Stackelberg Games
with Applications to Security

Chris Kiekintveld  Bo An  Albert Xin Jiang

UTEP  TEAMCORE  USC
Motivating real-world applications

Background and basic security games

Scaling to complex action spaces

Modeling payoff uncertainty: Bayesian Security Games

Human behavior and observation uncertainty

Evaluation and discussion
Motivation: Game Theory for Security

- Limited security resources: Selective checking
- Adversary monitors defenses, exploits patterns
Many Targets      Few Resources

How to assign limited resources to defend the targets?

Game Theory: Bayesian Stackelberg Games
Game Theory: Bayesian Stackelberg Games

- Security allocation: (i) Target weights; (ii) Opponent reaction
- Stackelberg: Security forces commit first
- Bayesian: Uncertain adversary types
- Optimal security allocation: Weighted random
- Strong Stackelberg Equilibrium (Bayesian)
  - \(\text{NP-hard (Conitzer/Sandholm '06)}\)

<table>
<thead>
<tr>
<th></th>
<th>Terminal #1</th>
<th>Terminal #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal #1</td>
<td>5, -3</td>
<td>-1, 1</td>
</tr>
<tr>
<td>Terminal #2</td>
<td>-5, 5</td>
<td>2, -1</td>
</tr>
</tbody>
</table>

Adversary

Police
ARMOR: Deployed at LAX 2007

“Assistant for Randomized Monitoring Over Routes”

- Problem 1: Schedule vehicle checkpoints
- Problem 2: Schedule canine patrols

Randomized schedule: (i) target weights; (ii) surveillance
ARMOR Canine: Interface

Available Canines

<table>
<thead>
<tr>
<th>Available Teams</th>
<th>Morning (AM)</th>
<th>Evening (PM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunday</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Monday</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Tuesday</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Wednesday</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Thursday</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Friday</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Saturday</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

Days to Schedule:

- July, 2009
- Sun: 28, 29, 30
- Mon: 1, 2, 3
- Wed: 15, 16
- Fri: 30

Set All: Morning 6, Evening 6

Generate Schedule
Undercover, in-flight law enforcement

Flights (each day)
~27,000 domestic flights
~2,000 international flights

Not enough air marshals:
Allocate air marshals to flights?
Federal Air Marshals Service (FAMS)

- Massive scheduling problem
- Adversary may exploit predictable schedules
- Complex constraints: tours, duty hours, off-hours

100 flights, 10 officers: 
\[1.7 \times 10^{13}\] combinations

Overall problem: 30000 flights, 3000 officers

Our focus: international sector
IRIS: “Intelligent Randomization in International Scheduling” (Deployed 2009)
US Coast Guard: *Port Resilience Operational / Tactical Enforcement to Combat Terrorism*

- Randomized patrols; deployed in Boston, with more to follow
- More realistic models of human behaviors
Application in Transition: GUARDS

• GUARDS: under evaluation for national deployment

• Transportation Security Administration
  ▶ Protect over 400 airports
    ▶ Limited security resources
    ▶ Numerous security measures
    ▶ Diverse potential threats
  ▶ Adaptive adversary
International Interest: Mumbai

- Protect networks
Urban Road Network Security

Southern Mumbai
Beyond Counterterrorism: Other Domains

- LA Sheriff’s dept (Crime suppression & ticketless travelers):

- Customs and Border Protection
- Cybersecurity
- Forest/environmental protection
- Economic leader/follower models
Research Challenges

- Scalable algorithms
- Rich representations; networks
- Payoff uncertainty, robustness
- Imperfect surveillance
- Evaluation of deployed systems
- Human behavior, bounded rationality
- Explaining game theory solutions
- …
Publications

Publications ~40 rigorously reviewed papers:

• AAMAS’ [06-12: (15)]
• AAAI [08,10-12: (10)]
• IJCAI’11: (2)
• ECAI’12: (1)
• IAAI’12: (1)
• JAIR’11
• JAAMAS’12
• AI Journal’10, 12
• Interfaces’10
• AI Magazine’09,12…
• Journal ITM’09
Motivating real-world applications

**Background and basic security games**

Scaling to complex action spaces

Modeling payoff uncertainty: Bayesian Security Games

Human behavior and observation uncertainty

Evaluation and discussion
Games

- Players:
  - \( 1, \ldots, n \)
  - *focus on 2 players*

- Strategies
  - \( a_i \in A_i \)
  - \( a = (a_1, \ldots, a_n) \in A \)

- Utility function
  - \( u_i : A \to R \)
Security Games

- Two players
  - Defender: $\Theta$
  - Attacker: $\psi$
- Set of targets: T
- Set of resources: R
  - Defender assigns resources to protect targets
  - Attacker chooses one target to attack
- Payoffs define the reward/penalty for each player for a successful or unsuccessful attack on each target
Are security games always zero-sum?

**NO!**

In real domains attackers and defenders often have different preferences and criteria

- Weighting casualties, economic consequences, symbolic value, etc.
- Player may not care about the other’s cost (e.g., cost of security, cost of carrying out an attack)

We often make a weaker assumption:

- An attack on a defended target is better than an attack on the same target if it is undefended (for the defender)
- The opposite holds for attackers (attackers prefer to attack undefended targets)
## Security Game

- **2 players**
- **2 targets**
- **1 defender resource**

<table>
<thead>
<tr>
<th></th>
<th>Target 1</th>
<th>Target 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target 1</td>
<td>1, -1</td>
<td>-2, 2</td>
</tr>
<tr>
<td>Target 2</td>
<td>-1, 1</td>
<td>2, -1</td>
</tr>
</tbody>
</table>
## Game Solutions

### Best Response

<table>
<thead>
<tr>
<th></th>
<th>Target 1</th>
<th>Target 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target 1</td>
<td>1, -1</td>
<td>-2, 2</td>
</tr>
<tr>
<td>Target 2</td>
<td>-1, 1</td>
<td>2, -1</td>
</tr>
</tbody>
</table>
## Best Response

