Audit Games

Anupam Datta
CMU
Fall 2014

Detecting Privacy Violations

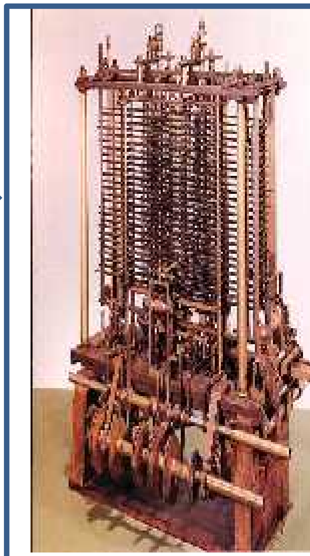
Privacy Policy

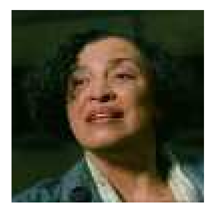
Organizational audit log

Complete formalization of HIPAA Privacy Rule, GLBA

$$\mathbf{G}\Big(\forall p_1, p_2, m.\ \mathrm{send}(p_1, p_2, m) \supset (\forall d, u, q, t.\ (m = \mathrm{info}(d, u)) \wedge \mathrm{contains}(m, q, t) \supset (\bigvee_i \varphi_i^{+}) \wedge (\bigwedge_j \varphi_j^{-})) \wedge (\forall t.\ (m = \mathrm{req\_for\_access}(p_1, t)) \supset \varphi \wedge \varphi)\Big)$$

Computer-readable privacy policy





Audit

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

	1	0	0	0
Inspections	0	0	1	0
	0	1	0	0

Audit Games:
Resource Allocation for Human Auditors

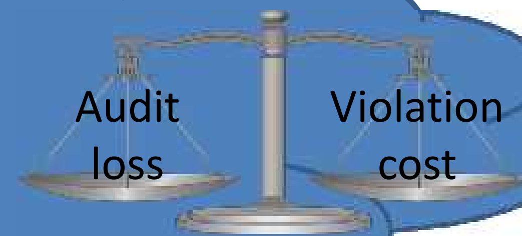
Regret Minimizing Audits

Byzantine Adversary Model

Model/Algorithm by Example

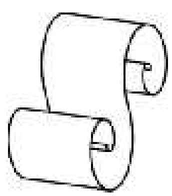
Auditor

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

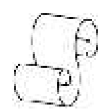


Access divided into 2 types

Loss from 1 violation (internal, external)

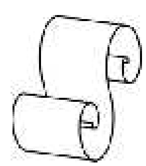


100 accesses

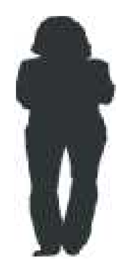


30 accesses

$500, $1000



70 accesses



$250, $500

Audit Algorithm Choices

Only 30 inspections

Consider 4 possible allocations
of the available 30 inspections

	0	10	20	30
	30	20	10	0
Weights	1.0	1.0	1.0	1.0

Choose allocation probabilistically based on weights

Byzantine Model

Audit Algorithm Run

No. of Access	Actual Violation
30	2
70	4


0	10	20	30
30	20	10	0

Observed Loss

Estimated Loss

Int. Caught	Ext. Caught
1	1
2	1


$2000	$1500	$1000	$1000
$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
  - $\vec{n} = n_1, \dots, n_k$ targets
  - $\vec{s}$ inspections, $\vec{v}$ violations
  - $\vec{\boldsymbol{O}}$ 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

- $\boldsymbol{U}(\vec{s},\overrightarrow{\boldsymbol{O}}) = \underbrace{\sum_k U_1(s_k)}_{\text{Audit Cost}} + \underbrace{\sum_k U_2(\boldsymbol{O}_k)}_{\text{Violation Cost}}$

- Average utility over T rounds $= \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{U}(\vec{s}^t,\overrightarrow{\boldsymbol{O}}^t)$

