Computing Optimal Randomized Resource Allocations for Massive Security Games

C. Kiekintveld et al., AAMAS 2009 (USC)
Presented by Neal Gupta
September 2016
Overview

- Motivation and Introduction
- Compact Representations of Security Games
- ERASER
- Alternative LP formulation (ORIGAMI)
- ERASER-C (more realistic)
- Results
Overview

- Motivation and Introduction
- Compact Representations of Security Games
- ERASER
- Alternative LP formulation (ORIGAMI)
- ERASER-C (more realistic)
- Results
Motivation and introduction

- \( m \) resources to cover \( n \) targets, \( m < n \)
- Defender can commit to a mixed strategy
- Attacker can observe the probabilities for each coverage set
  - Surveillance
  - Insider threat
- Attacker chooses a pure strategy
- Equilibrium concept not ex post
Motivation and introduction

- Initially assume interchangeable resources (to extend later)
- Assume players are risk neutral
- One type of follower (attacker)
  - Recall that this should be in P
  - But, the number of pure strategies now is unreasonably large, with 100 targets and 10 resources ~17 trillion!
Motivation and introduction

- Running example: 4 targets, 2 resources
- Qualitatively
  - Defender values all 4 targets equally (and prefers a covered attack to an uncovered attack).
  - Attacker gets twice as much utility for successful attack on target 3. All failed attacks get the same (lower) utility.
Motivation and introduction

- Running example
- Quantitatively

<table>
<thead>
<tr>
<th>Targets 1, 2, 4</th>
<th>Cov’d</th>
<th>Uncov’d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defender</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Attacker</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target 3</th>
<th>Cov’d</th>
<th>Uncov’d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defender</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Attacker</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>
Motivation and introduction

- Running example
- Quantitatively

<table>
<thead>
<tr>
<th>Targets 1, 2, 4</th>
<th>Cov’d</th>
<th>Uncov’d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defender</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Attacker</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target 3</th>
<th>Cov’d</th>
<th>Uncov’d</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defender</td>
<td>$4u^c_\Theta(3)$</td>
<td>$1u^u_\Theta(3)$</td>
</tr>
<tr>
<td>Attacker</td>
<td>$0u^c_\Psi(3)$</td>
<td>$2u^u_\Psi(3)$</td>
</tr>
</tbody>
</table>

Equations to copy to powerpoint

Neal Gupta
Overview

- Motivation and Introduction
- Compact Representations of Security Games
- ERASER
- Alternative LP formulation (ORIGAMI)
- ERASER-C (more realistic)
- Results
Defender commits to a mixed strategy (one of uncountably many)

\[
\Delta = (\delta_{12}, \delta_{13}, \delta_{14}, \delta_{23}, \delta_{24}, \delta_{34})
\]

\[
\forall i, j \ 0 \leq \delta_{ij} \leq 1
\]

\[
\sum_{i,j} \delta_{ij} = m
\]

Attacker strategy is an efficient algorithm, which given any mixed strategy, \( \Delta \) computes target

\[
\arg \max_{t \in \Gamma(\Delta)} U_{\Theta}(\Delta, t)
\]

\[
\Gamma(\Delta) = \{ t : t \in \arg \max U_{\Psi}(\Delta, t) \} 
\]
Compact Representations of Security Games

- *Key insight*: the only information needed to represent the defender strategy is the probabilities a target is covered
- With one attacker (and in extensions with no interactions between attacks) any feasible assignment of resources that achieves coverage vector C will be a SSE that is (weakly) optimal
Overview