<table>
<thead>
<tr>
<th></th>
<th>Target 1</th>
<th>Target 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target 1</td>
<td>1, -1</td>
<td>-2, 2</td>
</tr>
<tr>
<td>Target 2</td>
<td>-1, 1</td>
<td>2, -1</td>
</tr>
</tbody>
</table>
### Game Solutions

**Best Response**

<table>
<thead>
<tr>
<th></th>
<th>Target 1</th>
<th>Target 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target 1</td>
<td>1, -1</td>
<td>-2, 2</td>
</tr>
<tr>
<td>Target 2</td>
<td>-1, 1</td>
<td>2, -1</td>
</tr>
</tbody>
</table>
## Game Solutions

### Mixed Strategy

<table>
<thead>
<tr>
<th></th>
<th>Target 1</th>
<th>Target 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>50%</strong></td>
<td>1, -1</td>
<td>-2, 2</td>
</tr>
<tr>
<td><strong>50%</strong></td>
<td>-1, 1</td>
<td>2, -1</td>
</tr>
</tbody>
</table>
Nash Equilibrium

A mixed strategy for each player such that no player benefits from a unilateral deviation.

<table>
<thead>
<tr>
<th></th>
<th>Target 1</th>
<th>Target 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target 1</td>
<td>1, -1</td>
<td>-2, 2</td>
</tr>
<tr>
<td>Target 2</td>
<td>-1, 1</td>
<td>2, -1</td>
</tr>
</tbody>
</table>
Nash Equilibrium

A mixed strategy for each player such that no player benefits from a unilateral deviation.
Attackers use surveillance in planning attacks.

Defender commits to a mixed strategy.

Stackelberg Equilibrium

\{
0.1, 0.9
\} \quad \{0.5, 0.5\}

\[(0, 0) \quad (0, -0.5)\]

\((-0.9, 0.9) \quad (1.8, -0.9)\)
**Strong Stackelberg Equilibrium**

- **Strong Stackelberg Equilibrium (SSE)**
  - **Break ties in favor of the defender**
  - **Can often induce SSE by perturbing defender strategy**

- **More robust concepts**
  - **Weak Stackelberg Equilibrium not guaranenteed to exist**
  - **Payoff uncertainty**
  - **Quantal response**
  - **Equilibrium refinement**
Finding Stackelberg Equilibria

**Multi-linear programming formulation**

*Conitzer and Sandholm, 2006*

\[
\begin{align*}
\max & \sum_{s_1} p_{s_1} u_1(s_1, s_2) \\
\forall s_2', \quad & \sum_{s_1} p_{s_1} u_2(s_1, s_2') \leq \sum_{s_1} p_{s_1} u_2(s_1, s_2) \\
\sum_{s_1} p_{s_1} &= 1 \\
p_{s_1} &\geq 0
\end{align*}
\]

The formulation above gives the maximum utility of the leader when the follower chooses action \(a\) for the leader.

The Stackelberg equilibrium is obtained by maximizing over all the possible pure strategies for player two.
Single LP formulation (Korzhyk & Conitzer 2011)

\[
\begin{align*}
\max_{s_1, s_2} & \sum_{s_1, s_2} x_{s_1, s_2} u_1(s_1, s_2) \\
\forall s_2, s_2' : & \sum_{s_1} x_{s_1, s_2} u_2(s_1, s_2') \leq \sum_{s_1} x_{s_1, s_2} u_2(s_1, s_2) \\
\sum_{s_1, s_2} x_{s_1, s_2} &= 1 \\
x_{s_1, s_2} &\geq 0
\end{align*}
\]

- Relaxation of the LP for correlated equilibrium
  - removed player 1's incentive constraints
- Corollary: SSE leader expected utility at least that of best CE
Research Challenges

- Scalability
  - Large, complex strategy spaces

- Robustness
  - Payoff & observation uncertainty
  - Human decision-makers

- Not in this talk:
  - Stackelberg equilibria for dynamic games (Letchford & Conitzer 2010, Letchford et al. 2012)
  - Multiple objectives (Brown et al. 2012)
Outline

- Motivating real-world applications
- Background and basic security games
- *Scaling to complex action spaces*
- Modeling payoff uncertainty: Bayesian Security Games
- Human behavior and observation uncertainty
- Evaluation and discussion
Large Numbers of Defender Strategies

**FAMS: Joint Strategies or Combinations**

- 100 Flight tours
- 10 Air Marshals

\[1.73 \times 10^{13}\]

Schedules: ARMOR out of memory

---

**Don't enumerate ALL joint strategies**

- **Marginals** (IRIS I & II)
- **Branch and price** (IRIS III)
**IRIS I & II: Marginals Instead of Joint Strategies**

**ARMOR: 10 tours, 3 air marshals**

<table>
<thead>
<tr>
<th>ARMOR Actions</th>
<th>Tour combos</th>
<th>Prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,2,3</td>
<td>x1</td>
</tr>
<tr>
<td>2</td>
<td>1,2,4</td>
<td>x2</td>
</tr>
<tr>
<td>3</td>
<td>1,2,5</td>
<td>x3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>120</td>
<td>8,9,10</td>
<td>x120</td>
</tr>
</tbody>
</table>

**Payoff duplicates. Depends on target covered**

\[
\max_{x,q} \sum_{i \in X} \sum_{l \in L} p_i R_i x_i q^l \quad \text{subject to:}
\]

\[
\begin{align*}
1,2,3 & \quad x_i = 1, \\
1,2,4 & \quad \sum_{j \in Q} q^l_j = 1.
\end{align*}
\]

\[
0 \leq (a^l - \sum_{i \in X} C^l_{ij} x_i) \leq (1 - q^l_j) M
\]

\[x_i \in [0...1], q^l_j \in \{0,1\} \]

**MILP similar to ARMOR, y instead of x:**

- 10 instead of 120 variables
- \(y_1 + y_2 + y_3 \ldots + y_{10} = 3\)
- Sample from “y”, not enumerate “x”
- Only works for SIMPLE tours

(Korzhyk et al. 2010)
Max Defender Payoff
\[
\text{max} \quad d
\]
(5)

