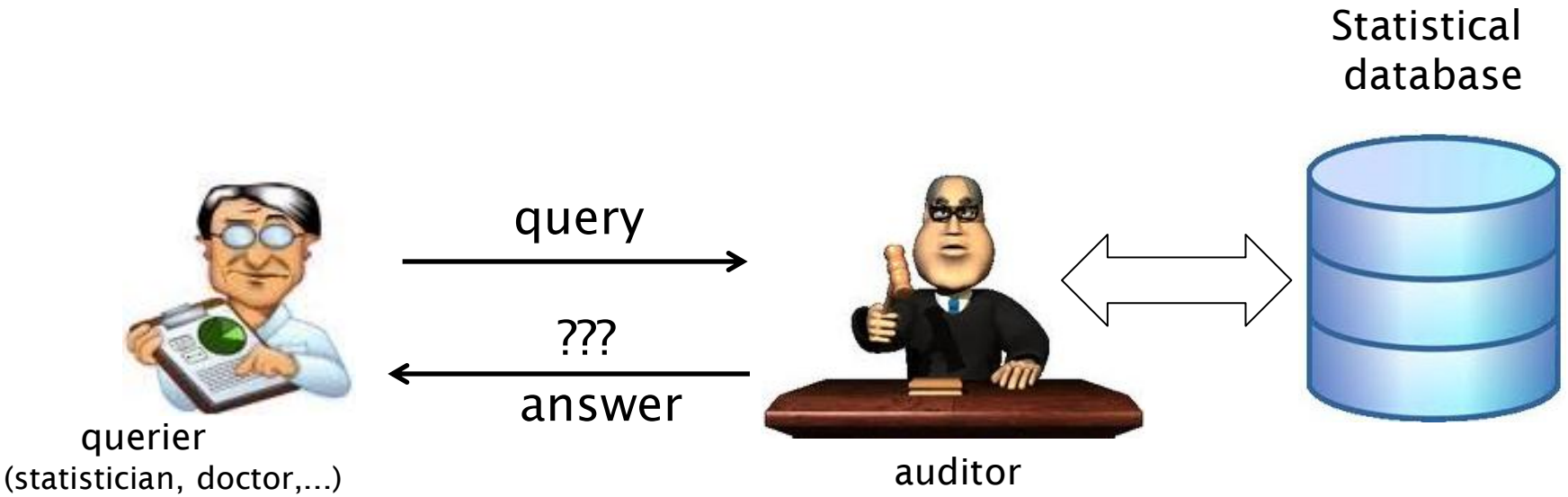
Query Auditing for Protecting Sensitive Attributes in Statistical Databases

Vinh-Thong Ta
INRIA Lyon, PRIVATICS Team
vinh-thong.ta@inria.fr

CAPPRIS Meeting, March 18-19, 2014, Paris

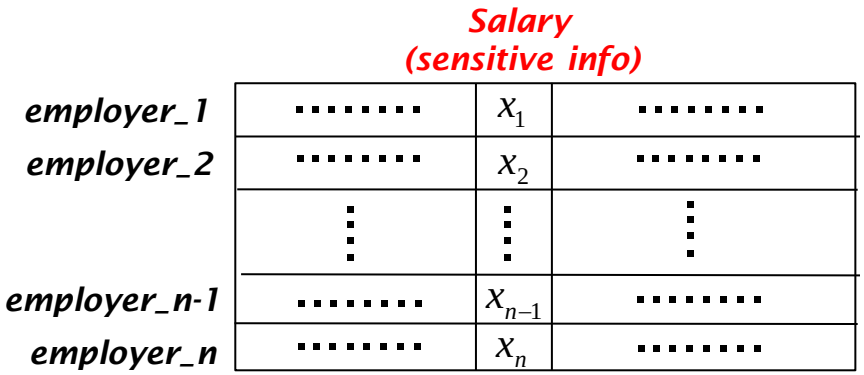
Query Auditing



Prevent or detect unintentional disclosure of sensitive information after a series of queries and corresponding answers

Typical Setting

- ***n*** denotes the total number of records in the DB
- ***X = {x1, x2, …, xn}*** are the sensitive attribute values in the records.
- ***q = (Xq, f)*** is an aggregate query, where
  - ***Xq*** specifies a subset of records, called the query set
  - ***f*** is aggregation function such as MAX, MIN, SUM, AVG, MEDIAN.
- ***a = f(Xq)*** is the result of applying *f* to *Xq*.