- Motivation and Introduction
- Compact Representations of Security Games
  - ERASER
  - Alternative LP formulation (ORIGAMI)
  - ERASER-C (more realistic)
- Results
ERASER – basic formulation

\[
\begin{align*}
\max & \quad d \\
\sum_{t \in T} a_t & \in \{0, 1\} \quad \forall t \in T \\
\sum_{t \in T} c_t & \in [0, 1] \quad \forall t \in T \\
\sum_{t \in T} c_t & \leq m \\
d - U_{\Theta}(t, C) & \leq (1 - a_t) \cdot Z \quad \forall t \in T \\
0 & \leq k - U_{\Psi}(t, C) \leq (1 - a_t) \cdot Z \quad \forall t \in T \\
U_{\Theta}(t, C) & = c_t U^c_{\Theta}(t) + (1 - c_t) U^u_{\Theta}(t)
\end{align*}
\]
ERASER – basic formulation

\[
\begin{align*}
\max & \quad d \\
\sum_{t \in T} a_t & \in \{0, 1\} \quad \forall t \in T \\
\sum_{t \in T} a_t &= 1 \\
c_t & \in [0, 1] \quad \forall t \in T \\
\sum_{t \in T} c_t & \leq m \\
d - U_\Theta(t, C) & \leq (1 - a_t) \cdot Z \quad \forall t \in T \\
0 & \leq k - U_\Psi(t, C) \leq (1 - a_t) \cdot Z \quad \forall t \in T \\
U_\Theta(t, C) &= c_t U_\Theta^c(t) + (1 - c_t) U_\Theta^u(t)
\end{align*}
\]
ERASER – basic formulation

\[
\begin{align*}
\max & \quad d \\
\quad a_t & \in \{0, 1\} \quad \forall t \in T \\
\sum_{t \in T} a_t & = 1 \\
\quad c_t & \in [0, 1] \quad \forall t \in T \\
\sum_{t \in T} c_t & \leq m \\
\quad d - U_\Theta(t, C) & \leq (1 - a_t) \cdot Z \quad \forall t \in T \\
0 & \leq k - U_\Psi(t, C) \leq (1 - a_t) \cdot Z \quad \forall t \in T \\
U_\Theta(t, C) & = c_t U_\Theta^c(t) + (1 - c_t) U_\Theta^u(t)
\end{align*}
\]

Exactly one attack occurs

Feasible use of all resources

Constraint only on attacked target – sets \( d \) equal to that utility
ERASER – basic formulation

\[
\begin{align*}
\text{max} & \quad d \\
\quad a_t & \in \{0, 1\} \quad \forall t \in T \\
\sum_{t \in T} a_t &= 1 \\
\quad c_t & \in [0, 1] \quad \forall t \in T \\
\sum_{t \in T} c_t & \leq m \\
d - U_\Theta(t, C) & \leq (1 - a_t) \cdot Z \quad \forall t \in T \\
0 & \leq k - U_\Psi(t, C) \leq (1 - a_t) \cdot Z \quad \forall t \in T \\
U_\Theta(t, C) &= c_t U_\Theta^c(t) + (1 - c_t) U_\Theta^u(t)
\end{align*}
\]

- Exactly one attack occurs
- Feasible use of all resources
- \( k \) is an upper bound on attacker's utility
ERASER – basic formulation

\[
\begin{align*}
\text{max} & \quad d \\
\sum_{t \in T} a_t & \in \{0, 1\} \quad \forall t \in T \\
\sum_{t \in T} c_t & \leq m \\
\sum_{t \in T} \left( d - U_\Theta(t, C) \right) & \leq (1 - a_t) \cdot Z \quad \forall t \in T \\
0 & \leq k - U_\Psi(t, C) \leq (1 - a_t) \cdot Z \quad \forall t \in T \\
U_\Theta(t, C) & = c_t U_\Theta^c(t) + (1 - c_t) U_\Theta^u(t)
\end{align*}
\]

Exactly one attack occurs

Feasible use of all resources

k is an upper bound on attacker’s utility
ERASER – basic formulation

\[
\begin{align*}
\max & \quad d \\
\sum_{t \in T} a_t & = 1 \\
\sum_{t \in T} c_t & \leq m \\
\sum_{t \in T} \left( d - U_\Theta(t, C) \right) & \leq (1 - a_t) \cdot Z \\
0 & \leq k - U_\Psi(t, C) \leq (1 - a_t) \cdot Z \\
U_\Theta(t, C) & = c_t U^c_\Theta(t) + (1 - c_t) U^u_\Theta(t)
\end{align*}
\]