Attacker Strategy
Definition
\[
a_t \in \{0, 1\} \quad \forall t \in T
\]
(6)

Defender Strategy
Definition
\[
\sum_{t \in T} c_t \leq m
\]
(9)

Best Responses
\[
d - U_\Theta(t, C) \leq (1 - a_t) \cdot Z \quad \forall t \in T
\]
(10)
\[
0 \leq k - U_\Psi(t, C) \leq (1 - a_t) \cdot Z \quad \forall t \in T
\]
(11)
IRIS I

Four flights
One marshal

Zero Sum
Attacker payoffs

Uncovered | Covered
---|---
4 | 0
3 | 0
2 | 0
1 | 0

Coverage Probability

0 0 0 0 0
IRIS I

**Attack Set:**
Set of targets with maximal expected payoff for the attacker

Coverage Probability

<table>
<thead>
<tr>
<th>Flight 1</th>
<th>Flight 2</th>
<th>Flight 3</th>
<th>Flight 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Observation 1
It never benefits the defender to add coverage outside the attack set.
Compute coverage necessary to make attacker indifferent between 3 and 4

Coverage Probability

0.25  0   0   0
Observation 2
It never benefits the defender to add coverage to a subset of the attack set.

<table>
<thead>
<tr>
<th>Flight 4</th>
<th>Flight 3</th>
<th>Flight 2</th>
<th>Flight 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
IRIS I

Coverage Probability

<table>
<thead>
<tr>
<th>Flight 1</th>
<th>Flight 2</th>
<th>Flight 3</th>
<th>Flight 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0.33</td>
<td>0.5</td>
</tr>
</tbody>
</table>
IRIS I

Need more than one air marshal!

Coverage Probability

<table>
<thead>
<tr>
<th>Flight 1</th>
<th>Flight 2</th>
<th>Flight 3</th>
<th>Flight 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75</td>
<td>0.66</td>
<td>0.5</td>
<td>0</td>
</tr>
</tbody>
</table>
IRIS I

Can still assign 0.17

Coverage Probability

<table>
<thead>
<tr>
<th>Flight 1</th>
<th>Flight 2</th>
<th>Flight 3</th>
<th>Flight 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.33</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Allocate all remaining coverage to flights in the attack set.

Fixed ratio necessary for indifference.

Coverage Probability:

<table>
<thead>
<tr>
<th>Flight</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flight 4</td>
<td>0.54</td>
</tr>
<tr>
<td>Flight 3</td>
<td>0.38</td>
</tr>
<tr>
<td>Flight 2</td>
<td>0.08</td>
</tr>
<tr>
<td>Flight 1</td>
<td>0</td>
</tr>
</tbody>
</table>
IRIS Speedups

**Scaling with Targets: Compact**

- **ARMOR**
- **IRIS I**
- **IRIS II**

<table>
<thead>
<tr>
<th>Targets</th>
<th>Runtimes (min)</th>
<th>ARMOR Actions</th>
<th>ARMOR Runtime</th>
<th>IRIS Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0</td>
<td>6,048</td>
<td>4.74s</td>
<td>0.09s</td>
</tr>
<tr>
<td>11</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>3.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>4.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ARMOR Actions**
- **FAMS Ireland**: 6,048
- **FAMS London**: 85,275

**ARMOR Runtime**
- **FAMS Ireland**: 4.74s
- **FAMS London**: ----

**IRIS Runtime**
- **FAMS Ireland**: 0.09s
- **FAMS London**: 1.57s
IRIS III: Branch and Price: Tours of Arbitrary Size

Branch & Price: Branch & Bound + Column Generation

- Not out of the box
- Upper bounds: IRIS I
- Column generation: Network flow

Lower bound 1: Adversary best response → Target1
Lower bound 2: Adversary best response → Target2
Lower bound N: Adversary best response → TargetN

Upper bound: Adversary → 2...N
LEAF NODE: Incrementally build support for mixed strategy

“Master” Problem (mixed integer program)

Solution supported by N pure strategies

(N+1)\textsuperscript{th} pure Strategy

Return the “best” joint schedule:
Minimum reduced cost

Minimum cost network flow

Lower bound 1: Adversary \rightarrow Target 1

Lower bound 2: Adversary \rightarrow Target 2

Lower bound N: Adversary \rightarrow Target N

Capacity 1 on all links
IRIS Results

Comparison (200 Targets, 10 Resources)

- IRIS II
- B&P
- IRIS III

Runtime (in secs) [log-scale]

Number of Schedules

ARMOR
Runs out of memory

Scale-up (200 Targets, 1000 schedules)

Runtime (in seconds)

Number of Resources
Los Angeles Metro Rail System

- Barrier-free system with random inspections
- Approximately 300,000 daily riders, ≈6% fare evaders
- Fare evasion costs ≈ $5.6 million annually (Booz Allen Hamilton 2007)
What is a pure strategy of the defender?

How to Model?
What is a pure strategy of the defender?

How to Model?

Check fares at “Mission” station
How to Model?

Check fares at “Mission”
Go to “Southwest Museum”
Check fares at “Southwest Museum”
What is a pure strategy of the defender?

How to Model?

Check fares at “Mission” from 7am to 7:50am
Go to “Southwest Museum” at 7:50am
Check fares at “Southwest Museum” from 8am to 9am
What is a pure strategy of the defender?

How to Model?

Check fares at “Mission” from 7am to 7:50am
Go to “Southwest Museum” at 7:50am
Check fares at “Southwest Museum” from 8am to 9am

How many such pure strategies?
Problem Setting

- Transition graph

Vertex: station and time pair
Problem Setting

*Transition graph*

**Edge: inspection action**
Problem Setting

- Transition graph

Edge: inspection action
Problem Setting

- Transition graph

Edge: *inspection action*

- $l_e$ - action duration
- $f_e$ - fare-check effectiveness
Problem Setting

- Transition graph

Patrols: bounded-length paths
Problem Setting

- Transition graph

Patrols: *bounded-length paths*

\[ \gamma \rightarrow \text{patrol units} \]

\[ \kappa \rightarrow \text{patrol hours per unit} \]
Problem Setting cont.

