Algorithm[Den80]

1. Define elimination probability $P_e = 2^{-k}$.
2. Preprocessing: For each record r_i compute a hash $h_i \in \{0,1\}^k$.
3. Query processing: Compute hash $H_q \in \{0,1\}^k$ from the query.
4. Eliminate all r_i where $h_i = H_q$ from query set.

Announce k, but keep hash function secret.

Random Sampling

- **Idea** Pseudo-randomly select candidates for a given query
- same queries must operate on the same subset

Simplified Algorithm[Den80]

1. Define elimination probability $P_e = 2^{-k}$.
2. Preprocessing: For each record r_i compute a hash $h_i \in \{0,1\}^k$.
3. Query processing: Compute hash $H_q \in \{0,1\}^k$ from the query.
4. Eliminate all r_i where $h_i = H_q$ from query set.

Announce k, but keep hash function secret.

Random Sampling

- **Idea** Pseudo-randomly select candidates for a given query
- same queries must operate on the same subset

Simplified Algorithm[Den80]

1. Define elimination probability $P_e = 2^{-k}$.
2. Preprocessing: For each record r_i compute a hash $h_i \in \{0,1\}^k$.
3. Query processing: Compute hash $H_q \in \{0,1\}^k$ from the query.
4. Eliminate all r_i where $h_i = H_q$ from query set.

Announce k, but keep hash function secret.

Random Sampling

- **Idea** Pseudo-randomly select candidates for a given query
- same queries must operate on the same subset

Simplified Algorithm[Den80]

1. Define elimination probability $P_e = 2^{-k}$.
2. Preprocessing: For each record r_i compute a hash $h_i \in \{0,1\}^k$.
3. Query processing: Compute hash $H_q \in \{0,1\}^k$ from the query.
4. Eliminate all r_i where $h_i = H_q$ from query set.

Announce k, but keep hash function secret.

Random Sampling

- **Idea** Pseudo-randomly select candidates for a given query
- same queries must operate on the same subset

Simplified Algorithm[Den80]

1. Define elimination probability $P_e = 2^{-k}$.
2. Preprocessing: For each record r_i compute a hash $h_i \in \{0,1\}^k$.
3. Query processing: Compute hash $H_q \in \{0,1\}^k$ from the query.
4. Eliminate all r_i where $h_i = H_q$ from query set.

Announce k, but keep hash function secret.

Random Sampling

- **Idea** Pseudo-randomly select candidates for a given query
- same queries must operate on the same subset

Simplified Algorithm[Den80]

1. Define elimination probability $P_e = 2^{-k}$.
2. Preprocessing: For each record r_i compute a hash $h_i \in \{0,1\}^k$.
3. Query processing: Compute hash $H_q \in \{0,1\}^k$ from the query.
4. Eliminate all r_i where $h_i = H_q$ from query set.

Announce k, but keep hash function secret.

Random Sampling

- **Idea** Pseudo-randomly select candidates for a given query
- same queries must operate on the same subset

Simplified Algorithm[Den80]

1. Define elimination probability $P_e = 2^{-k}$.
2. Preprocessing: For each record r_i compute a hash $h_i \in \{0,1\}^k$.
3. Query processing: Compute hash $H_q \in \{0,1\}^k$ from the query.
4. Eliminate all r_i where $h_i = H_q$ from query set.

Announce k, but keep hash function secret.

Summary 1960 - 1990

Query Restriction Approach: (Inspired by manual work performed by Census Officers in pre-DBMS age)

- Minimum Query Set Control
- Minimum Overlap Control
- Auditing (for small databases)

Query Set Approximation Approach:

- Partitioning/Microaggregation (for static Databases)
- Random Sampling

Summary 1960 - 1990

Query Restriction Approach: (Inspired by manual work performed by Census Officers in pre-DBMS age)

- Minimum Query Set Control
- Minimum Overlap Control
- Auditing (for small databases)

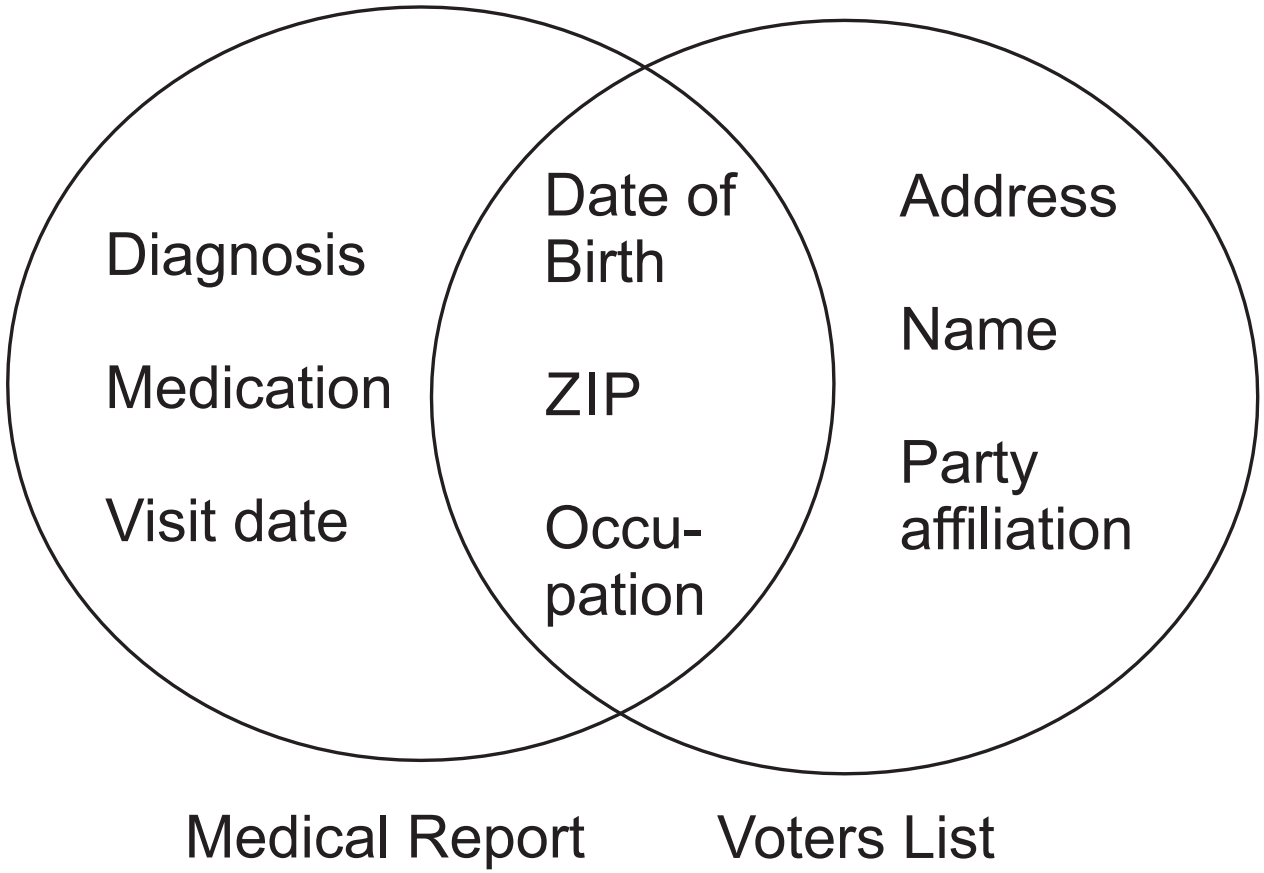
Query Set Approximation Approach:

- Partitioning/Microaggregation (for static Databases)
- Random Sampling

Structure

Anonymizing Data

Removing names and adresses in a clinical report may not be sufficient...