Exactly one attack occurs
Feasible use of all resources
For the attacked target only, the attacker's utility must be at least \( k \)
ERASER – running example

\[
\begin{align*}
\max d \\
\text{s.t.} \\
\quad a_1 + a_2 + a_3 + a_4 &= 1 \\
\quad c_1 + c_2 + c_3 + c_4 &\leq m \\
\quad d - 4c_1 + (c_1 - 1) &\leq (1 - a_1)Z \\
\quad d - 4c_2 + (c_2 - 1) &\leq (1 - a_2)Z \\
\quad d - 4c_3 + (c_3 - 1) &\leq (1 - a_3)Z \\
\quad d - 4c_4 + (c_4 - 1) &\leq (1 - a_4)Z \\
\quad 0 &\leq k + c_1 - 1 \leq (1 - a_1)Z \\
\quad 0 &\leq k + c_2 - 1 \leq (1 - a_2)Z \\
\quad 0 &\leq k + 2c_3 - 2 \leq (1 - a_3)Z \\
\quad 0 &\leq k + c_4 - 1 \leq (1 - a_4)Z \\
\quad c_t &\in [0, 1] \\
\quad a_t &\in \{0, 1\}
\end{align*}
\]
ERASER – running example

```python
# c_1 \in [0, 1]
# c_2 \in [0, 1]
# c_3 \in [0, 1]
# c_4 \in [0, 1]

from __future__ import print_function
import cplex
from cplex.exceptions import CplexError

my_obj = [1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
my_ub = [cplex.infinity, cplex.infinity, cplex.infinity, cplex.infinity, cplex.infinity, cplex.infinity, cplex.infinity, cplex.infinity, cplex.infinity, cplex.infinity]
lb=0.0
my_lb = [lb, lb, lb, lb, lb, lb, lb, lb, lb, lb]
my_ctype = "CIIICCCCC"
my_colnames = ["d", "a1", "a2", "a3", "a4", "c1", "c2", "c3", "c4", "k"]
Z = 100.0
my_rhs = [1.0, 2.0, Z+1.0, Z+1.0, Z+1.0, 1.0, 1.0, 2.0, 1.0, Z+1.0, Z+1.0, Z+2.0, Z+1.0]
my_rownames = ["r1", "r2", "r3", "r4", "r5", "r6", "r7", "r8", "r9", "r10", "r11", "r12", "r13", "r14"]
my_sense = "ELLLLLGGGGLLLL"

def populatebyrow(prob):
    prob.objective.set_sense(prob.objective.sense.maximize)
```

Modification of IBM’s CPLEX MILP demo code
ERASER – running example

Elapsed time = 0.01 sec. (0.25 ticks, tree = 0.01 MB, solutions = 3)

Root node processing (before b&c):
  Real time = 0.01 sec. (0.25 ticks)
Parallel b&c, 4 threads:
  Real time = 0.00 sec. (0.00 ticks)
  Sync time (average) = 0.00 sec.
  Wait time (average) = 0.00 sec.
  -----------
Total (root+branch&cut) = 0.01 sec. (0.26 ticks)