- Riders: *multiple types*
  - *Each type takes fixed route*
  - *Fully observes the probability of being inspected*
  - *Binary decision: buy or not buy the ticket*
  - *Perfectly rational and risk-neutral*

<table>
<thead>
<tr>
<th></th>
<th>6PM</th>
<th>7PM</th>
<th>8PM</th>
<th>9PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A, 6PM</td>
<td>A, 7PM</td>
<td>A, 8PM</td>
<td>A, 9PM</td>
</tr>
<tr>
<td>B</td>
<td>B, 6PM</td>
<td>B, 7PM</td>
<td>B, 8PM</td>
<td>B, 9PM</td>
</tr>
<tr>
<td>C</td>
<td>C, 6PM</td>
<td>C, 7PM</td>
<td>C, 8PM</td>
<td>C, 9PM</td>
</tr>
</tbody>
</table>
Problem Setting cont.

- Riders: *multiple types*
  - *Each type takes fixed route*
  - *Fully observes the probability of being inspected*
  - *Binary decision: buy or not buy the ticket*
  - *Perfectly rational and risk-neutral*

Why do we need this edge?
Based on transition graph

Strategy representation: *marginal coverage on edges*
Based on *transition graph*

Strategy representation: *marginal coverage on edges*
**Basic Compact Formulation**

- **Transition graph:** \( G = \langle V, E \rangle \)
  - *Dummy source* \( v^+ \), possible starting vertices \( V^+ \)
  - *Dummy sink* \( v^- \), possible ending vertices \( V^- \)

\[
\begin{align*}
\max_{x, u} & \sum_{\lambda \in \Lambda} p_{\lambda} u_{\lambda} \\
\text{s.t.} & \quad u_{\lambda} \leq \min \{ \rho, \tau \sum_{e \in \lambda} x_{e} f_{e} \}, \text{ for all } \lambda \in \Lambda \\
& \sum_{v \in V^+} x_{(v^+, v)} = \sum_{v \in V^-} x_{(v, v^-)} \leq \gamma \\
& \sum_{(v', v) \in E} x_{(v', v)} = \sum_{(v, v^+) \in E} x_{(v, v^+)}, \text{ for all } v \in V \\
& \sum_{e \in E} l_{e} \cdot x_{e} \leq \gamma \cdot \kappa, 0 \leq x_{e} \leq \alpha, \forall e \in E
\end{align*}
\]
Issues with Basic Compact Formulation

- Patrol length may not be bounded by $\kappa$

  \[ E.g., \gamma = 1, \kappa = 1 \]

\[
\sum_{v \in V^+} x_{(v^+, v)} = \sum_{v \in V^-} x_{(v, v^-)} \leq \gamma \quad (4)
\]
\[
\sum_{(v', v) \in E} x_{(v', v)} = \sum_{(v, v^+) \in E} x_{(v, v^+)} \text{, for all } v \in V \quad (5)
\]
\[
\sum_{e \in E} l_e \cdot x_e \leq \gamma \cdot \kappa, 0 \leq x_e \leq \alpha, \forall e \in E \quad (6)
\]
Issues with Basic Compact Formulation

- Patrol length may not be bounded by $\kappa$
  - $E.g., \gamma = 1, \kappa = 1$

0.5, $v^+ \rightarrow v_3 \rightarrow v^-$

0.5, $v^+ \rightarrow v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v^-$
Extended Compact Formulation

- History-duplicate transition graph
  - Store history information in vertices
  - Access necessary patrol information without exponential blowup
Extended Compact Formulation cont.

- History-duplicate transition graph
  - Store history information in vertices
  - Access necessary patrol information without exponential blowup
- E.g., to forbid patrols longer than 2 hours
  - What information should be duplicated?
Extended Compact Formulation cont.

- **History-duplicate transition graph**
  - Store history information in vertices
  - Access necessary patrol information without exponential blowup
- E.g., to forbid patrols longer than 2 hours
  - 2 subgraphs corresponding to 2 starting time: 6pm and 7pm

```
A, 6PM (6PM) ----> A, 7PM (6PM) ----> A, 8PM (6PM)
|                  /                     |                  /                     |
B, 6PM (6PM) ----> B, 7PM (6PM) ----> B, 8PM (6PM)
|                  |                     |                  |                     |
C, 6PM (6PM) ----> C, 7PM (6PM) ----> C, 8PM (6PM)
```

```
A, 7PM (7PM) ----> A, 8PM (7PM) ----> A, 9PM (7PM)
|                  /                     |                  /                     |
B, 7PM (7PM) ----> B, 8PM (7PM) ----> B, 9PM (7PM)
|                  |                     |                  |                     |
C, 7PM (7PM) ----> C, 8PM (7PM) ----> C, 9PM (7PM)
```
Outline

- Motivating real-world applications
- Background and basic security games
- Scaling to complex action spaces
- **Modeling payoff uncertainty: Bayesian Security Games**
- Human behavior and observation uncertainty
- Evaluation and discussion
# Robustness

<table>
<thead>
<tr>
<th></th>
<th>Target 1</th>
<th>Target 2</th>
<th>Target 3</th>
<th>Target 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Defender</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reward</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>3</td>
</tr>
<tr>
<td>Penalty</td>
<td>-1</td>
<td>-4</td>
<td>-6</td>
<td>-10</td>
</tr>
<tr>
<td><strong>Attacker</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Penalty</td>
<td>-2</td>
<td>-3</td>
<td>-3</td>
<td>-5</td>
</tr>
<tr>
<td>Reward</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>9</td>
</tr>
</tbody>
</table>

**How do we know the model is correct?**

**If it is not exactly correct, how robust is the solution?**
What is the attacker’s value for a successful attack on a particular target?

- What is the likely number of casualties?
- What is the economic cost?
- What is the value of the media exposure?
- What is the symbolic value of the attack?
- How should these factors be weighted?

Answers can only be estimated
Modeling Choices

Players
- How many?
- Model organizations as individuals?
- Specific people or generic types of people?
- Are players rational?
- If not, how do they behave?