- Adversary utility unknown

Regret by Example

		Org 1	Org 2
Emp	3,1	$5	$6
	3, 2	$0	$5

Strategy: outputs an action for every round

$$Total\ Regret(s, s_1) = -5 - (-6) = 1$$

$$regret(s, s_1) = \frac{1}{2}$$

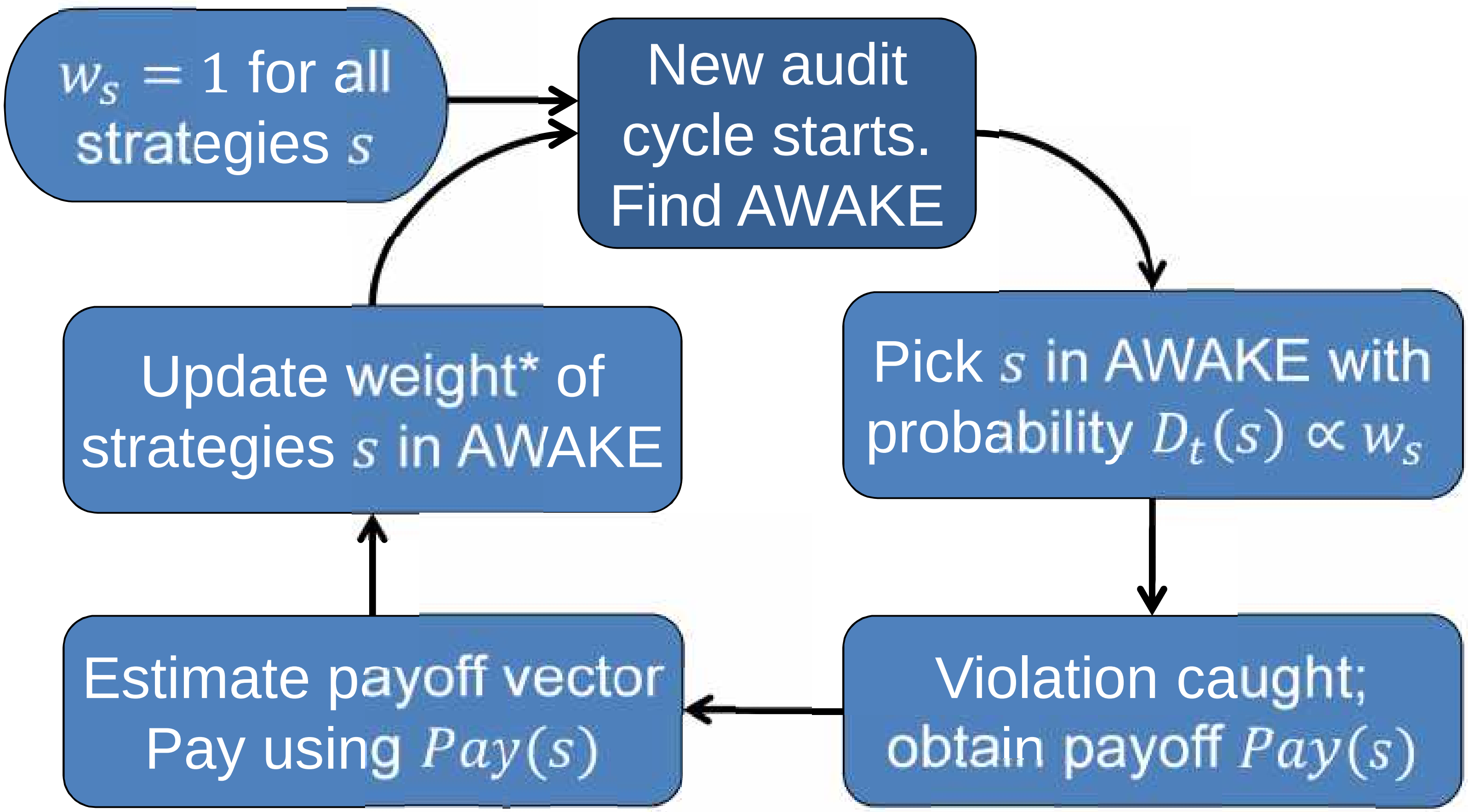
Players	Round 1	Round 2	Total Payoff
• Emp • Org: s	• 3,1 • 2 ($6)	• 3,2 • 1 ($0)	• Unknown • $6
Org : s_1	1 ($5)	1 ($0)	$5