Solution status = 101 : MIP_optimal
Solution value = 3.14285714286
Row 0: Slack = 0.000000
Row 1: Slack = 0.000000
Row 2: Slack = 99.142857
Row 3: Slack = 99.142857
Row 4: Slack = 0.000000
Row 5: Slack = 99.142857
Row 6: Slack = 0.000000
Row 7: Slack = 0.000000
Row 8: Slack = 0.000000
Row 9: Slack = 0.000000
Row 10: Slack = 100.000000
Row 11: Slack = 100.000000
Row 12: Slack = 0.000000
Row 13: Slack = 100.000000
Column 0: Value = 3.142857
Column 1: Value = -0.000000
Column 2: Value = -0.000000
Column 3: Value = 1.000000
Column 4: Value = 0.000000
Column 5: Value = 0.428571
Column 6: Value = 0.428571
Column 7: Value = 0.714286
Column 8: Value = 0.428571
Column 9: Value = 0.571429
Coverage vector: [0.428571428571, 0.428571428571, 0.714285714286, 0.428571428571]
Adversary attack vector: [-0.0, -0.0, 1.0, 0.0]
mb_pro_umd:mech ngupta$
\[
\begin{align*}
\delta_{12} + \delta_{13} + \delta_{14} &= \frac{3}{7} \\
\delta_{12} + \delta_{23} + \delta_{24} &= \frac{3}{7} \\
\delta_{13} + \delta_{23} + \delta_{34} &= \frac{5}{7} \\
\delta_{14} + \delta_{24} + \delta_{34} &= \frac{3}{7}
\end{align*}
\]

\[0 \leq \delta_{12} \leq 1\]
\[0 \leq \delta_{13} \leq 1\]
\[0 \leq \delta_{14} \leq 1\]
\[0 \leq \delta_{23} \leq 1\]
\[0 \leq \delta_{24} \leq 1\]
\[0 \leq \delta_{34} \leq 1\]
Overview

- Motivation and Introduction
- Compact Representations of Security Games
- ERASER
- Alternative LP formulation (ORIGAMI)
- ERASER-C (more realistic)
- Results
Additional assumption
- Attackers like it when attacks succeed
- Defenders like it when attacks fail

When is this reasonable? Almost always.
- Exception: Honeypots?
- Disputed: Coventry blitz
There is no marginal benefit for either player of adding coverage to targets outside the attack set

Because of tiebreak rule, adding targets (without removing them) from attack set benefits defender
\[ \begin{align*}
\text{min} & \quad k \\
\gamma_t & \in \{0, 1\} \quad \forall t \in T \\
c_t & \in [0, 1] \quad \forall t \in T \\
\sum_{t \in T} c_t & \leq m \\
U_{\Psi}(t, C) & \leq k \quad \forall t \in T \\
k - U_{\Psi}(t, C) & \leq (1 - \gamma_t) \cdot Z \quad \forall t \in T \\
c_t & \leq \gamma_t \quad \forall t \in T \\
U_{\Psi}(t, C) & = c_t U_{\Psi}^C(t) + (1 - c_t) U_{\Psi}^u(t)
\end{align*} \]

Indicator variables for attack set
**ORIGAMI-LP**

\[
\begin{align*}
\min & \quad k \\
\gamma_t & \in \{0, 1\} \quad \forall t \in T \\
c_t & \in [0, 1] \quad \forall t \in T \\
\sum_{t \in T} c_t & \leq m \\
U_\Psi(t, C) & \leq k \quad \forall t \in T \\
k - U_\Psi(t, C) & \leq (1 - \gamma_t) \cdot Z \quad \forall t \in T \\
c_t & \leq \gamma_t \quad \forall t \in T \\
U_\Psi(t, C) &= c_t U^c_\Psi(t) + (1 - c_t) U^u_\Psi(t)
\end{align*}
\]
**ORIGAMI-LP**

\[
\begin{align*}
\text{min} & \quad k \\
\gamma_t & \in \{0, 1\} \quad \forall t \in T \\
c_t & \in [0, 1] \quad \forall t \in T \\
\sum_{t \in T} c_t & \leq m \\
U_\Psi(t, C) & \leq k \quad \forall t \in T \\
k - U_\Psi(t, C) & \leq (1 - \gamma_t) \cdot Z \quad \forall t \in T \\
c_t & \leq \gamma_t \quad \forall t \in T \\
U_\Psi(t, C) & = c_t U_\Psi^c(t) + (1 - c_t) U_\Psi^u(t)
\end{align*}
\]
**ORIGAMI-$$\text{LP}$$**