Actions
- What is the set of feasible actions?
- Do players know all of the actions?
- If the set is infinite, how do we represent it?
- Are some actions similar to others?
- Are actions sequential?

Payoffs
- How do we determine payoffs?
- Are payoffs known to all players?
- What is the uncertainty about the payoffs?
- Are payoffs deterministic or stochastic?
- Do players care about risk?

Solution concepts
- What to do if there are multiple equilibria?
- Do we care about the worst case?
- Bounded rationality
- Limited observability
- Can the solution be computed?
Robustness Perspectives

- Game theorist’s perspective
  - *The model is given, and known to everyone*
  - *We can model uncertainty explicitly by making the model more complex*

- Engineer’s perspective:
  - *Do the math*
  - *Add a “fudge factor” to for safety*
  - *The cost is worth the risk reduction*
  - *“Unknown unknowns”*
  - *Confidence is critical*

*Real problems force us to deal with robustness*
Research on Robustness

- Payoff uncertainty

- Human behavior

- Observation/Execution uncertainty
  - Yin et al 2010, Pita et al 2011, Yin et al 2011, An et al 2012, ...
Diverse Techniques

Bayesian Models

Finite Models
Infinite Models

Interval Models

Modified Strategy Models
Finite Bayesian Games

P = 0.3

<table>
<thead>
<tr>
<th>Term #1</th>
<th>Term #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term #1</td>
<td>5, -3</td>
</tr>
<tr>
<td>Term #2</td>
<td>-5, 5</td>
</tr>
</tbody>
</table>

P = 0.5

<table>
<thead>
<tr>
<th>Term #1</th>
<th>Term #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term #1</td>
<td>2, -1</td>
</tr>
<tr>
<td>Term #2</td>
<td>-1, 1</td>
</tr>
</tbody>
</table>

P = 0.2

<table>
<thead>
<tr>
<th>Term #1</th>
<th>Term #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term #1</td>
<td>4, -2</td>
</tr>
<tr>
<td>Term #2</td>
<td>-4, 3</td>
</tr>
</tbody>
</table>

Harsanyi Transformation

NP-Hard
Multiple LPs Method

[Conitzer and Sandholm 2006]

- First optimization formulation for FBSG
- Basic idea:
  - Enumerate attacker pure strategies
  - Solve an LP to maximize leader’s payoff

\[
\begin{align*}
\max_{a \in A_2} & \quad \max_{\sigma_1} \sum\limits_{a' \in A_1} p_1(a') u_1(a', a) \\
\text{s.t.} & \quad \sum\limits_{a' \in A_1} p_1(a') u_2(a', a) \geq \sum\limits_{a' \in A_1} p_1(a') u_2(a', a'') \quad \forall a'' \in A_1 \\
& \quad \sum\limits_{a \in A_1} p_1(a') = 1 \\
& \quad p_1(a) \geq 0 \quad \forall a \in A_1
\end{align*}
\]
Challenge: Exponential number of type combinations
Handling Multiple Adversary Types: ARMOR

<table>
<thead>
<tr>
<th>Term #1</th>
<th>Term #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term #1</td>
<td>5, -3</td>
</tr>
<tr>
<td>Term #2</td>
<td>-5, 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Term #1</th>
<th>Term #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term #1</td>
<td>2, -1</td>
</tr>
<tr>
<td>Term #2</td>
<td>-1, 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Term #1</th>
<th>Term #2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term #1</td>
<td>4, -2</td>
</tr>
<tr>
<td>Term #2</td>
<td>-4, 3</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\max_{x, q} & \quad \sum_{i \in X} \sum_{l \in L} \sum_{j \in Q} p^l R_{ij} x_i q_j^l \\
\text{s.t.} & \quad \sum_{i} x_i = 1, \sum_{j \in Q} q_j^l = 1 \\
& \quad 0 \leq (a^l - \sum_{i \in X} C_{ij} x_i) \leq (1 - q_j^l) M \\
& \quad x_i \in [0...1], q_j^l \in \{0,1\}
\end{align*}
\]
ARMOR: Run-time Results

- Multiple LPs (Conitzer & Sandholm’06)
- MIP-Nash (Sandholm et al’05)
- Sufficient for LAX
Scaling Up: Hierarchical Solver (HBGS)

[Jain et al. 2011]

- Efficient tree search
  - Bounds and pruning
  - Branching heuristics
- Evaluate fewer LPs
- Column generation
  - Consider restricted games
  - Solve much smaller LPs
Scaling Up: Hierarchical Solver (HBGS)

- Key Idea: solve restricted games (few types)
- Use solutions to generate bounds/heuristics

Each node in this tree represents a full Bayesian Stackelberg game

Can use column generation to solve these nodes
Theorem 1: If a pure strategy is infeasible in a “restricted” game, all its combinations are infeasible in the Bayesian game.
**Theorem 2**: Leader payoff in the Bayesian game is upper bounded by the sum of leader payoffs in the corresponding restricted games.

\[ \mathcal{V}(< t_1, t_2, t_3, t_4 >) \leq \sum_{\lambda \in \Lambda} p_\lambda B_\lambda(t_\lambda) \]
Column Generation

Master Problem

Slave Problem

Defender and Attacker Optimization Constraints

Scheduling Constraints
### HBGS Results

<table>
<thead>
<tr>
<th>Types</th>
<th>Follower Pure Strategy Combinations</th>
<th>Runtime (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>9.7e7</td>
<td>0.41</td>
</tr>
<tr>
<td>20</td>
<td>9.5e13</td>
<td>16.33</td>
</tr>
<tr>
<td>30</td>
<td>9.3e20</td>
<td>239.97</td>
</tr>
<tr>
<td>40</td>
<td>9.1e27</td>
<td>577.49</td>
</tr>
<tr>
<td>50</td>
<td>8.9e34</td>
<td>3321.68</td>
</tr>
</tbody>
</table>
Approximation
HUNTER

[Yin et al. 2012]

- Improves on tree search from HBGS
- Improved bounds (convex hulls on types)
- Bender’s decomposition on LPs

(a) Scaling up types.
(b) Scaling up pure strategies.
Finite vs Infinite BSG

- Finite games capture distinct attacker types
  - *Terrorists vs. local criminal activity*
  - *Attackers with different motivations*

- Infinite games capture distributional uncertainty
  - *E.g., Gaussian, Uniform distributions*
  - *Natural for expressing beliefs over possible values*
  - *Useful in knowledge acquisition from experts*
Distributional Payoff Representation

[Kiekintveld et al. 2011]
given a coverage vector $C$...

given a coverage vector $C$...
Problem 1 of 2

given a coverage vector $C$...