Anonymizing Data

Medical databases:
Publish data for researchers.

Example Data:

Job	Gender	Age	Diagnosis
Engineer	Male	35	Hepatitis
Engineer	Male	38	Hepatitis
Lawyer	Male	38	HIV
Writer	Female	30	Flu
Writer	Female	30	HIV
Dancer	Female	30	HIV
Dancer	Female	30	HIV

Quasi-Identifiers

(Job, Gender, Age) can identify the record owner given the background knowledge. It's a **Quasi Identifier (QID).**

k-Anonymity [Agg05]: Each QID is associated with at least k records.

Example Data: k = 3

Job	Gender	Age	Diagnosis
Professional	Male	35 - 40	Hepatitis
Professional	Male	35 - 40	Hepatitis
Professional	Male	35 - 40	HIV
Artist	Female	30 - 34	Flu
Artist	Female	30 - 34	HIV
Artist	Female	30 - 34	HIV
Artist	Female	30 - 34	HIV

k-Anonymity

k-Anonymity [Agg05]: Each QID is associated with at least k records.

Example Data: k = 3

Job	Gender	Age	Diagnosis
Artist	Female	30 - 34	Flu
Artist	Female	30 - 34	HIV
Artist	Female	30 - 34	HIV
Artist	Female	30 - 34	HIV

Partial Disclosure

Background knowledge about female writer aged 30 yields diagnosis HIV with 75% confidence.

Anonymization Algorithms

Input: Parameter k and QID

Choose Generalizations with **least Information Loss:**

- Numbers to **Ranges**
  - Age 34: [30 – 34] preferred to [30 – 39]
- Categories to **Hypernyms**
  - *Dancer*: *Performing Artist* better than *Artist*

Implementations: (Optimal Solution is *NP-Hard*!)

- Start with original Dataset, iteratively generalize an attribute by a small amount until k-Anonymity reached. (Or start with most general and specialize)[SS98]
- Apply Genetic Algorithms (Start with random generalizations, iteratively combine best-performing configurations adding some mutations, continue until convergence)[Iye02]

Anonymization Algorithms

Input: Parameter k and QID

Choose Generalizations with **least Information Loss:**

- Numbers to **Ranges**
  - Age 34: [30 – 34] preferred to [30 – 39]
- Categories to **Hypernyms**
  - *Dancer*: *Performing Artist* better than *Artist*

Implementations: (Optimal Solution is *NP-Hard*!)

- Start with original Dataset, iteratively generalize an attribute by a small amount until k-Anonymity reached. (Or start with most general and specialize)[SS98]
- Apply Genetic Algorithms (Start with random generalizations, iteratively combine best-performing configurations adding some mutations, continue until convergence)[Iye02]

Anonymization Algorithms

Input: Parameter k and QID

Choose Generalizations with **least Information Loss:**

- Numbers to **Ranges**
  - Age 34: $[30-34]$ preferred to $[30-39]$
- Categories to **Hypernyms**
  - *Dancer*: *Performing Artist* better than *Artist*

Implementations: (Optimal Solution is *NP-Hard*!)

- Start with original Dataset, iteratively generalize an attribute by a small amount until k-Anonymity reached. (Or start with most general and specialize)[SS98]
- Apply Genetic Algorithms (Start with random generalizations, iteratively combine best-performing configurations adding some mutations, continue until convergence)[Iye02]

Anonymization Algorithms

Input: Parameter k and QID

Choose Generalizations with **least Information Loss:**

- Numbers to **Ranges**
  - Age 34: $[30-34]$ preferred to $[30-39]$
- Categories to **Hypernyms**
  - *Dancer*: *Performing Artist* better than *Artist*

Implementations: (Optimal Solution is *NP-Hard*!)

- Start with original Dataset, iteratively generalize an attribute by a small amount until k-Anonymity reached. (Or start with most general and specialize)[SS98]
- Apply Genetic Algorithms (Start with random generalizations, iteratively combine best-performing configurations adding some mutations, continue until convergence)[Iye02]

Anonymization Algorithms

Input: Parameter k and QID

Choose Generalizations with **least Information Loss:**

- Numbers to **Ranges**
  - Age 34: $[30-34]$ preferred to $[30-39]$
- Categories to **Hypernyms**
  - *Dancer*: *Performing Artist* better than *Artist*

Implementations: (Optimal Solution is *NP-Hard*!)

- Start with original Dataset, iteratively generalize an attribute by a small amount until k-Anonymity reached. (Or start with most general and specialize)[SS98]
- Apply Genetic Algorithms (Start with random generalizations, iteratively combine best-performing configurations adding some mutations, continue until convergence)[Iye02]