E.g., q = ({x1, x2}, SUM)

Offline vs. Online Auditing

- Offline auditing: Detection
  - Given *t* queries ***q1,..., qt*** and their answers ***a1,..., at*** *over* ***X***
  - **Goal**: determine if sensitive information has been disclosed.

$$
\begin{aligned}
q1: &\quad xi + \ldots + xn = a1 \\
q2: &\quad xj + \ldots. + xm = a2 \\
&\ldots \\
qt\text{-}1: &\quad xk + \ldots + xp = at\text{-}1 \\
qt: &\quad x1 + x2 = at
\end{aligned}
$$

can the value of ***xi*** *be determined?*

Offline vs. Online Auditing

- Online auditing: Prevention (and Detection)
  - Given *(t-1)* queries ***q1,..., qt-1*** and their *(t-1)* answers ***a1,..., at-1*** *over* ***X***
  - **Goal:** determine if answering the new query ***qt*** would cause disclosure of sensitive information, and in case yes it denies to answer.

$$
\begin{aligned}
&q1: \quad xi + \ldots + xn = a1 \\
&q2: \quad xj + \ldots. + xm = a2 \\
&\ldots \\
&qt\text{-}1: \quad xk + \ldots + xp = at\text{-}1
\end{aligned}
$$

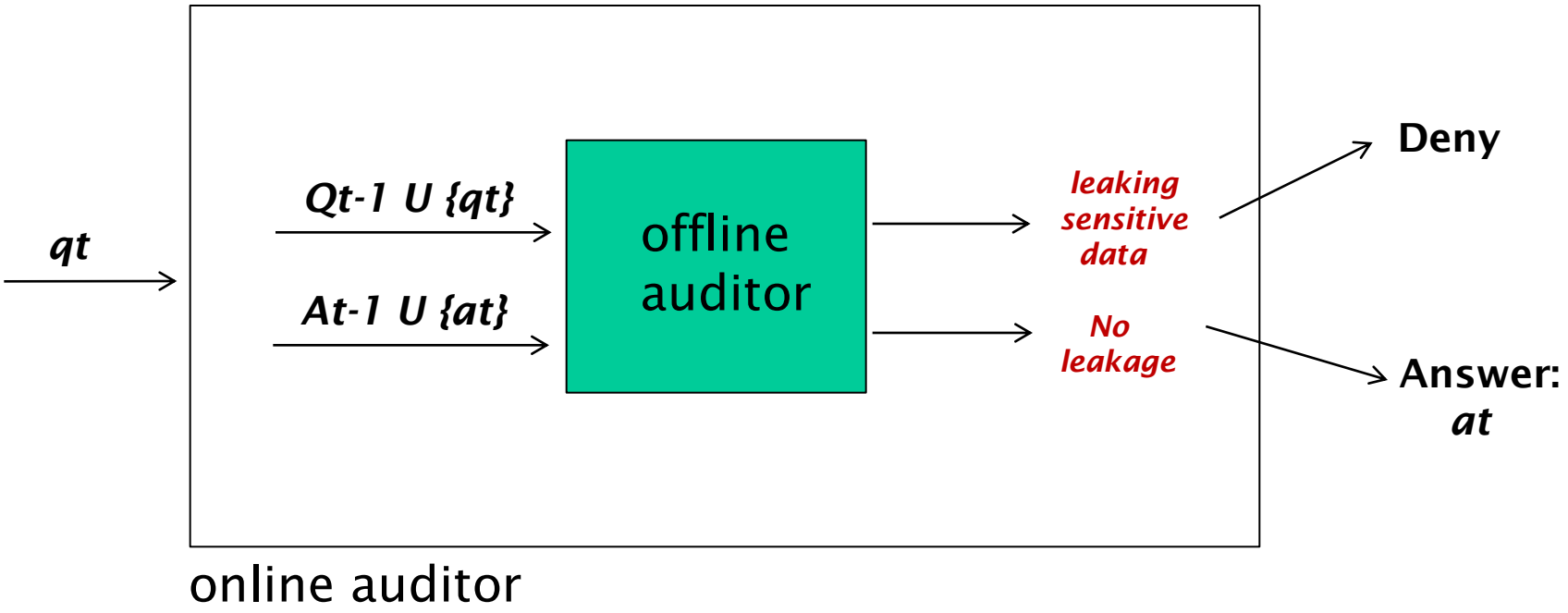
no value of ***xi*** *can be determined yet.*