...and payoff distributions

<table>
<thead>
<tr>
<th>Target</th>
<th>Coverage</th>
<th>Payoff Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target 1</td>
<td>20%</td>
<td>Payoff 0</td>
</tr>
<tr>
<td>Target 2</td>
<td>80%</td>
<td>Payoff 0</td>
</tr>
<tr>
<td>Target T</td>
<td>50%</td>
<td>Payoff 0</td>
</tr>
</tbody>
</table>

$attack\ vector\ A(C)$
Problem 2 of 2

find the optimal coverage vector $C^*$.

... given $A(C)$ for every $C$

$a_1(C)$

$a_2(C)$

$a_3(C)$

... $a_T(C)$
**Approach**

**Coverage Vector**

- Attack Vector

- (1) Monte-Carlo estimation
- (2) Numerical methods
- (1) Optimal Finite Algorithms
- (2) Sampled Replicator Dynamics
- (3) Greedy Monte-Carlo
- (4) Decoupled Target Sets
Attacker Response Estimation

Attacker Response (Gaussian)

- RMSE vs Runtime (ms)
  - PWC Approximation
  - Monte Carlo

Graph showing the comparison between PWC Approximation and Monte Carlo methods for estimating attacker response over runtime.
Computing Coverage Vectors

- **Baselines**
  - *Mean (ignore uncertainty)*
  - *Uniform Random*

- **Exact optimization given sampled types**
  - *SBE (ARMOR variation)*

- **Worst-case optimization**
  - *BRASS*

- **Approximate optimization**
  - *Replicator Dynamics (SRD)*
  - *Greedy Monte Carlo (GMC)*
  - *Decoupled Target Sets (DTS)*
Assuming perfect information is very brittle

Approximate both type distribution and optimization
Beyond Bayesian Games

- Bayesian games are powerful
  - General framework for model uncertainty
  - Exact behavior predictions based on uncertainty

- Some limitations
  - Require distributional information
    - Even MORE parameters to specify!
    - What if these are wrong?
  - Computational challenges (NP-hard)
  - Uncertainty about human decision making is hard to capture in Bayesian models
# Interval Security Games

[Kiekintveld et al. 2012]

<table>
<thead>
<tr>
<th></th>
<th>Target 1</th>
<th>Target 2</th>
<th>Target 3</th>
<th>Target 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defender</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reward</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Penalty</td>
<td>-1</td>
<td>-4</td>
<td>-6</td>
<td>-10</td>
</tr>
<tr>
<td>Attacker</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Penalty</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Reward</td>
<td>[1,3]</td>
<td>[2,5]</td>
<td>[4,7]</td>
<td>[6,10]</td>
</tr>
</tbody>
</table>

- Attacker payoffs represented by intervals
- Maximize worst case for defender
- Distribution-free
• Fast feasibility checks
  – Given resource constraint, can the defender guarantee a given payoff?
  – Exploits structure of security games

• Binary search on defender payoffs

• Polynomial time: $O(n^2 \times \log(1/\varepsilon))$
Attacker Payoffs

5 Targets
Bars represent range of possible attacker payoffs

Defender Coverage

| 0 | 0 | 0 | 0 | 0 | 0 |
When targets are covered, payoffs decrease and range shrinks.
Given a coverage strategy, which set of targets *could* be attacked?

Minimum attacker payoff is $R$

Any target with a possible value greater than $R$ is in the *potential attack set*. 

<table>
<thead>
<tr>
<th>Defender Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
</tbody>
</table>
Main Idea:

- Design fast feasibility check to determine if a given defender payoff is possible
- Use binary search on defender payoffs
- Necessary resources increases monotonically with defender payoff
Determine whether we can guarantee a defender payoff of $D^*$ using $m$ or fewer resources.

Challenge: potential attack set depends on coverage, and number of possible sets is combinatorial.
For any potential attack set, there is some target $t'$ that determines the value of $R$

We will guess which target is $t'$ and *construct* a minimal solution for this guess (n choices)

As soon as we find a choice of $t'$ that works, we have a feasible solution
Consider the selection  
\[ t' = t_2 \]

Since \( t' \) is in the PAS, it must give \( D^* \) if attacked.

Calculate minimal coverage on \( t' \) using:

\[
c_i^1 = \max(0, 1 - \frac{D^*}{U^u_{\Theta}(t_i)})
\]
Consider the selection \( t' = t_2 \)

Since \( t' \) is in the PAS, it must give \( D^* \) if attacked

Calculate minimal coverage on \( t' \) using:

\[
c_i^1 = \max(0, 1 - \frac{D^*}{U_\Theta^u(t_i)})
\]
For every other target \( t \), consider two cases:

1) Target is in the PAS
2) Target is not in the PAS
For every other target $t''$, consider two cases:

1) Target is in the PAS
2) Target is not in the PAS

**Case 1**

Payoff for $t''$ must be at least $D^*$

\[ c_i^1 = \max(0, 1 - \frac{D^*}{U^u_{\Theta}(t_i)}) \]
For every other target $t''$, consider two cases:

1) Target is in the PAS
2) Target is not in the PAS

**Case 2**

Max payoff to attacker for $t''$ must be $< R$

$$c_i^2 = \max(0, 1 - \frac{R}{U_{\max}(t_i)})$$
Constructing a Solution

Final consistency check

No target other than $t'$ can have a higher minimum attacker payoff

Otherwise, $t'$ does not set $R$ contradicting the initial assumption

$$c^3_i = \max(0, 1 - \frac{R}{U_{\Psi}^u_{\min}(t_i)})$$

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0.3</th>
<th>0.4</th>
<th>0</th>
</tr>
</thead>
</table>