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
  - $\max_{s' \in S} regret(s, s') \to 0 \; as \; T \to \infty$

Regret Minimizing Algorithm



$$^* \; w_s \leftarrow w_s \cdot \gamma^{-Pay(s)+\gamma \sum_{s'} D_t(s')Pay(s')}$$

Audit Algorithm Choices

Only 30 inspections

Consider 4 possible allocations
of the available 30 inspections

Sandra Bullock	0	10	20	30
Other	30	20	10	0
Weights	1.0	1.0	1.0	1.0

Choose allocation probabilistically based on weights

Audit Algorithm Run

No. of Access	Actual Violation
30	2
70	4

0	10	20	30
30	20	10	0

Observed Loss

Estimated Loss

Int. Caught	Ext. Caught
1	1
2	1

$2000	$1500	$1000	$1000
$750	$1250	$1250	$1500

Updated weights	0.5	0.5	2.0	1.5

Learn from observed and estimated loss

Guarantees of RMA

- With probability $1 - \epsilon$ RMA achieves the regret bound

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

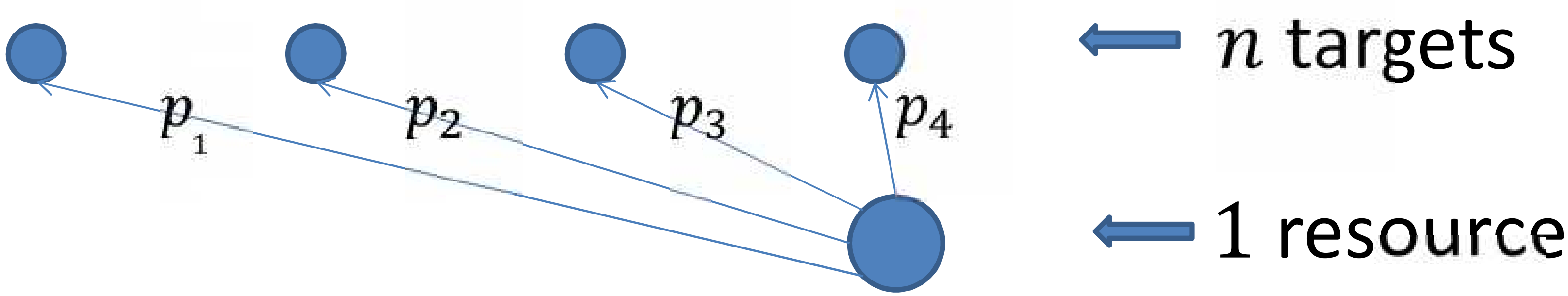
  - N is the set of strategies
  - T is the number of rounds
  - All payoffs scaled to lie in [0,1]

- 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 $i \in \{1, \dots, n\}$ 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

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

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

and p_i's lie on the probability simplex

and $0 \leq x \leq 1$

Quadratic

Non-convex

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 $i \in \{1, \dots, n\}$
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