Anonymization Algorithms

Input: Parameter k and QID

Choose Generalizations with **least Information Loss:**

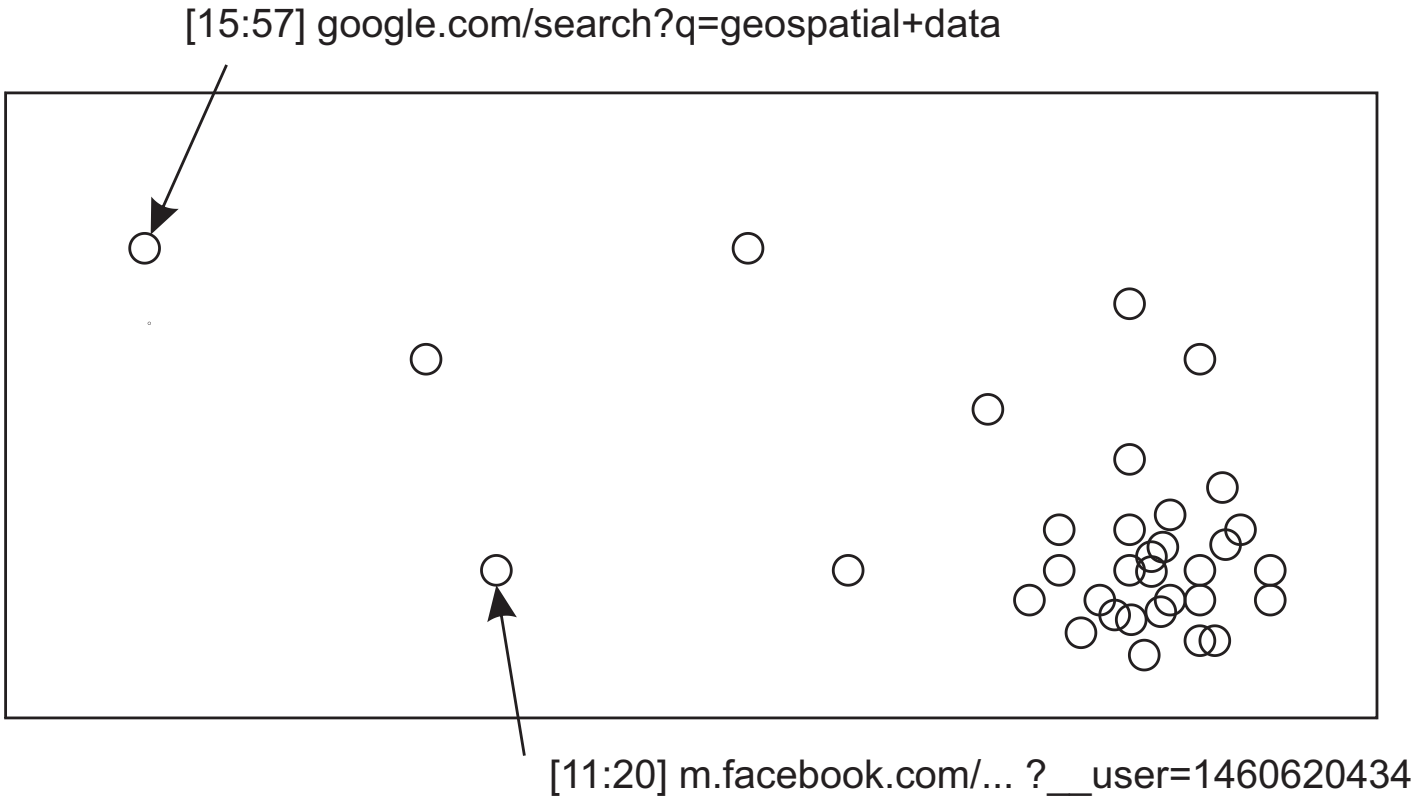
- Numbers to **Ranges**
  - Age 34: $[30-34]$ preferred to $[30-39]$
- Categories to **Hypernyms**
  - *Dancer*: *Performing Artist* better than *Artist*

Implementations: (Optimal Solution is *NP-Hard*!)

- Start with original Dataset, iteratively generalize an attribute by a small amount until k-Anonymity reached. (Or start with most general and specialize)[SS98]
- Apply Genetic Algorithms (Start with random generalizations, iteratively combine best-performing configurations adding some mutations, continue until convergence)[Iye02]

k-Anonymity on Geospatial Data

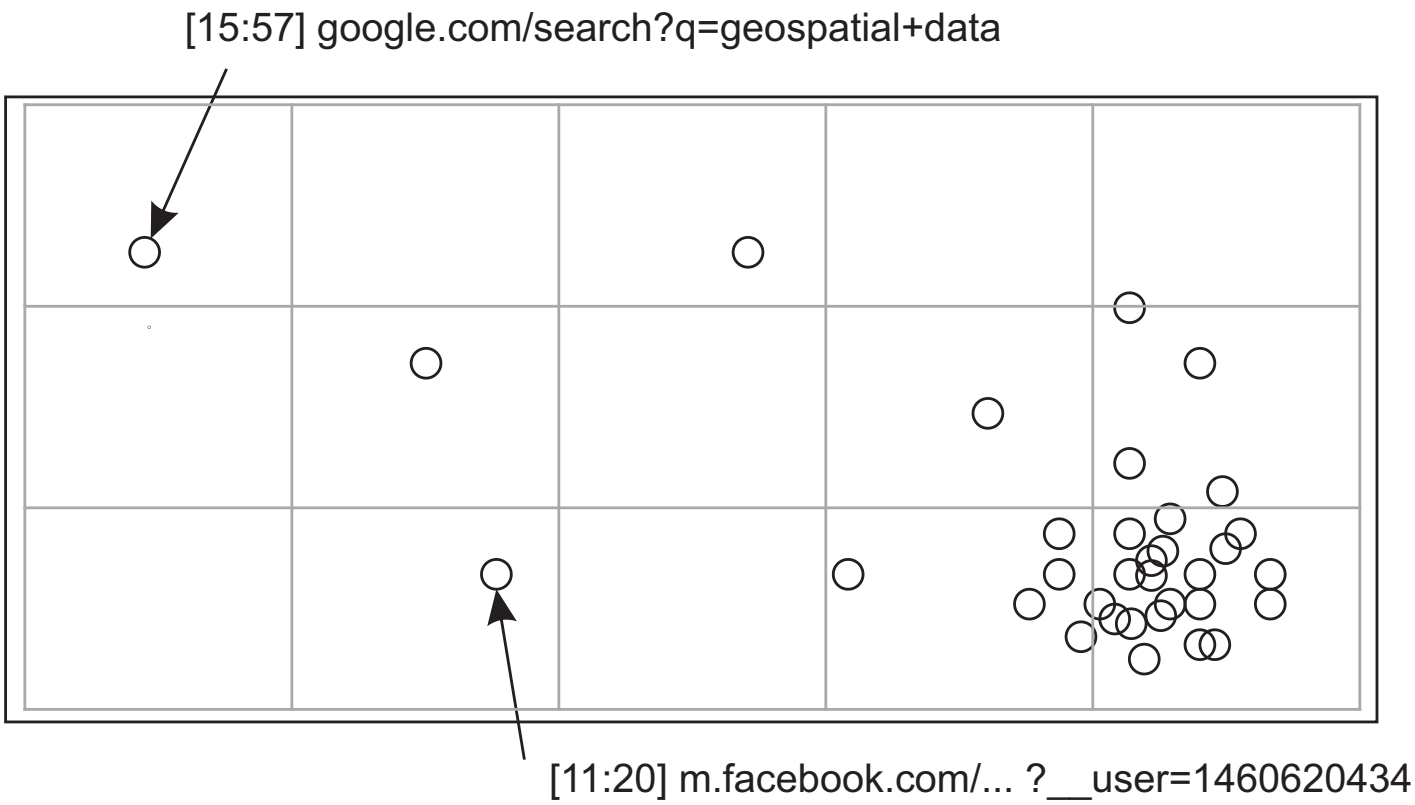
[Qar13]