Defender Coverage
For each target, compute three coverage values:

- $c^1$: coverage for $D^*$
- $c^2$: coverage not in PAS
- $c^3$: consistency with $R$

Best value given by:

$$\max(c^3_i, \min(c^1_i, c^2_i))$$
Defender Coverage

Need to check each target as \( t' \)

\( O(n^2) \) worst case to test feasibility for \( D^* \)

Binary search on \( D \)

\( O(n^2 \log(1/\varepsilon)) \)

where \( \varepsilon \) is error term
Interval Solver Scalability

Fastest Bayesian solvers (HBGS, HUNTER) scale only to 10s or 100s of targets
Motivating real-world applications
Background and basic security games
Scaling to complex action spaces
Modeling payoff uncertainty: Bayesian Security Games
**Human behavior and observation uncertainty**
Evaluation and discussion
Key Topics

PART I: Integrate models of human decision making as attacker’s response

Key model used:
- Anchoring bias and epsilon-bounded rationality
- Prospect Theory [Kahneman and Tvesky, 1979]
- Quantal Response [McKelvey and Palfrey, 1995]

New efficient algorithms

Results from experiments with human subjects
- Quantal Response (QRE) outperforms other algorithms

PART II: Impact of limited observations assuming rational attacker
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td><img src="image2.png" alt="Image" /></td>
<td><img src="image3.png" alt="Image" /></td>
<td><img src="image4.png" alt="Image" /></td>
<td><img src="image5.png" alt="Image" /></td>
<td><img src="image6.png" alt="Image" /></td>
<td><img src="image7.png" alt="Image" /></td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
</tbody>
</table>

**Your Rewards:**

<table>
<thead>
<tr>
<th>8</th>
<th>5</th>
<th>3</th>
</tr>
</thead>
</table>

**Your Penalties:**

<table>
<thead>
<tr>
<th>−3</th>
<th>−2</th>
<th>−3</th>
</tr>
</thead>
</table>

**Pirate’s Rewards:**

<table>
<thead>
<tr>
<th>4</th>
<th>3</th>
<th>1</th>
</tr>
</thead>
</table>

**Pirate’s Penalties:**

<table>
<thead>
<tr>
<th>−8</th>
<th>−10</th>
<th>−1</th>
</tr>
</thead>
</table>
Uncertainty: Human Bounded Rationality and Observations

Average expected reward

- Unobserved
- 5 Observations
- 20 Observations
- Unlimited

178 total subjects, 2480 trials, 40 subjects for each setting

Four reward structures, four observation conditions

DOBSS: Outperforms uniform random, similar to Maximin
Uncertainty: Human Bounded Rationality and Observations

**COBRA:**
- “epsilon optimality”
- Anchoring bias: Full observation vs no observation: $\alpha$

$$\max_{x,q} \sum_{i \in X} \sum_{l \in L} \sum_{j \in Q} p^l R^l_{ij} x_i q^l_j$$

s.t. $x' = (1 - \alpha) x + \alpha (1 | X |)$

$\varepsilon (1 - q^l_j) \leq (a^l - \sum_{i \in X} C^l_{ij} x_i') \leq \varepsilon + (1 - q^l_j) M$

Choosing:
- No observation: $\alpha = 1$
- Full observation: $\alpha = 0$
Unlimited Observations: Choosing $\alpha$

- Average Reward
- DOBSS
- MAXIMIN
- UNIFORM
- COBRA(0,2.5)
- COBRA(0.20,2.5)
- COBRA(0.37,2.5)
- COBRA(0.60,2.5)
- COBRA(0.70,2.5)

- Entropy
- $\alpha$ - Setting
Prospect Theory

- Model human decision making under uncertainty
- Maximize the ‘prospect’ [Kahneman and Tversky, 1979]

\[
\text{prospect} = \sum_{i \in \text{AllOutcomes}} \pi(x_i) \cdot V(C_i)
\]

- \( \pi(\cdot) \): weighting function
- \( V(\cdot) \): value function
Empirical weighting function

- Slope gets steeper as $x$ gets closer to 0 and 1
- Not consistent with probability definition
  - $\pi(x) + \pi(1-x) < 1$
- Empirical value:
  - $\gamma = 0.64$ ($0 < \gamma < 1$)
Compute Defender Strategy

- Piecewise Linear Approximation

![Graph showing piecewise linear approximation with function Π(x)]
Empirical value function

- Risk averse regarding gain
- Risk seeking regarding loss
- Empirical value:
  \[ \alpha = \beta = 0.88, \lambda = 2.25 \]
BRPT: Best Response to PT

- Mixed-Integer Linear Program
- Goal: maximize defender expected utility

\[
\begin{align*}
\max_{x} & \quad \text{DefenderUtility} \\
\text{s.t.} & \quad \sum_{i \in X} x_i \leq \text{Total\_Resources} \quad (1) \\
\pi(x_i) &= \sum_{k=1..5} b_k \cdot x_{ik} \quad (2) \\
\sum_{j \in Q} q_j &= 1 \quad (3) \\
0 \leq \text{Adversary Prospect} &- \sum_{i \in X} \pi(x_i) \cdot V(C_{ij}) \leq M \cdot (1 - q_j), \ \forall j \in Q \quad (4) \\
\text{DefenderUtility} - \sum_{i \in X} x_i \cdot R_{ij} &\leq M \cdot (1 - q_j) \quad (5)
\end{align*}
\]
Quantal Response Equilibrium