$$qt: \quad x1 + x2$$

Should we provide the answer at ?

Can we apply an offline auditor directly to solve the on-line auditing problem?

- *Past queries* ***Qt-1 = {q1,..., qt-1}*** and the corresponding ***At-1 = {a1,..., at-1}*** *over set* ***X.***
- Determine whether to answer the *new* query ***qt.***



Can we apply an offline auditor directly to solve the on-line auditing problem?

This approach does not work in general, because **denials also leak information**!

Example:

- let n = 3 and X = {5, 5, 5}, SUM and MAX queries are allowed.
- let q1 = SUM(x1,x2,x3) then A = {15}
- let q2 = MAX(x1,x2,x3) → this is ***denied,***
  - because answering with 5 would lead to leakage: ***x1=x2=x3= 5***.

- however, based on the deny it can be logically deduced that ***x1=x2=x3= 5***
  - MAX(x1, x2, x3) cannot be smaller than 5, otherwise the SUM cannot be 15
  - if MAX(x1, x2, x3) > 5 then the query would have been answered

  Hence: MAX(x1, x2, x3) must be 5

Another bad approach that applies offline auditor

Deny whenever the offline auditor does, and in addition, *randomly deny some queries* that would normally be answered by offline auditor.

- Now denials leak less information, but leakage is not generally prevented
- The auditing algorithm needs to remember which queries were randomly denied, since otherwise an attacker can repeatedly pose the same query until it is answered
- A difficulty is then to define whether two queries are equivalent

Better Approach: Simulatable Online Auditor

Crucial observation:

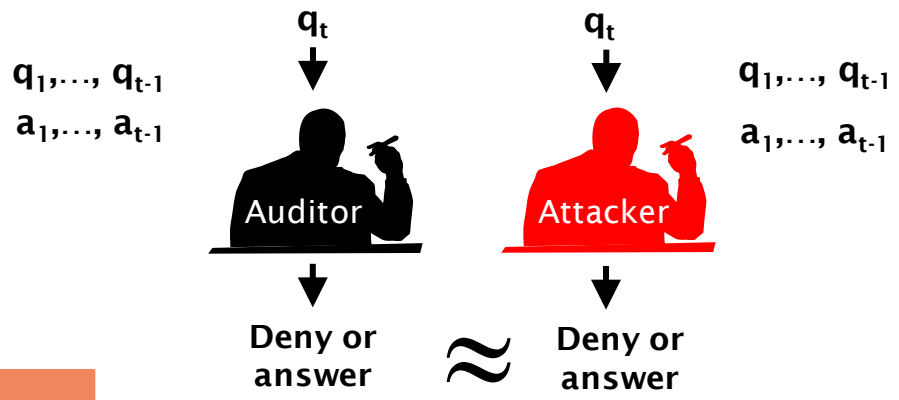
- query denials have the potential to leak information *if when deciding to deny, the auditor uses information that is unavailable to the attacker* (i.e., the answer ***at*** to the current query ***qt***)

The idea behind simulatable auditors:

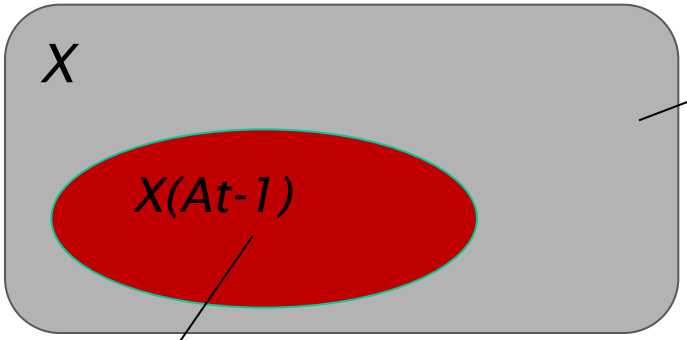
- the attacker should be able to ***simulate or mimic*** the auditors decisions to answer or deny a given query.

- as the attacker can equivalently determine for himself when his queries will be denied, *denials leak no more information than what the attacker already knows*.

→ ***provable privacy***



A sufficient condition for simulatability



Set of all possible data sets
i.e. all possible values of (x1,..., xn)
(from the attacker's point of view)

Set of all data sets consistent with past answers, *At-1 = {a1, ..., at-1}*
(from the attacker's point of view)

If there is **one** *Xi* in *X(At-1)* that leads to data leakage then the auditor denies ***qt***, otherwise it answers with ***at***.

Simulatable Auditing Example

Example revisited:

- let $n = 3$ and $X = \{x1, x2, x3\}$
- $q1 = (\{x1, x2, x3\}, SUM)$, which can be responded, $a1 = 15$.
- then $q2 = (\{x1, x2, x3\}, MAX)$ should ***always*** be denied

(even if for the values (2, 5, 8) of (x1, x2, x3), it would be safe to respond *q2*)

Because the data set *(5, 5, 5)*, consistent with *a1=15,* would lead to data leakage. This is also known by attacker.

x1	2
x2	5
x3	8

database

☹ Simulatability has bad usability

- deny too much

Full disclosure vs. Partial disclosure

Full disclosure model

- A value of ***xi*** is fully disclosed by (Q, A) if it can be **uniquely** determined *(i.e., xi is the same in all possible data sets consistent with (Q,A))*

Partial disclosure model

- ***xi*** cannot be uniquely determined, but it may be deduced to lie in a tiny interval, or in a large interval with a heavily skewed distribution.

The auditor is randomized
- it's decision to answer or deny needs not be deterministic.
- ***xi*** is drawn from some distribution *D on* $(-\infty,\infty)^n$ known to both the attacker and the auditor

State-of-the-art

Auditing/Disclosure	Online Auditing	Offline Auditing	Handling Update
Probabilistic Disc.	Sim. MAX, MIN, MAX & MIN - real, unbound values Sim. SUM - real, unbound values		
Full Disc.	Sim. MAX, MIN, MAX & MIN - real, unbound values Sim. SUM - real, unbound values	MAX & SUM (NP-hard) - real, unbound values SUM - real, unbound values - boolean values MAX, MIN, MAX & MIN - real, unbound values	MAX, MIN, MAX & MIN - delete, modify, insert SUM - delete, modify, insert
Interval Disc.		SUM (real, unbound values)	

TABLE I

SUMMARY OF QUERY AUDITING PROBLEMS AND RELATED WORKS. THE ABBREVIATION SIM. MEANS SIMULATABLE.

Challenges and Open questions

- While there has been some investigation into auditing SUM, MAX, MIN, MEDIAN queries, intermingling these queries has proven to be a greater challenge.
- Auditing SQL style, Select-Project-Join queries
- Existing works are concerning with protecting individual values (e.g., salary of one person), it would be interesting to protect aggregated values (e.g., MAX, MIN).
- Collusion is a largely unaddressed issue in most interactive data sharing mechanisms today. Users can cooperate and share info.
- Utility, usability measurement
  - e.g; how to define and measure?

Our work

- While there has been some investigation into auditing SUM, MAX, MIN, MEDIAN queries, intermingling these queries has proven to be a greater challenge.
- Auditing SQL style, Select-Project-Join queries
- Existing works are concerning with protecting individual values (e.g., salary of one person), it would be interesting to protect aggregated values (e.g., MAX, MIN).
- Collusi
  sharing
- Utility,
  - e.g; how to define and measure?