The Data sold by EE last week contained cells of **at least 50 individuals [Kob13]**

k-Anonymity on Geospatial Data

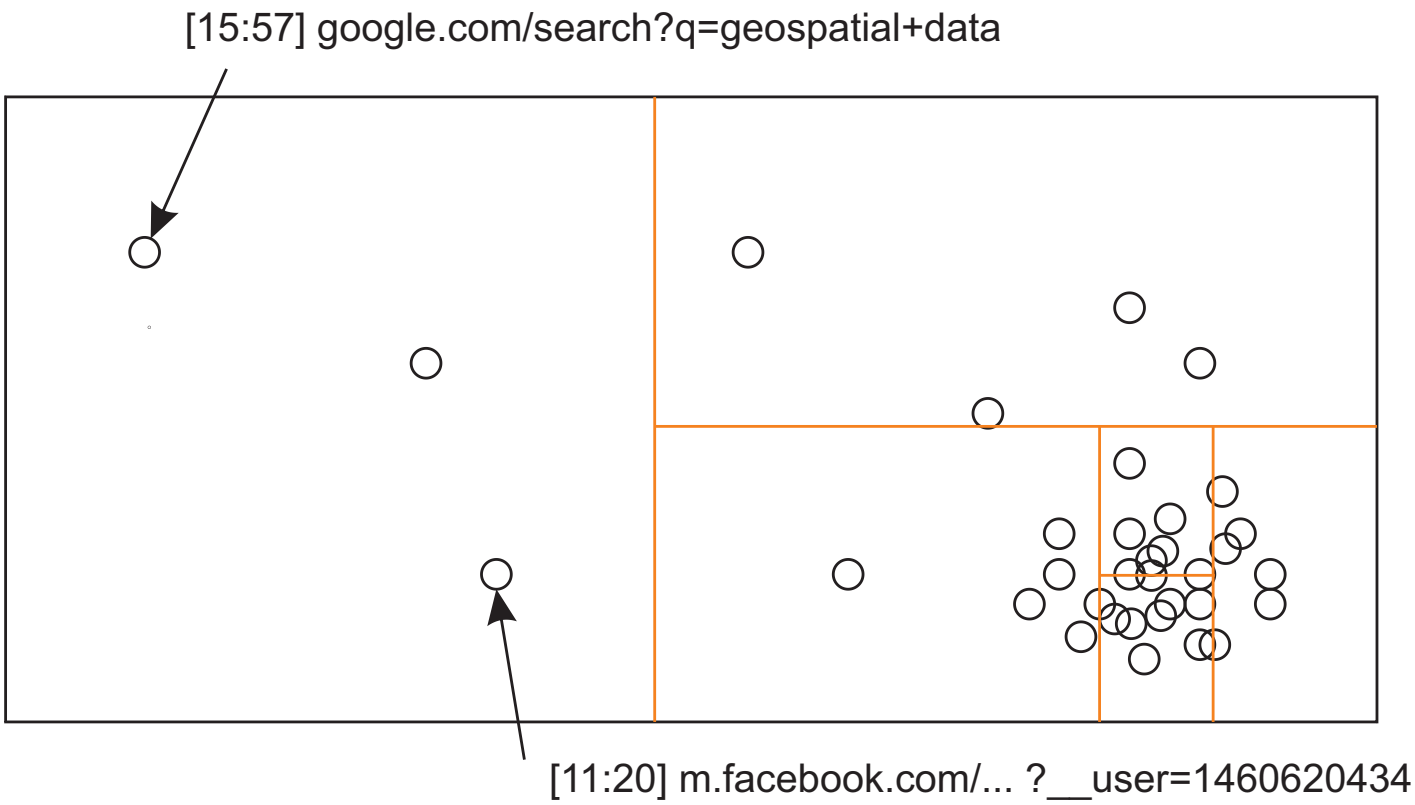
[Qar13]



The Data sold by EE last week contained cells of **at least 50 individuals [Kob13]**

k-Anonymity on Geospatial Data

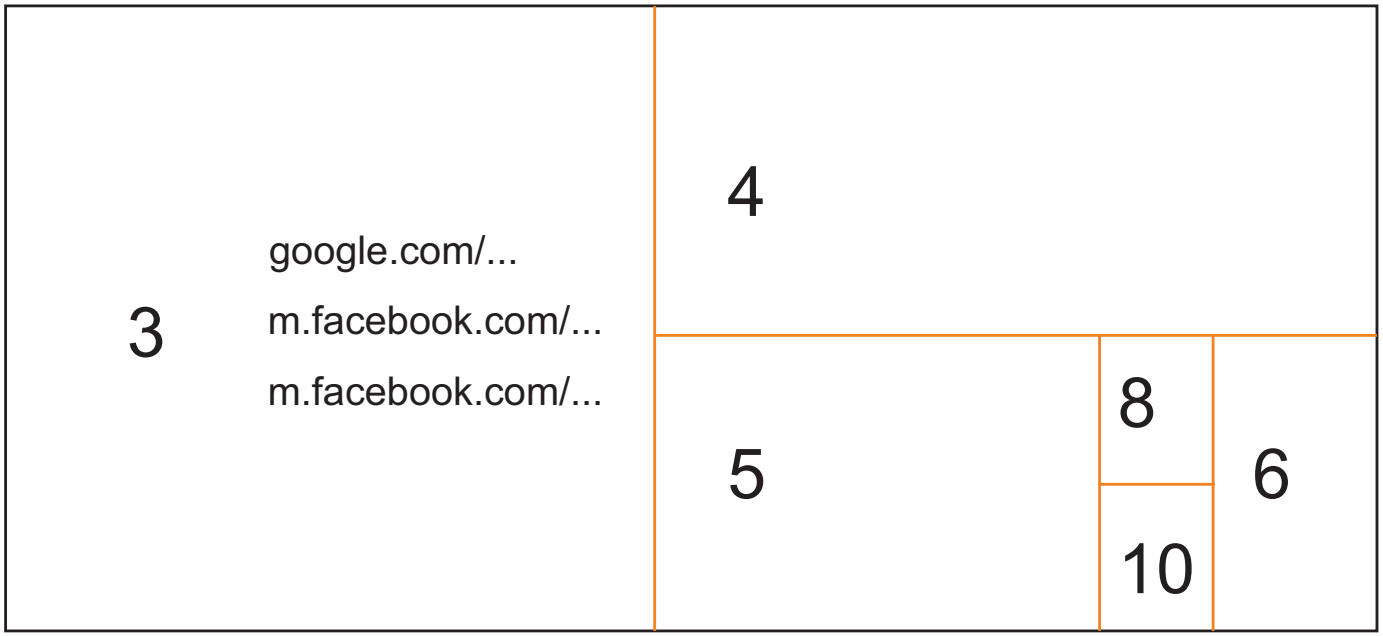
[Qar13]



The Data sold by EE last week contained cells of **at least 50 individuals [Kob13]**

k-Anonymity on Geospatial Data

[Qar13]



The Data sold by EE last week contained cells of **at least 50 individuals [Kob13]**

l-Diversity

l-Diversity [2006]: Each QID is associated with at least l different values for sensitive attributes.