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

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

and p_i's lie on the probability simplex

and $0 \leq x \leq 1$

Quadratic
Non-convex

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

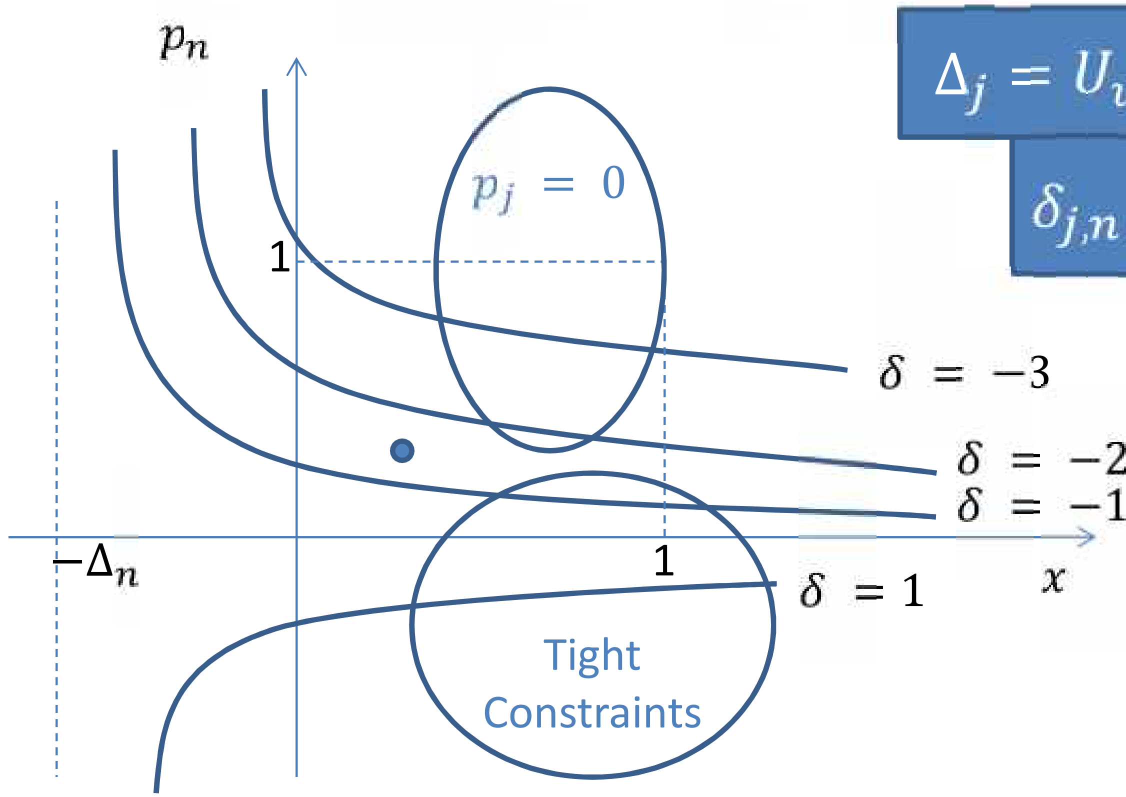
Properties of Optimal Point

- Rewriting quadratic constraints

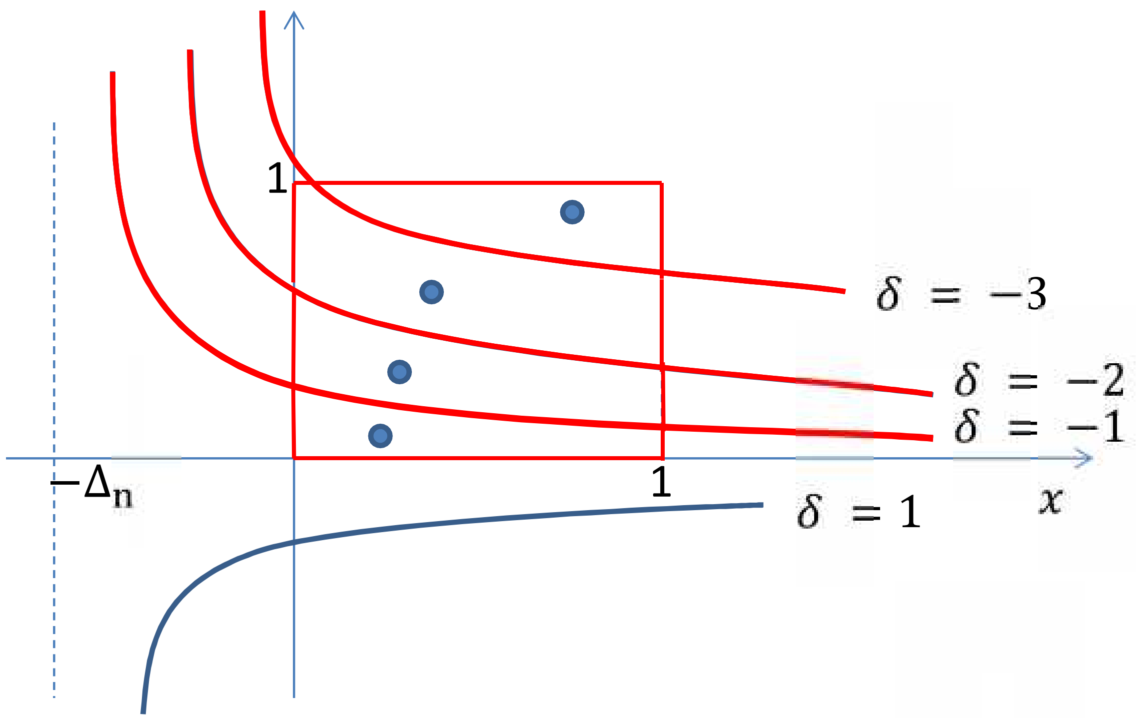
$$p_j(-x - \Delta_j) + p_n(x + \Delta_n) + \delta_{j,n} \leq 0$$