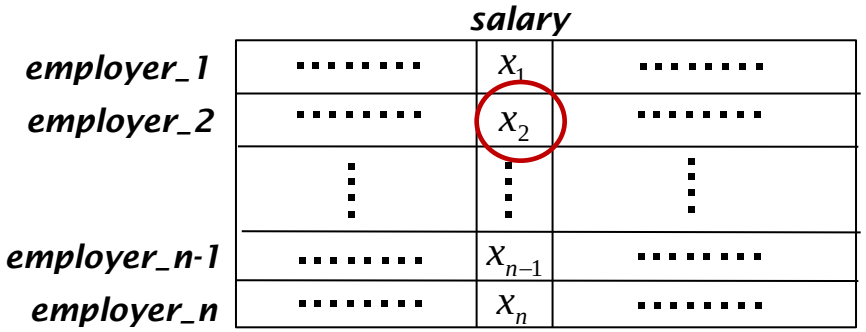
Vinh-Thong Ta and Levente Buttyán. *Query Auditing for Protecting Max/Min Values of Sensitive Attributes in Statistical Databases.* In 9th International Conference on Trust, Privacy, Security in Digital Business (Trustbus 2012), pp. 171-186, Springer LNCS, July 2012.

Related Works vs. Our Work

Related works:

- Detect or prevent the disclosure of the sensitive fields of ***individual records*** in the database
  - e.g., the salary of a given employee (x_i)

· $x_i \in (-\infty, \infty)$, real number



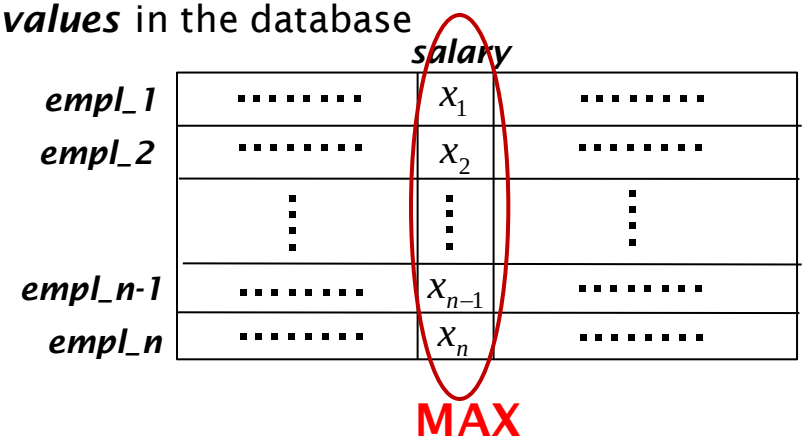
Our work:

- No solution proposed previously

•Detect or prevent the disclosure of ***aggregate values*** in the database
  - •e.g., the maximum salary, MAX(x_1,...,x_n)

· $x_i \in [\alpha, \beta]$, real number

→ yields new problems which cannot be solved with existing methods!!!



The motivation behind our work

CHIRON
Cyclic and person-centric Health management

- Use body mounted WSNs to collect medical data from a patient
  - e.g., ECG signals, blood pressure measurements, temperature samples
- Use a personal device (e.g., a smart phone) to collect data
- Provide controlled access to the data for external parties
  - e.g., hospital personnel, personal coach services, and health insurance companies

- The records in database all belong to the same patient.
- Individual values (i.e., sensor readings) may not be sensitive,
- **Aggregates computed over those values** can disclose the health status of the patient
  - e.g., the maximum of the blood pressure in a given time interval

- Some of the accessing parties (e.g., health insurance companies) should be prevented to learn that information.