\[
\begin{align*}
\text{min} & \quad k \\
\gamma_t & \in \{0, 1\} \quad \forall t \in T \\
c_t & \in [0, 1] \quad \forall t \in T \\
\sum_{t \in T} c_t & \leq m \\
U_{\Psi}(t, C) & \leq k \quad \forall t \in T \\
k - U_{\Psi}(t, C) & \leq (1 - \gamma_t) \cdot Z \quad \forall t \in T \\
c_t & \leq \gamma_t \quad \forall t \in T
\end{align*}
\]

\[
U_{\Psi}(t, C) = c_t U_{\Psi}^c(t) + (1 - c_t) U_{\Psi}^u(t)
\]
**ORIGAMI-LP**

\[
\begin{align*}
\min & \quad k \\
\gamma_t & \in \{0, 1\} \quad \forall t \in T \\
c_t & \in [0, 1] \quad \forall t \in T \\
\sum_{t \in T} c_t & \leq m \\
U_\Psi(t, C') & \leq k \quad \forall t \in T \\
k - U_\Psi(t, C') & \leq (1 - \gamma_t) \cdot Z \quad \forall t \in T \\
c_t & \leq \gamma_t \quad \forall t \in T
\end{align*}
\]

\[U_\Psi(t, C) = c_t U^c_\Psi(t) + (1 - c_t) U^u_\Psi(t)\]
ORIGAMI-LP

\[
\begin{align*}
\min & \quad k \\
\gamma_t & \in \{0, 1\} \quad \forall t \in T \\
c_t & \in [0, 1] \quad \forall t \in T \\
\sum_{t \in T} c_t & \leq m \\
U_\Psi(t, C') & \leq k \quad \forall t \in T \\
k - U_\Psi(t, C') & \leq (1 - \gamma_t) \cdot Z \quad \forall t \in T \\
c_t & \leq \gamma_t \quad \forall t \in T
\end{align*}
\]

\[
U_\Psi(t, C) = c_t U_\Psi^c(t) + (1 - c_t) U_\Psi^u(t)
\]

Indicator variables for attack set
Feasible use of all resources
Defender only tries to cover targets in the attack set
**Observation 3.** If $\hat{U}_\Psi(C) = x$, then $c_t \geq \frac{x - U^u_\Psi(t)}{U^c_\Psi(t) - U^u_\Psi(t)}$ for every target $t$ with $U^u_\Psi(t) > x$.

If there was less coverage, the attacker would always attack and get strictly better utility.
Iteration 1
Attack set: \{Target 3\}

New coverage for Target 3: 0.5
Iteration 2
Attack set: \{Target 1, Target 3\}

New coverage for Target 3: 0.5
New coverage for Target 1: 0
ORIGAMI – Running Example

Iteration 3
Attack set: {Target 1, Target 2, Target 3}

New coverage for Target 3: 0.5
New coverage for Target 1: 0
New coverage for Target 2: 0
ORIGAMI – Running Example

Iteration 4
Attack set: {Target 1, Target 2, Target 3, Target 4}

New coverage for Target 3: 0.5
New coverage for Target 1: 0
New coverage for Target 2: 0
New coverage for Target 4: 0
Postprocessing after loop
Attack set: {Target 1, Target 2, Target 3, Target 4}

Sum of ratio\[t\] = 3.5
Resources Left = 1.5

Target 3 final coverage = 0.5 + 0.5*1.5/3.5 = 5/7
Target 1 final coverage = 0 + 1.5/3.5 = 3/7
Target 2 final coverage = 0 + 1.5/3.5 = 3/7
Target 4 final coverage = 0 + 1.5/3.5 = 3/7
ERASER-C

- Solves the FAMS problem
- More realistic – resources now only have a set of schedules they can work with (multiple flights at same time, can’t teleport between cities, etc.)
- Resources have types, analogous to the starting location
ERASER-C