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

$$\delta_{j,n} = U_{u,A}(t_j) - U_{u,A}(t_n)$$



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

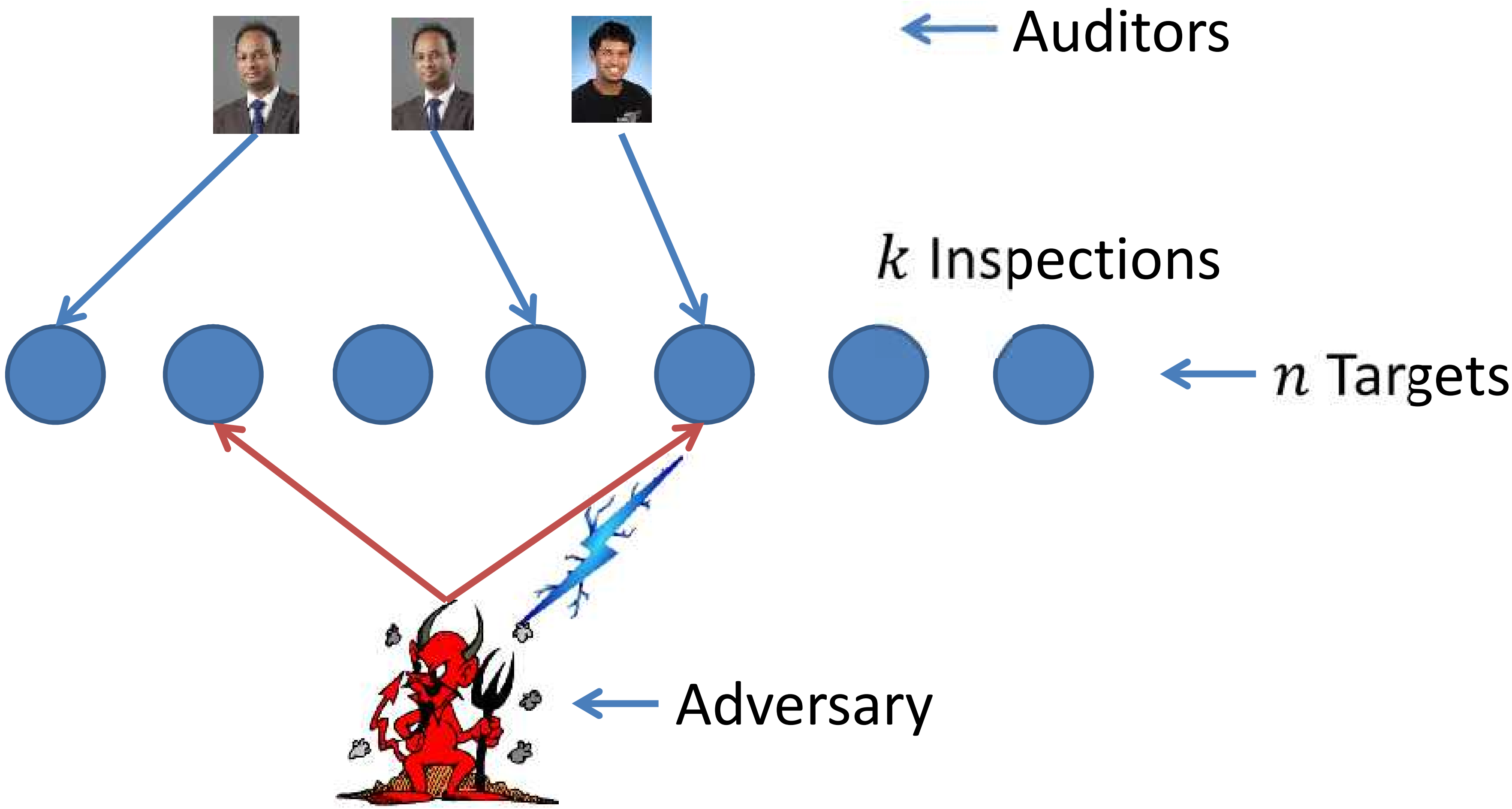
Solving Sub-problem EQ_j

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

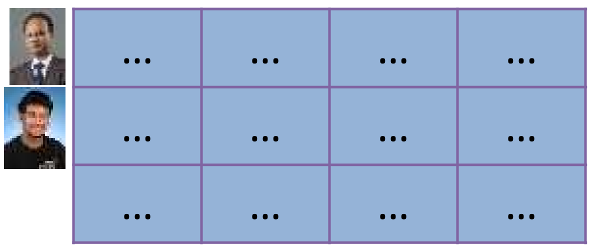
Rational Model



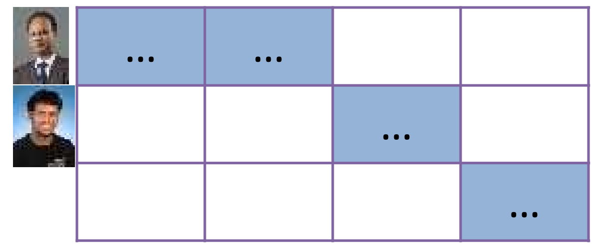
Rational Model

Captures Real Scenarios

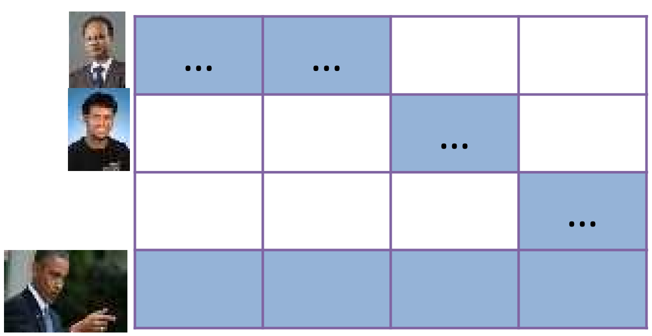
All targets auditable
by all inspections



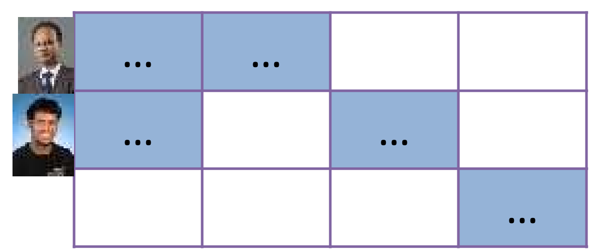
Localized auditing/
Audit by managers



Localized auditing with
central auditors



Audit by managers
with shared managers



Summary of Results

Model Features	FPT Approximation	FPTAS (under certain conditions)
Multiple defender resources	✓	✓
Subset restriction	✓	✓
Multiple (constant number) attacks	✓	?
Target-Specific punishments	✓	?

Conclusion

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?