Audit Games

Anupam Datta
CMU
Fall 2014
Detecting Privacy Violations

Privacy Policy → Organizational audit log

Complete formalization of HIPAA Privacy Rule, GLBA

Computer-readable privacy policy

Automated audit for black-and-white policy concepts

Detect policy violations

Oracles to audit for grey policy concepts
Audit algorithms suggest cases for resource-constrained human auditors to investigated
Audit in Practice

• FairWarning: popular tool for auditing in hospitals

• Provides heuristics to guide human effort
  – Inspect all celebrity record accesses

<table>
<thead>
<tr>
<th>Inspections</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Audit Games:
Resource Allocation for Human Auditors
Regret Minimizing Audits
Byzantine Adversary Model
Model/Algorithm by Example

Auditing budget: $3000/ cycle
Cost for one inspection: $100
Only 30 inspections per cycle
Employee incentives unknown

Auditor

Access divided into 2 types
30 accesses
70 accesses
100 accesses

Loss from 1 violation (internal, external)
$500, $1000
$250, $500

Audit loss
Violation cost

Byzantine Model
Audit Algorithm Choices

Consider 4 possible allocations of the available 30 inspections

<table>
<thead>
<tr>
<th>Weights</th>
<th>0</th>
<th>10</th>
<th>20</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30</td>
<td>20</td>
<td>10</td>
<td>0</td>
</tr>
</tbody>
</table>

Choose allocation probabilistically based on weights

Byzantine Model
Audit Algorithm Run

<table>
<thead>
<tr>
<th>No. of Access</th>
<th>Actual Violation</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>2</td>
</tr>
<tr>
<td>70</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Int. Caught</th>
<th>Ext. Caught</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Updated weights: 0.5, 0.5, 2.0, 1.5

Learn from observed and estimated loss

Observed Loss: $2000, $1500, $1000, $1000
Estimated Loss: $750, $1000, $1250, $1500

Updated weights: 0.5, 0.5, 2.0, 1.5

Learn from observed and estimated loss
Byzantine model

- $k$ types of target
  - $\vec{n} = n_1, \ldots, n_k$ targets
  - $\vec{s}$ inspections, $\vec{v}$ violations
  - $\vec{\Omega}$ violations – parameterized by $\vec{n}, \vec{s}, \vec{v}$
  - Fixed probability $p$ of external detection

- Defender action - Inspections: $\vec{s}$ chosen at random

- Adversary action - Violations: $\vec{v}, \vec{n}$

- Repeated game
  - Rounds correspond to audit cycle
Utilities

\[ U(\vec{s}, \vec{O}) = \sum_k U_1(s_k) + \sum_k U_2(O_k) \]
- Audit Cost
- Violation Cost

• Average utility over \( T \) rounds
  \[ = \frac{1}{T} \sum_{t=1}^{T} U(\vec{s}^t, \vec{O}^t) \]

• Adversary utility unknown
Regret by Example

**Total Payoff**

- **Emp**: $6
- **Org**: $5

**Players**

- **Emp**
- **Org**: $s$

**Org**: $s_1$

**Round 1**

- **3, 1**
- **2 ($6)**

**Round 2**

- **3, 2**
- **1 ($0)**

**Strategy**: outputs an action for every round

\[ \text{Total Regret}(s, s_1) = -5 - (-6) = 1 \]

\[ \text{regret}(s, s_1) = \frac{1}{2} \]
Meaning of Regret

- Low regret of $s$ w.r.t. $s_1$ means $s$ performs as well as $s_1$

- Desirable property of an audit mechanism
  - Low regret w.r.t. a set of strategies $S$
  - $\max_{s' \in S} \text{regret}(s, s') \to 0$ as $T \to \infty$
Regret Minimizing Algorithm

$w_s = 1$ for all strategies $s$

Find AWAKE

New audit cycle starts.

Pick $s$ in AWAKE with probability $D_t(s) \propto w_s$

Pay using $\text{Pay}(s)$

Violation caught; obtain payoff $\text{Pay}(s)$

Estimate payoff vector

Update weight* of strategies $s$ in AWAKE

$* w_s \leftarrow w_s \cdot \gamma^{-\text{Pay}(s)} + \gamma \sum_{s'} D_t(s') \text{Pay}(s')$
Audit Algorithm Choices

Consider 4 possible allocations of the available 30 inspections

Weights

Choose allocation probabilistically based on weights
## Audit Algorithm Run

<table>
<thead>
<tr>
<th>No. of Access</th>
<th>Actual Violation</th>
<th>Observed Loss</th>
<th>Estimated Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>2</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>70</td>
<td>4</td>
<td>30</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Int. Caught</th>
<th>Ext. Caught</th>
<th>Actual Loss</th>
<th>Estimated Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>$2000</td>
<td>$1000</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>$750</td>
<td>$1500</td>
</tr>
</tbody>
</table>

Updated weights:
- Observed: 0.5, 0.5, 2.0, 1.5

Learn from observed and estimated loss

---

**Byzantine Model**
Guarantees of RMA

• With probability $1 - \varepsilon$ RMA achieves the regret bound

$$2\sqrt{\frac{2 \log(N)}{T}} + \frac{2 \log(N)}{T} + 2\sqrt{\frac{2 \log(4N/\varepsilon)}{T}}$$