Our **Example Data** is only **2-Diverse**!
Increase group size achieving **3-Diversity:**

Job	Gender	Age	Diagnosis
Professional	Male	35 - 40	Hepatitis
Professional	Male	35 - 40	Hepatitis
Professional	Male	35 - 40	HIV
Artist	Female	30 - 34	Flu
Artist	Female	30 - 34	HIV
Artist	Female	30 - 34	HIV
Artist	Female	30 - 34	HIV

l-Diversity

l-Diversity [2006]: Each QID is associated with at least l different values for sensitive attributes.

Our **Example Data** is only **2-Diverse**!
Increase group size achieving **3-Diversity:**

Job	Gender	Age	Diagnosis
Professional	Male	35 - 40	Hepatitis
Professional	Male	35 - 40	Hepatitis
Professional	Male	35 - 40	HIV
Professional	Male	35 - 40	**Flu**
Artist	Female	30 - 34	**Hepatitis**
Artist	Female	30 - 34	Flu
Artist	Female	30 - 34	HIV
Artist	Female	30 - 34	HIV
Artist	Female	30 - 34	HIV

t-Closeness

t-Closeness [2007]: Distribution of sensitive values inside a group closely resembles the overall Distribution.

t: Upper bound on *Distance* between distributions.

Example for categorical values:

Element	Overall	10%-Closeness	5%-Closeness
Flu	90%	80 - 100%	85 - 95%
Hepatitis	5%	0 - 15%	0 - 10%
HIV	5%	0 - 15%	0 - 10%

(Numeric values use *Earth Mover's Distance*)

t-Closeness

t-Closeness [2007]: Distribution of sensitive values inside a group closely resembles the overall Distribution.

t: Upper bound on *Distance* between distributions.

Example for categorical values:

Element	Overall	10%-Closeness	5%-Closeness
Flu	90%	80 - 100%	85 - 95%
Hepatitis	5%	0 - 15%	0 - 10%
HIV	5%	0 - 15%	0 - 10%

(Numeric values use *Earth Mover's Distance*)

t-Closeness [2007]: Distribution of sensitive values inside a group closely resembles the overall Distribution.

t: Upper bound on *Distance* between distributions.

Example for categorical values:

Element	Overall	10%-Closeness	5%-Closeness
Flu	90%	80 - 100%	85 - 95%
Hepatitis	5%	0 - 15%	0 - 10%
HIV	5%	0 - 15%	0 - 10%

(Numeric values use *Earth Mover's Distance*)

Other Approaches [FWCY10]

Resampling:

- Construct distribution of original data.
- Replace $n\%$ values by random values drawn from this distribution.

Permutation:

- Select groups of some size.
- Shuffle sensitive values inside group to **de-associate QID and value.**

Preserve Mean, Variance and Distribution of isolated Attributes but **heavily impact** Correlation and Covariance between Attributes.

Other Approaches [FWCY10]

Resampling:

- Construct distribution of original data.
- Replace $n\%$ values by random values drawn from this distribution.

Permutation:

- Select groups of some size.
- Shuffle sensitive values inside group to **de-associate QID and value.**

Preserve Mean, Variance and Distribution of isolated Attributes but **heavily impact** Correlation and Covariance between Attributes.

Other Approaches [FWCY10]

Resampling:

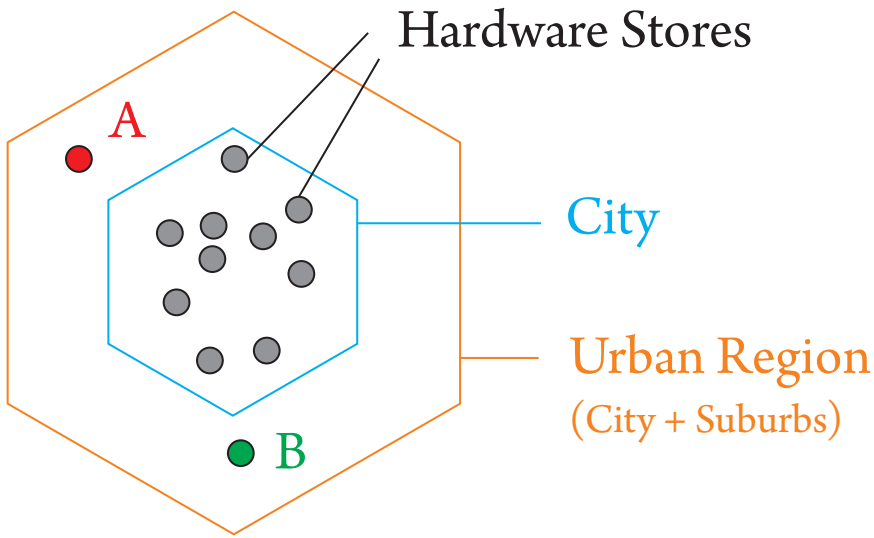
- Construct distribution of original data.
- Replace $n\%$ values by random values drawn from this distribution.

Permutation:

- Select groups of some size.
- Shuffle sensitive values inside group to **de-associate QID and value.**

Preserve Mean, Variance and Distribution of isolated Attributes but **heavily impact** Correlation and Covariance between Attributes.

Knowing the previous techniques: What about the Census Example from the beginning?



Attack Vector

$$B = \sum Region - \sum City - A \tag{5}$$

Attack Vector?

$$B = \sum Region - \sum City - A \tag{6}$$

Rendering the sums **ineffective** for precise disclosure:

- Query Set Overlap Control: reject query
- Random Sampling: additional missing stores
- Microaggregations with size ≥ 3 / 3-Anonymity: aggregates differ in 0 or 3 stores
- Resampling: A computes either the real or a random number.
- Permutation: A computes a random competitor's revenue.

Attack Vector?

$$B = \sum Region - \sum City - A \tag{6}$$

Rendering the sums **ineffective** for precise disclosure:

- Query Set Overlap Control: reject query
- Random Sampling: additional missing stores
- Microaggregations with size ≥ 3 / 3-Anonymity: aggregates differ in 0 or 3 stores
- Resampling: A computes either the real or a random number.
- Permutation: A computes a random competitor's revenue.

Attack Vector?

$$B = \sum Region - \sum City - A \tag{6}$$

Rendering the sums **ineffective** for precise disclosure:

- Query Set Overlap Control: reject query
- Random Sampling: additional missing stores
- Microaggregations with size ≥ 3 / 3-Anonymity: aggregates differ in 0 or 3 stores
- Resampling: A computes either the real or a random number.
- Permutation: A computes a random competitor's revenue.