Contributions

- Query: $AVG(Q),\ Q \subseteq \{x_1,...,x_n\},\ \forall x_i \in [\alpha, \beta]$
- Goal: detecting/preventing disclosure of $MAX\{x_1,...,x_n\}$ (MIN)

- We proposed three query aufitors for three different settings
  - An offline query auditor for the full dislosure model
    - Based on Linear optimization problem
  - An online query auditor for the full disclosure model
    - Based on Linear optimization problem
  - An online simulatable auditor for the partial disclosure model
    - Application of the random sampling method proposed by other researchers (Lovász et. al.).

- In each case, we proved that our proposed query auditor is secure, and detects/prevents the MAX (MIN) value from disclosure.
- We also showed that they are polynomial-time algorithms.

Conclusions and future directions

- Query auditing is a broadly investigated problem, and relevant in database data protection.

- Investigating the case of combined queries (e.g., MAX & MIN & SUM).
- Investigating the case of SQL style queries.
- Addressing the protection of other types of aggregated values than MAX and MIN.
- Examining the impact of collusion attackers.
- Defining and measuring the precise degree of utility, usability.
- Proposing something with better usability than simulatable auditing.

Offline $Auditor^{max}_{avg}$ in the Full Disclosure Model

t queries
$b_{i,j} \in \{0,1\}$

$$\bar{A} = \begin{pmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,n} \\ b_{2,1} & b_{2,2} & \dots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{t,1} & b_{t,2} & \dots & b_{t,n} \end{pmatrix}, \bar{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_t \end{pmatrix}$$

corr. answers

$$L_j = \begin{cases} \alpha \le x_i \le \beta, \forall x_i : x_i \in \{x_1, \dots, x_n\} \\ \overline{Ax} = \bar{a}, where\ \bar{x} = (x_1, \dots, x_n)^T \\ ob_j = maximize\,(x_j) \end{cases} F$$

Lin. Eq. System
(Feasible set)

We have n linear programming problems P_j, $j \in \{1, \dots, n\}$

Maximum of $x_1, \dots, x_n$: $\max\{ob_1, \dots, ob_n\}$

problem P

Simulatable Online $Auditor^{\max}_{avg}$ in the Partial Disclosure Model

λ-safe :
A sequence of queries and answers, *q1, q2, …, qt* and *a1, a2, …, at* is said to be **λ-safe** with respect to *MAX* and an interval $J \subseteq [\alpha, \beta]$ if

$$Safe_{\lambda,J}(q_1,...,q_t,a_1,...,a_t) =$$

$$\begin{cases} 1, & \frac{1}{1+\lambda} \le \frac{\Pr_D(MAX \in J \mid \wedge_{j=1}^{t}(avg(\mathrm{Q_j}) = a_j))}{\Pr_D(MAX \in J)} \le 1+\lambda \\ 0, \text{otherwise} \end{cases}$$

i.e., the attacker's confidence that $MAX \in I$ *does not change significantly upon seeing the queries and answers.*

AllSafe : $AllSafe_{\lambda,\omega}(q_1,...,q_t,a_1,...,a_t) =$

$$\begin{cases} 1, \text{ if } Safe_{\lambda,J}(q_1,...,q_t,a_1,...,a_t), \forall J \\ 0, \text{otherwise} \end{cases}$$

$$\text{where interval J is } \omega - \text{significant, namely, } P(MAX \in J) \ge \frac{1}{\omega}$$

Simulatable Online $Auditor_{avg}^{\max}$ in the Partial Disclosure Model

$(\lambda, \omega, T) - \text{privacy game}$: there are up to T rounds

in each round t:

- the attacker (adaptively) poses a query $q_t = (\mathrm{Q}_t, AVG)$
- the auditor determines whether *qt* should be answered; the auditor responds with $a_t = AVG_X(\mathrm{Q}_t)$ if qt is allowed, and *"denies"* otherwise
- the attacker wins if $AllSafe_{\lambda,\omega}(q_1,...,q_t,a_1,...,a_t) = 0$

$(\lambda, \omega, T, \delta) - \text{private auditor}$:

for any attacker A

$$\Pr\{\text{A wins the } (\lambda, \omega, \mathrm{T})\text{- privacy game}\} \leq \delta$$

where the probability is taken over the distribution D that the data comes from and the coin tosses of the randomized auditor and the attacker