- Now resources only have feasible schedules.
  - Example: 3 guards, 20 flights
  - Feasible schedule: 10 disjoint pairs of flights, e.g. (1, 2), (3,4), (5,6), (7,8), (9, 10), (11, 12), (13, 14), (15, 16), (17, 18), (19, 20)
- Again, naïve representation of pure strategies is infeasibly large

<table>
<thead>
<tr>
<th>Action</th>
<th>Targets</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1,2),(3,4),(5,6)</td>
<td>$p_1$</td>
</tr>
<tr>
<td>2</td>
<td>(1,2),(3,4),(7,8)</td>
<td>$p_2$</td>
</tr>
<tr>
<td>3</td>
<td>(1,2),(3,4),(9,10)</td>
<td>$p_3$</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>120</td>
<td>(15,16),(17,18),(19,20)</td>
<td>$p_{120}$</td>
</tr>
</tbody>
</table>

[Tsai et. al., IRIS A Tool for Strategic Security Allocation in Transportation Networks, 2016]
ERASER-C

• Now resources only have feasible schedules.
  • Example: 3 guards, 20 flights
  • Feasible schedule: 10 disjoint pairs of flights, e.g. (1, 2), (3,4), (5,6), (7,8), (9, 10), (11, 12), (13, 14), (15, 16), (17, 18), (19, 20)
  • Instead put a probability on each schedule.
    • In general problem number of schedules could be infeasibly large, if schedules can be represented compactly
    • However, in FAMS application schedules are length 2

<table>
<thead>
<tr>
<th>Action</th>
<th>Targets</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(1,2)</td>
<td>$p_1$</td>
</tr>
<tr>
<td>2</td>
<td>(3,4)</td>
<td>$p_2$</td>
</tr>
<tr>
<td>3</td>
<td>(5,6)</td>
<td>$p_3$</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>10</td>
<td>(19,20)</td>
<td>$p_{10}$</td>
</tr>
</tbody>
</table>

[Tsai et. al., IRIS A Tool for Strategic Security Allocation in Transportation Networks, 2016]
ERASER-C

\[
\begin{align*}
\text{max} & \quad d \\
\quad a_t & \in \{0, 1\} \quad \forall t \in T \\
\quad c_t & \in [0, 1] \quad \forall t \in T \\
\quad q_s & \in [0, 1] \quad \forall s \in S \\
\quad h_{s,\omega} & \in [0, 1] \quad \forall s, \omega \in S \times \Omega \\
\sum_{t \in T} a_t & = 1 \\
\sum_{\omega \in \Omega} h_{s,\omega} & = q_s \quad \forall s \in S \\
\sum_{s \in S} q_s M(s, t) & = c_t \quad \forall t \in T \\
\sum_{s \in S} h_{s,\omega} Ca(s, \omega) & \leq R(\omega) \quad \forall \omega \in \Omega \\
\quad h_{s,\omega} & \leq Ca(s, \omega) \quad \forall s, \omega \in S \times \Omega \\
\quad d - U_\Theta(t, C) & \leq (1 - a_t) \cdot Z \quad \forall t \in T \\
\quad 0 & \leq k - U_\Psi(t, C) \leq (1 - a_t) \cdot Z \quad \forall t \in T
\end{align*}
\]
ERASER-C – Odd Cycles

- A feasible solution to the MILP may not be feasible using the schedule
- E.g. 1 resource can cover the schedule \{1, 2\} and \{2, 3\}
- It will not be possible to split 50% coverage of 1 and 50% coverage of 3
Overview

- Motivation and Introduction
- Compact Representations of Security Games
- ERASER
- Alternative LP formulation (ORIGAMI)
- ERASER-C (more realistic)
- Results
Results

![Runtime scaling with Targets](image)

![Memory scaling with Targets](image)
Results

![Runtime scaling with Targets](image1)

![Memory scaling with Targets](image2)
Results