Attack Vector?

$$B = \sum Region - \sum City - A \tag{6}$$

Rendering the sums **ineffective** for precise disclosure:

- Query Set Overlap Control: reject query
- Random Sampling: additional missing stores
- Microaggregations with size ≥ 3 / 3-Anonymity: aggregates differ in 0 or 3 stores
- Resampling: A computes either the real or a random number.
- Permutation: A computes a random competitor's revenue.

Attack Vector?

$$B = \sum Region - \sum City - A \tag{6}$$

Rendering the sums **ineffective** for precise disclosure:

- Query Set Overlap Control: reject query
- Random Sampling: additional missing stores
- Microaggregations with size ≥ 3 / 3-Anonymity: aggregates differ in 0 or 3 stores
- Resampling: A computes either the real or a random number.
- Permutation: A computes a random competitor's revenue.

Attack Vector?

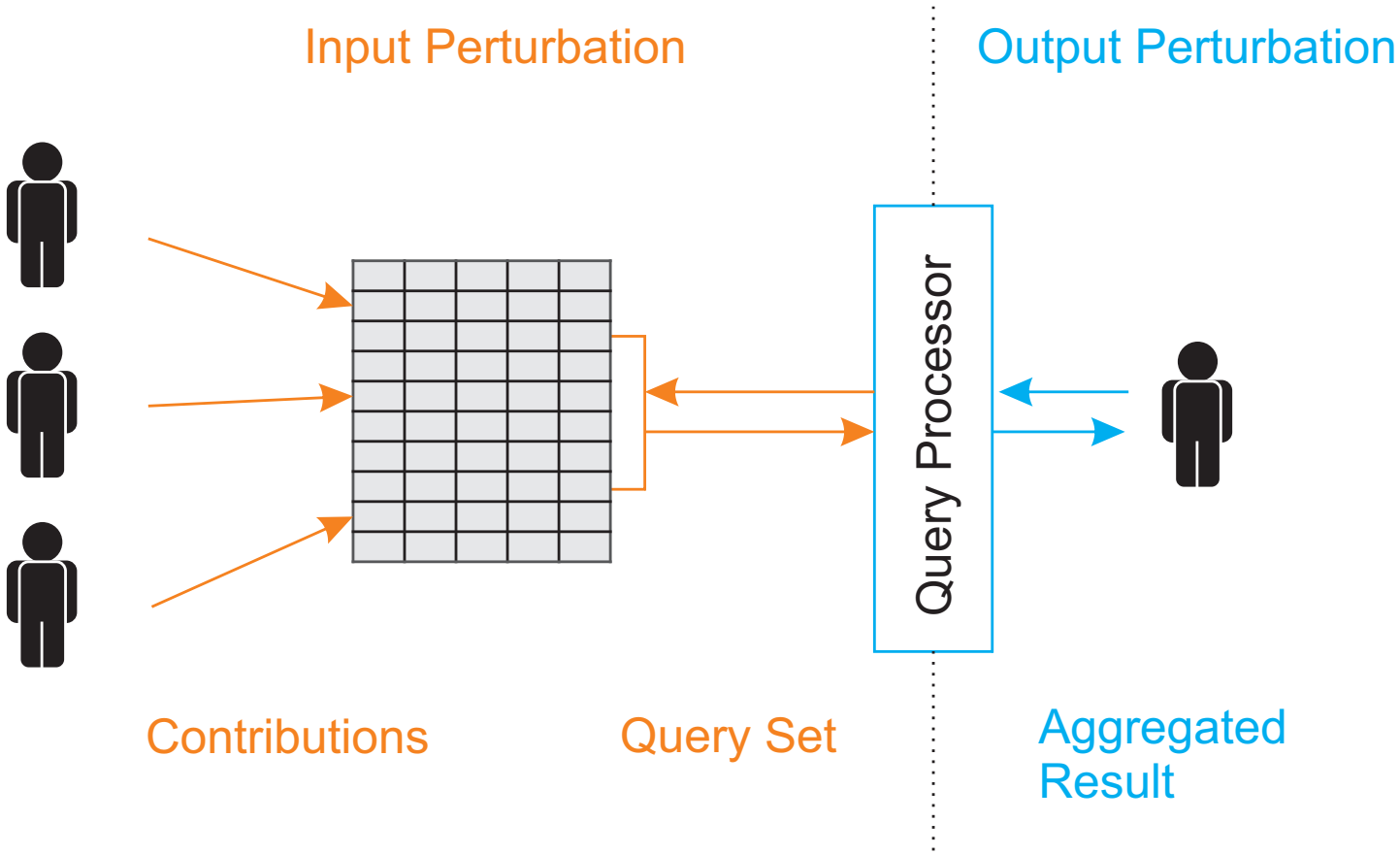
$$B = \sum Region - \sum City - A \tag{6}$$

Rendering the sums **ineffective** for precise disclosure:

- Query Set Overlap Control: reject query
- Random Sampling: additional missing stores
- Microaggregations with size ≥ 3 / 3-Anonymity: aggregates differ in 0 or 3 stores
- Resampling: A computes either the real or a random number.
- Permutation: A computes a random competitor's revenue.

Structure

Modern Applications



ϵ-Differential Privacy

Person	Attribute X	Attribute Y
Participant A	No	No
Participant B	Yes	Yes
Participant C	Yes	Yes
Participant D	Yes	Yes

Attack Vector

Query $COUNT(X = Y) = COUNT(ALL) = 4$ and you can infer $X \Rightarrow Y$

Solution: Modify COUNT

Randomly **add or subtract 1**. Each participant can now plausibly claim he had no influence on the result, because **the answer can also be generated by a Database not containing his contribution**.

ϵ-Differential Privacy

Person	Attribute X	Attribute Y
Participant A	No	No
Participant B	Yes	Yes
Participant C	Yes	Yes
Participant D	Yes	Yes

Attack Vector

Query $COUNT(X = Y) = COUNT(ALL) = 4$ and you can infer $X \Rightarrow Y$

Solution: Modify COUNT

Randomly **add or subtract 1**. Each participant can now plausibly claim he had no influence on the result, because **the answer can also be generated by a Database not containing his contribution**.

ϵ-Differential Privacy

Person	Attribute X	Attribute Y
Participant A	No	No
Participant B	Yes	Yes
Participant C	Yes	Yes
Participant D	Yes	Yes

Attack Vector

Query $COUNT(X = Y) = COUNT(ALL) = 4$ and you can infer $X \Rightarrow Y$

Solution: Modify COUNT

Randomly **add or subtract 1**. Each participant can now plausibly claim he had no influence on the result, because **the answer can also be generated by a Database not containing his contribution**.

ϵ-Differential Privacy

Definition

An aggregated result $y = f(D)$ over Database D is **differentially private** if each Database D_Δ differing in a **single element** from D can plausibly generate the same result y.

Can we measure **plausibility**?

