Query Auditing

Vitaly Shmatikov
Reading Assignment

◆ Read Kenthapadi, Mishra, Nissim. “Simulatable Auditing” (PODS 2005).
Query Audit Problem

- Maintaining “privacy” of data

Q₁ Q₂ ... Qₙ → ... → Q → A₁ A₂ ... Aₙ

Auditor

Database

Does answer to Q combined with answers to Q₁,...,Qₙ reveal something?

A or “Denied”
Variations of the Problem

Specifies subset of the variables

Q₁ Q₂ ... Qₙ

A₁ A₂ ... Aₙ

Wants to learn value of some variable

Min, max, median, sum, average, or count of specified subset

List of real, integer, or Boolean values

Auditor

Database
Offline vs. Online

- **Offline auditing**
  - Given a collection of queries and answers to them, check whether anything “forbidden” was revealed
  - Detects privacy breaches after the fact

- **Online auditing**
  - Queries are presented to auditor one at a time; auditor checks if answering the current query (in combination with past answers) reveals “forbidden” information
  - Prevents privacy breaches on-the-fly

- Is there a difference?
Auditing Sum Queries on Booleans

- **Database**: collection of secret Boolean variables
- **Query**: specifies subset $S$ of variables
- **Answer**: sum of variables in $S$
- **Privacy breach**: after asking several queries, user learns the value of some secret variable(s)
- **Auditing problem**: given a set of Boolean equations, is there a variable that has the same value in all solutions?
  - Weaker version: does system have a unique solution?
Why Is This Interesting?

- Query can be safe on real-valued, unbounded data, but reveal information when the data are discrete, with known bounds

\[
\begin{align*}
  x + y + w &= 1 \\
  y + z &= 1 \\
  x + z &= 1
\end{align*}
\]

Real: multiple solutions, secure
Boolean: unique solution, insecure (why?)
Issues with Bounded Data

- Traditional query auditing: does the given set of queries compromise security for some values of the variables?
  - ... as opposed to their actual values in the database
- With bounded data, the answer is always Yes
  - “Sum of subset” Boolean query always reveals whether variables are all equal to 1
    - For example, if subset = \{x,y\}, then the fact that x+y=2 will reveal that x=y=1
- This suggests that auditor should consider actual values in the database
Approximate Auditing

- For a query set, answer only when it is safe; otherwise deny query
  - Conservative: a safe query may be denied

- Given Boolean variables $x_1 \ldots x_n$ and query sets $S_1 \ldots S_m$, let trace of $x_i$ $T(x_i) = \{ p: x_i \in S_p \}$

- Theorem [KPR]: If for every variable $x_i$, there is a variable $x_j$ s.t. $x_i = 1-x_j$ and $T(x_i)=T(x_j)$, then no variable is revealed by answers to $S_1 \ldots S_m$
  - Intuition: if values of $x_i$ and $x_j$ were switched, the answers to queries would have been the same
Max Queries on Reals

- **Database**: collection of real-valued variables
- **Query**: specifies subset $S$ of variables
- **Answer**: maximum over variables in $S$
- **Privacy breach**: after asking several queries, user learns the value of some secret variable(s)
Define $m_i = \min_S\{ \max(S_p) : i \in S_p \}$
- Suppose $S_1=\{1,2\}$, $\max(S_1)=9$; $S_2=\{1,3\}$, $\max(S_2)=4$
- Then $m_1=\max(S_2)$
- Intuition: among all queries that include variable $y_i$, $m_i$ is the query that gives the minimum answer
  - Call this query $i$-extreme

Theorem [KPR]: The value of a variable $i$ is determined if and only if there exists a query $S_p$ that is $i$-extreme but is not $l$-extreme for any $l \neq i$
- Intuition: $y_i \leq m_i$ (by definition). If $S_p$ is $i$-extreme but not $l$-extreme, then for all variables $l$, $y_l < m_i$, so $y_i=m_i$
Auditing in a Nutshell

“Denied” if answering $Q_{i+1}$ would cause a privacy breach
Nissim’s Example: Sum/Max

- Variables $d_i$ are real, privacy breached if adversary learns some $d_i$

  Gimme sum($d_1,d_2,d_3$)  \
  Answer=15  \
  Gimme max($d_1,d_2,d_3$)  \
  "Denied"

  Oh well
  Wait... there must be a reason why second query was denied
  The only possible reason for denial is if $d_1=d_2=d_3=5$
Nissim’s Example: Intervals

\[ d_i \in [0,100], \] privacy breached if adversary learns some \( d_i \pm 1 \)

First query denied
\[ \Rightarrow d_1, d_2 \in [0,1], \text{ or } d_1, d_2 \in [99,100] \]

But
\[ d_2 + d_3 = 50, \text{ so } d_2 < 99 \]

\[ d_1, d_2 \in [0,1], \text{ and } d_3 \in [49,50] \]
Sounds Familiar?

Colonel Oliver North, on the Iran-Contra arms deal

“On the advice of my counsel I respectfully and regretfully decline to answer the question based on my constitutional rights.”

David Duncan, former auditor for Enron and partner in Arthur Andersen

“Mr. Chairman, I would like to answer the committee's questions, but on the advice of my counsel I respectfully decline to answer the question based on the protection afforded me under the Constitution of the United States.”
Two Problems

◆ Obvious problem: denied queries ignored
  • Algorithmic problem: not clear how to incorporate denials in the audit decision

◆ Subtle problem: denials leak information!

Possible assignments to \{d_1,\ldots,d_n\}

Assignments consistent with \((q_1,\ldots,q_i; a_1,\ldots,a_i)\)

\[q_{i+1} \text{ denied}\]
When Do Denials NOT Leak Info?

An auditor is simulatable if there exists a simulator such that...

- Database: $q_1, \ldots, q_i$
- Auditor: $q_i \rightarrow q_{i+1}$
- Deny or answer

How do we know that this auditor does not leak data values?

- Simulator: $q_1, \ldots, q_i, a_1, \ldots, a_i$
- Deny or answer

$\approx$
Simulatable Auditing

Possible assignments to \( \{d_1, \ldots, d_n\} \)

Assignments consistent with \( (q_1, \ldots, q_i, a_1, \ldots, a_i) \)

\( q_{i+1} \) denied/allowed
Summary

◆ Auditing decisions can leak information
  • Denials can reveal sensitive data!
◆ Simulatable auditors provably don’t leak information about actual data values
◆ There are many alternatives to query auditing
  • Add random noise to data and/or perturb answers
  • Cryptographic techniques such as secure multi-party computation