– $N$ is the set of strategies
– $T$ is the number of rounds
– All payoffs scaled to lie in $[0,1]$

• Better bound than existing algorithm (under mild assumptions)
Audit Games
Rational Adversary Model
Simple Rational Model

- Adversary commits one violation
- If a violation is detected, adversary is fined $x$
- Utility when target $t_i$ is attacked
  - $p_i U_{a,D}(t_i) + (1 - p_i)U_{u,D}(t_i) - a_0 x$
  - $p_i (U_{a,A}(t_i) - x) + (1 - p_i)U_{u,A}(t_i)$

Utility when audited  Utility when unaudited
Stackelberg Equilibrium Concept

- Defender commits to a randomized resource allocation strategy ($p_i$’s and $x$)
- Adversary plays best response to that strategy
- For defender Stackelberg better than Nash eq.

- Goal
  - Compute optimal defender strategy
Computing Optimal Defender Strategy

Solve optimization problems $P_i$ for all $i \in \{1, \ldots, n\}$ and pick the best solution

$$\max p_i U_{a,D}(t_i) + (1 - p_i)U_{u,D}(t_i) - a_0x$$

subject to

$\forall j \in \{1, \ldots, n\}$

$$p_j \left( U_{a,A}(t_j) - x \right) + (1 - p_j)U_{u,A}(t_j) \leq$$

$$p_i \left( U_{a,A}(t_i) - x \right) + (1 - p_i)U_{u,A}(t_i)$$

and $p_i$'s lie on the probability simplex

and $0 \leq x \leq 1$
Special Case

• Assume punishment $x$ is a constant
• Corresponds to setting of physical security games
• Reduces to a set of linear programs (LPs)
  • Can be solved efficiently using an LP solver
Physical Security Games

• Game model for physical security (Tambe et al.)
  – LAX airport deployment
  – Air marshals deployment

• High level (basic) model
  – n targets defended by m resources
  – Stackelberg equilibrium
  – No punishments
Computing Optimal Defender Strategy

Solve optimization problems $P_i$ for all $i \in \{1, \ldots, n\}$ and pick the best solution

$$\max p_i U_{a,D}(t_i) + (1 - p_i) U_{u,D}(t_i) - a_0 x$$

subject to

$\forall j \in \{1, \ldots, n\}$

$$p_j (U_{a,A}(t_j) - x) + (1 - p_j) U_{u,A}(t_j) \leq p_i (U_{a,A}(t_i) - x) + (1 - p_i) U_{u,A}(t_i)$$

and $p_i$'s lie on the probability simplex

and $0 \leq x \leq 1$
Idea of Algorithm

• Transform problem of multiple variables into a problem of a single variable $x$
  • Express $p_j$’s in terms of $x$
  • Utility is a polynomial function of $x$

• Compute values of $x$ that maximize the utility function
Main Theorem

• *The problem can be approximately solved in polynomial time using an algorithm for computing roots of polynomials*
Details of Algorithm
Properties of Optimal Point

- Rewriting quadratic constraints

\[ p_j(-x - \Delta_j) + p_n(x + \Delta_n) + \delta_{j,n} \leq 0 \]

\[ \Delta_j = U_{u,A}(t_j) - U_{a,A}(t_j) \geq 0 \]

\[ \delta_{j,n} = U_{u,A}(t_j) - U_{u,A}(t_n) \]
Main Idea in Algorithm

- Iterate over regions, solve sub-problems \( EQ_j \)
  - Set probabilities to zero for curves that lie above & make other constraints tight
- Pick best solution of all \( EQ_j \)
Solving Sub-problem $EQ_j$

1. $p_j(-x - \Delta_j) + p_n(x + \Delta_n) + \delta_{j,n} = 0$
   - Eliminate $p_j$ to get an equation in $p_n$ and $x$ only

2. Express $p_n$ as a function $f(x)$
   - Objective becomes a polynomial function of $x$ only

3. Find $x$ where derivative of objective is zero & constraints are satisfied
   - Local maxima

4. Find $x$ values on the boundary
   - Found by finding intersection of $p_n = f(x)$ with the boundaries
   - Other potential points of maxima

5. Take the maximum over all $x$ values from steps 3,4
Audit Games with Multiple Defender Resources
Rational Adversary Model
Rational Model

Auditors

$k$ Inspections

$n$ Targets

Adversary
Captures Real Scenarios

<table>
<thead>
<tr>
<th>All targets auditable by all inspections</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Localized auditing/ Audit by managers</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Localized auditing with central auditors</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image7.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Audit by managers with shared managers</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image8.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image9.png" alt="Image" /></td>
</tr>
</tbody>
</table>
## Summary of Results

<table>
<thead>
<tr>
<th>Model Features</th>
<th>FPT Approximation</th>
<th>FPTAS (under certain conditions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple defender resources</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Subset restriction</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Multiple (constant number) attacks</td>
<td>✓</td>
<td>?</td>
</tr>
<tr>
<td>Target-Specific punishments</td>
<td>✓</td>
<td>?</td>
</tr>
</tbody>
</table>
A resource-constrained auditor's interaction with an adaptive adversary can be formalized using game-theoretic models and audit algorithms can be designed that provably optimize the defender's utility function in these models against Byzantine and rational adversaries.

• Questions?