- Error in individual’s response
  - *Still: more likely to select better choices than worse choices*
- Probability distribution of different responses
- Quantal best response:
  \[ q_j = \frac{e^{\lambda U(j, x)}}{\sum_{k=1}^{M} e^{\lambda U(k, x)}} \]
  - \( \lambda \): represents error level (=0 means uniform random)
  - *Maximal likelihood estimation* (\( \lambda = 0.76 \))
Solve the Nonlinear optimization problem

\[
\max_x \frac{\sum_{j \in Q} \sum_{i \in X} x_i R_{ij} \cdot \prod_{l \in X} e^{\lambda C_{ij} x_l}}{\sum_{k \in Q} \prod_{l \in X} e^{\lambda C_{lk} x_l}}
\]

s.t. \[\sum_{i \in X} x_i \leq \text{Total Resource}\]
\[0 \leq x_i \leq 1, \quad \forall i \in X\]
Subjects are given $8 as the starting budget

For each point they gain, $0.1 real money is paid
Experiment Setting

- 7 payoff structures
  - 4 new, 3 from previous tests with COBRA

- 5 strategies for each payoff structure
  - New methods: BRPT, RPT and BRQR
  - Leading contender: COBRA
  - Perfect rational baseline: DOBSS

- Subjects play all games (randomized orders)
- No feedback until subject finishes all games
Average Defender Expected Utility

[Graph showing expected utility values for different payoffs and strategies: BRPT, RPT, BRQR, COBRA, DOBSS.]

- Payoff 1
- Payoff 2: BRPT = -2.5, RPT = -1.5, BRQR = -2, COBRA = -1, DOBSS = 0
- Payoff 3: BRPT = -2.5, RPT = -1.5, BRQR = -2, COBRA = -1, DOBSS = 0
- Payoff 4: BRPT = -2.5, RPT = -1.5, BRQR = -2, COBRA = -1, DOBSS = 0

Y-axis: Utility values ranging from -3 to 2.
**Result Summary**

- **BRQR** outperforms **DOBSS** in all 7 payoffs
  - *In payoff 1, 3 and 4, the result is statistically significant*

- **BRQR** outperforms **COBRA** in all 7 payoffs
  - *In payoff 2, 3 and 4, the result is statistically significant*

- The poor performance **BRPT** is surprising!
Uncertainty in Adversary Decision: MATCH

**Builds on QR, exploiting security game structure:**
- Like QR: Adversary response error; better choice more likely
- Bound loss to defender on adversary deviation

Results on 100 games

<table>
<thead>
<tr>
<th></th>
<th>MATCH wins</th>
<th>Draw</th>
<th>QR wins</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\alpha = 0.05)</td>
<td>42</td>
<td>52</td>
<td>6</td>
</tr>
</tbody>
</table>

![Graph showing comparison between BRQR and MATCH](image)
Uncertainty in Attacker Surveillance: Stackelberg vs Nash

Defender commits first:
- Attacker conducts surveillance
- Stackelberg (SSE)

Simultaneous move game:
- Attacker conducts no surveillance
- Mixed strategy Nash (NE)

How should a defender compute her strategy?

For security games (*):

Set of defender strategies

NE = Minimax

SSE
**RECON:**

- **Worst-case protection against action-execution & observation uncertainty**
- **Efficient MILP and heuristics**

![Graph showing solution quality against #Targets for RECON, MAXIMIN, ERASER Worst, and COBRA Worst.](graph.png)
Motivating real-world applications
Background and basic security games
Scaling to complex action spaces
Modeling payoff uncertainty: Bayesian Security Games
Human behavior and observation uncertainty
Evaluation and discussion
How Do We Evaluate Deployed Systems?

- “Main” vs “Application track”: Evaluating deployed systems not easy
  - Cannot switch security on/off for controlled experiments
  - Cannot show we are “safe” (no 100% security)

- Are our systems useful: Are we better off than previous approaches?
  1. Models and simulations
  2. Human adversaries in the lab
  3. Actual security schedules before vs after
  4. Expert evaluation
  5. “Adversary” teams simulate attack
  6. Supportive data from deployment
  7. Future deployments
Key Conclusions

- Human schedulers:
  - Predictable patterns, e.g. LAX, FAMS (GAO-09-903T)
  - Scheduling burden

- Uniform random:
  - Non-weighted, e.g. officers to sparsely crowded terminals

- Simple weighted random:
  - No adversary reactions, & enumerate large number of combinations?

Systems in use for a number of years: without us “forcing” use
  - Internal evaluations, e.g. LAX evaluation by FBI, foreign experts
1. Models and Simulations: Example from IRIS (FAMS)
3. Actual Security Schedules Before vs After: Example from PROTECT (Coast Guard)

Patrols Before PROTECT: Boston

Patrols After PROTECT: Boston
4. Expert Evaluation

Example from ARMOR, IRIS & PROTECT

February 2009: Commendations
LAX Police (City of Los Angeles)

July 2011: Operational Excellence Award (US Coast Guard, Boston)

September 2011: Certificate of Appreciation (US Federal Air Marshals Service)
5. “Red” Teaming, Supportive data
   Example from PROTECT

- “Mock attacker” team deployed in Boston
  - Incorporated adversary’s known intent, capability
  - Comparing PRE- to POST-PROTECT: “deterrence” improved

- Additional real-world indicators from Boston:
  - PRE- to POST-PROTECT: Actual reports of illicit activity
  - Industry port partners comments:
    - “The Coast Guard seems to be everywhere, all the time."
      (With no actual increase in the number of resources)
6. What Happened at Checkpoints before and after ARMOR -- Not a Controlled Experiment!

- January 3rd: Loaded 9/mm pistol
- January 9th: 16-handguns, 4-rifles, 1-assault rifle; 1000 rounds of ammo
- January 10th: Two unloaded shotguns
- January 12th: Loaded 22/cal rifle
- January 17th: Loaded 9/mm pistol
- January 22nd: Unloaded 9/mm pistol
Deployed Applications:
ARMOR, IRIS, PROTECT, GUARDS

Research challenges

- **Efficient algorithms**: Scale-up to real-world problems
- **Observability**: Adversary surveillance uncertainty
- **Human adversary**: Bounded rationality, observation power
- **Uncertainty**...
Thank you!

Chris Kiekintveld  Bo An  Albert Xin Jiang

cdkiekintveld@utep.edu
boa@usc.edu
jiangx@usc.edu
http://teamcore.usc.edu/security