Plausibility

A result $y = f(a)$ can be plausibly generated by a different value b if the outcomes are equally probable:
$Pr(y = f(a)) \approx Pr(y = f(b))$

ϵ-Differential Privacy

$$Pr(y = f(a)) \approx Pr(y = f(b)) \tag{7}$$

(8)

$$Pr(y = f(a)) \approx Pr(y = f(b)) \tag{7}$$
$$\Leftrightarrow \quad Pr(y = f(D)) \approx Pr(y = f(D_\Delta)) \tag{8}$$
$$\tag{9}$$

ϵ-Differential Privacy

$$Pr(y = f(a)) \approx Pr(y = f(b)) \tag{7}$$

$$\Leftrightarrow \quad Pr(y = f(D)) \approx Pr(y = f(D_\Delta)) \tag{8}$$

$$\Leftrightarrow \quad \frac{Pr(y = f(D))}{Pr(y = f(D_\Delta))} \leq e^\epsilon \tag{9}$$

ϵ-Differential Privacy

Definition [DMNS06]

The result y of an aggregating function f satisfies ϵ-**Indistinguishability** if for each two Databases D and D_Δ differing in a single Element:

$$\frac{Pr(y = f(D))}{Pr(y = f(D_\Delta))} \leq e^\epsilon \tag{10}$$

Example:
COUNT answers...

- the truth with P = 0.5
- one less with P = 0.25
- one more with P = 0.25

Given a Database D with $COUNT(D) = 4$ and a Database D_Δ with $COUNT(D_\Delta) = 3$, making them differ in 1 element.

$$\frac{Pr(COUNT(D) = 4))}{Pr(COUNT(D') = 4))} \tag{11}$$

$$= \frac{0.5}{0.25} = 2 = e^{0.69} \tag{12}$$

ϵ-Differential Privacy

Example:
COUNT answers...

- the truth with P = 0.5
- one less with P = 0.25
- one more with P = 0.25

Given a Database D with $COUNT(D) = 4$ and a Database D_Δ with $COUNT(D_\Delta) = 3$, making them differ in 1 element.

$$\frac{Pr(COUNT(D) = 4))}{Pr(COUNT(D') = 4))} \tag{11}$$

$$= \frac{0.5}{0.25} = 2 = e^{0.69} \tag{12}$$

Example:
COUNT answers...

- the truth with P = 0.5
- one less with P = 0.25
- one more with P = 0.25

Given a Database D with $COUNT(D) = 4$ and a Database D_Δ with $COUNT(D_\Delta) = 3$, making them differ in 1 element.

$$\frac{Pr(COUNT(D) = 4))}{Pr(COUNT(D') = 4))} \tag{11}$$

$$= \frac{0.5}{0.25} = 2 = e^{0.69} \tag{12}$$

Example:
COUNT answers...

- the truth with P = 0.5
- one less with P = 0.25
- one more with P = 0.25

Given a Database D with $COUNT(D) = 4$ and a Database D_Δ with $COUNT(D_\Delta) = 3$, making them differ in 1 element.

$$\frac{Pr(COUNT(D) = 4))}{Pr(COUNT(D') = 4))} \tag{11}$$

$$= \frac{0.5}{0.25} = 2 = e^{0.69} \tag{12}$$

What about *SUM* and other queries?

Add **noise** proportional to the **influence of a single individual:**

- COUNT: 1 (+1 or -1 is fine)
- SUM: $range(D) = max(d) - min(d)$
- MEAN: $range(D)/count(D)$

Sensitivity

The Sensitivity $S(f)$ of a function f is defined as the maximum change a single contribution in a Database D can cause to $f(D)$

What about *SUM* and other queries?

Add **noise** proportional to the **influence of a single individual:**

- COUNT: 1 (+1 or -1 is fine)
- SUM: $range(D) = max(d) - min(d)$
- MEAN: $range(D)/count(D)$

Sensitivity

The Sensitivity $S(f)$ of a function f is defined as the maximum change a single contribution in a Database D can cause to $f(D)$

What about *SUM* and other queries?

Add **noise** proportional to the **influence of a single individual:**

- COUNT: 1 (+1 or -1 is fine)
- SUM: $range(D) = max(d) - min(d)$
- MEAN: $range(D)/count(D)$

Sensitivity

The Sensitivity $S(f)$ of a function f is defined as the maximum change a single contribution in a Database D can cause to $f(D)$

What about *SUM* and other queries?

Add **noise** proportional to the **influence of a single individual:**

- COUNT: 1 (+1 or -1 is fine)
- SUM: $range(D) = max(d) - min(d)$
- MEAN: $range(D)/count(D)$

Sensitivity

The Sensitivity $S(f)$ of a function f is defined as the maximum change a single contribution in a Database D can cause to $f(D)$

What about *SUM* and other queries?

Add **noise** proportional to the **influence of a single individual:**

- COUNT: 1 (+1 or -1 is fine)
- SUM: $range(D) = max(d) - min(d)$
- MEAN: $range(D)/count(D)$

Sensitivity

The Sensitivity $S(f)$ of a function f is defined as the maximum change a single contribution in a Database D can cause to $f(D)$

What about *SUM* and other queries?
Add **noise** proportional to the **influence of a single individual:**

- COUNT: 1 (+1 or -1 is fine)
- SUM: $range(D) = max(d) - min(d)$
- MEAN: $range(D)/count(D)$

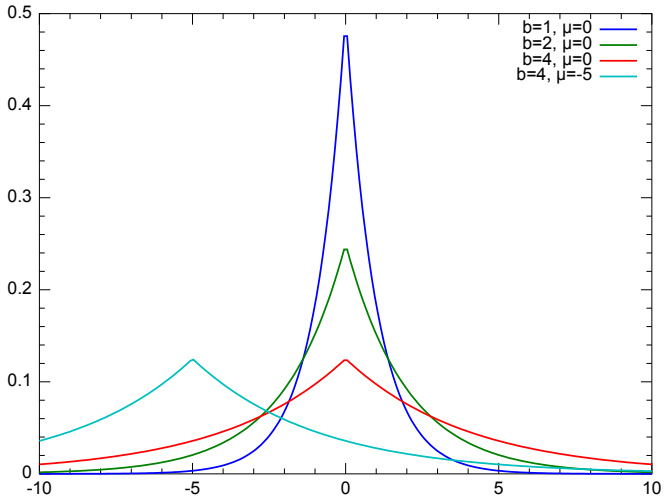
Sensitivity

The Sensitivity $S(f)$ of a function f is defined as the maximum change a single contribution in a Database D can cause to $f(D)$

Laplace Distribution

$$Lap(\lambda) : Pr[X = x] \propto e^{\frac{-|x|}{\lambda}} \tag{13}$$

λ: standard deviation



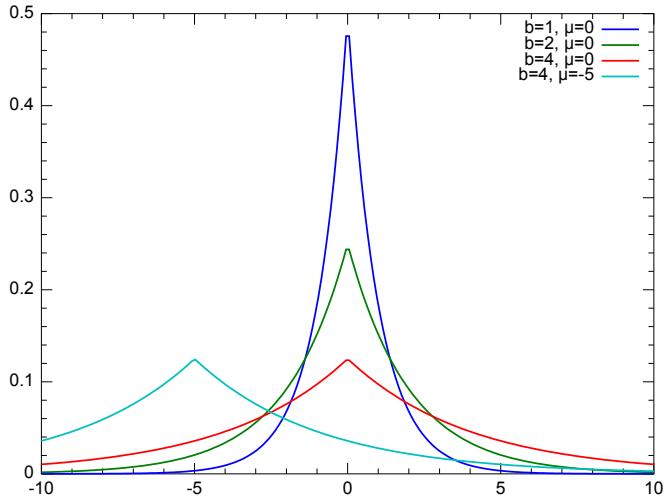
Theorem [DMNS06]

Answering Query $f(D)$ with $f(D) + x$ where $x \sim Lap(S(f)/\epsilon)$ always satisfies ϵ-Indistinguishability.

Laplace Distribution

$$Lap(\lambda) : Pr[X = x] \propto e^{\frac{-|x|}{\lambda}} \tag{13}$$

λ: standard deviation



Theorem [DMNS06]

Answering Query $f(D)$ with $f(D) + x$ where $x \sim Lap(S(f)/\epsilon)$ always satisfies ϵ-Indistinguishability.

Summary of Modern Developments

Data Anonymization Approach:

- k-Anonymity (Groups of k indistinguishable individuals)
- l-Diversity (Groups of l different sensitive values)
- t-Closeness (Groups reflecting overall distribution)
- Resampling / Compression (Random data reflecting the real-world distribution)
- Permutation (deassociating individual and sensitive data)

Output Perturbation Approach:

- ϵ-Differential Privacy

Summary of Modern Developments

Data Anonymization Approach:

- k-Anonymity (Groups of k indistinguishable individuals)
- l-Diversity (Groups of l different sensitive values)
- t-Closeness (Groups reflecting overall distribution)
- Resampling / Compression (Random data reflecting the real-world distribution)
- Permutation (deassociating individual and sensitive data)

Output Perturbation Approach:

- ϵ-Differential Privacy

Structure

Challenges and Current Research Topics

How to ensure privacy concerning...

- Distributed Sources?
  - Secure Multiparty-Computation
- Tracking data? (RFID, Cellphones, Credit Card usage, ...)
  - Quite stable against perturbation due to high dimensionality
  - Causes of combining multiple sources unforeseeable
- Genetic sequences?
  - Privacy risk underestimated
  - Potentially identifiable numerous generations later
- Social networks and interactions?
  - Extremely stable against perturbation

Challenges and Current Research Topics

How to ensure privacy concerning...

- Distributed Sources?
  - Secure Multiparty-Computation
- Tracking data? (RFID, Cellphones, Credit Card usage, ...)
  - Quite stable against perturbation due to high dimensionality
  - Causes of combining multiple sources unforeseeable
- Genetic sequences?
  - Privacy risk underestimated
  - Potentially identifiable numerous generations later
- Social networks and interactions?
  - Extremely stable against perturbation

Challenges and Current Research Topics

How to ensure privacy concerning...

- Distributed Sources?
  - Secure Multiparty-Computation
- Tracking data? (RFID, Cellphones, Credit Card usage, ...)
  - Quite stable against perturbation due to high dimensionality
  - Causes of combining multiple sources unforeseeable
- Genetic sequences?
  - Privacy risk underestimated
  - Potentially identifiable numerous generations later
- Social networks and interactions?
  - Extremely stable against perturbation

Challenges and Current Research Topics

How to ensure privacy concerning...

- Distributed Sources?
  - Secure Multiparty-Computation
- Tracking data? (RFID, Cellphones, Credit Card usage, ...)
  - Quite stable against perturbation due to high dimensionality
  - Causes of combining multiple sources unforeseeable
- Genetic sequences?
  - Privacy risk underestimated
  - Potentially identifiable numerous generations later
- Social networks and interactions?
  - Extremely stable against perturbation

Thanks. Questions?

Sources and

References

Aggarwal, Charu C.:

On k-anonymity and the curse of dimensionality.
In: Proceedings of the 31st international conference on Very large data bases, VLDB Endowment, 2005 (VLDB '05). –
ISBN 1–59593–154–6, 901–909

Adam, Nabil R. ; Worthmann, John C.:

Security-control methods for statistical databases: a comparative study.
In: ACM Comput. Surv. 21 (1989), Dezember, Nr. 4, 515–556.
http://dx.doi.org/10.1145/76894.76895. –
DOI 10.1145/76894.76895. –
ISSN 0360–0300

Denning, Dorothy E.:

Secure statistical databases with random sample queries.
In: ACM Trans. Database Syst. 5 (1980), September, Nr. 3, 291–315.
http://dx.doi.org/10.1145/320613.320616. –
DOI 10.1145/320613.320616. –
ISSN 0362–5915

Dwork, Cynthia ; McSherry, Frank ; Nissim, Kobbi ; Smith, Adam:

Calibrating noise to sensitivity in private data analysis.
In: Proceedings of the Third conference on Theory of Cryptography.
Berlin, Heidelberg : Springer-Verlag, 2006 (TCC'06). –
ISBN 3–540–32731–2, 978–3–540–32731–8, 265–284

Fung, Benjamin C. M. ; Wang, Ke ; Chen, Rui ; Yu, Philip S.:

Privacy-preserving data publishing: A survey of recent developments.
In: ACM Comput. Surv. 42 (2010), Juni, Nr. 4, 14:1–14:53.
http://dx.doi.org/10.1145/1749603.1749605. –
DOI 10.1145/1749603.1749605. –
ISSN 0360–0300

Hansen, Morris H.:

Insuring confidentiality of individual records in data storage and retrieval for statistical purposes.
In: Proceedings of the November 16-18, 1971, fall joint computer conference.
New York, NY, USA : ACM, 1971 (AFIPS '71 (Fall)), 579–585

Iyengar, Vijay S.:

Transforming data to satisfy privacy constraints.
In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining.
New York, NY, USA : ACM, 2002 (KDD '02). –
ISBN 1–58113–567–X, 279–288

Kobie, Nicole:

EE data being sold to track customers.
http://www.pcpro.co.uk/news/381766/ee-data-being-sold-to-track-customers, 2013

Qardaji, Wahbeh:

Differentially Private Publishing of Geospatial Data.
http://www.cerias.purdue.edu/news_and_events/events/security_seminar/details/index/9nr7je9aqbqneqem9hrg2salic, 2013

Samarati, Pierangela ; Sweeney, Latanya:

Generalizing data to provide anonymity when disclosing information (abstract).
In: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems.
New York, NY, USA : ACM, 1998 (PODS '98). –
ISBN 0–89791–996–3, 188–