BINGHAMTON UNIVERSITY
STATE UNIVERSITY OF NEW YORK
Thomas J. Watson School of Engineering and Applied Science

COMPUTING OPTIMAL RANDOMIZED RESOURCE ALLOCATIONS FOR MASSIVE SECURITY GAMES

COURSE: EECE 580A – CYBER PHYSICAL SYSTEMS SECURITY

PRESENTER: JOSEPH ST. CYR

DATE: DECEMBER 2ND, 2019

SECURITY CONUNDRUM

- Providing security for transportation systems, computer networks, and other critical infrastructure is a large and growing problem.
- Central to many of these security problems is a resource allocation task.
  - E.g., A police force may have limited personnel to conduct patrols, operate checkpoints, conduct random searches, etc.
- *How can we efficiently allocate available resources to protect against a wide variety of potential threats?*

RESOURCE ALLOCATION RANDOMIZATION

- Adversarial aspect of security domains poses unique challenges for resource allocation.
- A motivated attacker can gather information about security measures by surveillance to plan more effective attacks.
- Predictable resource allocations may be exploited by an attacker, greatly reducing resource effectiveness.
- A better approach for deploying security resources is to use randomization to increase the uncertainty of potential attackers.

"COMBINATORIAL EXPLOSION"

- Game theory offers a more sophisticated approach to randomization.
- Allows the analyst to factor differential risks and costs into the game model.
- Incorporates predictions of how the attacker will respond to a given security policy.
- A significant limitation of existing solution methods:
  - Handle multiple resources by exhaustively enumerating all possible combinations (Normal Form).
  - Grows combinatorially in the number of resources, which makes it computationally infeasible to solve large problems.
    - 100 targets and 10 resources yields approx. 1.7×10^{13} allocation combinations (17 Trillion!!!)

GAME-THEORETIC MODELING

- Model the security resource allocation problem as a Stackelberg game.
  - Playing order to game (i.e., leader and follower).
  - Author's security games, the defender is a Stackelberg leader, and the attacker is a follower.
    - This models the capability of malicious attackers to employ surveillance in planning attacks.
- Defender Θ, and an attacker Ψ
- Each player has a set of possible pure strategies, denoted $\sigma_\Theta \in \Sigma_\Theta$ and $\sigma_\Psi \in \Sigma_\Psi$.
- A mixed strategy allows a player to play a probability distribution over pure strategies, denoted $\delta_\Theta \in \Delta_\Theta$ and $\delta_\Psi \in \Delta_\Psi$.

GAME-THEORETIC MODELING

- Payoffs for each player are defined over all possible joint pure strategy outcomes: $\Omega_\Theta : \Sigma_\Psi \times \Sigma_\Theta \rightarrow \mathbb{R}$ for the defender and similarly for the attacker.
- The payoff functions are extended to mixed strategies.
- Formally, the attacker's strategy in a Stackelberg security game is a function that selects a strategy in response to each leader strategy: $F_\Psi : \Delta_\Theta \rightarrow \Delta_\Psi$.

STRONG STACKELBERG EQUILIBRIUM

- Stackelberg equilibrium is a refinement of Nash equilibrium.
- Form of subgame perfect equilibrium, such that each player chooses a best-response.
- Eliminates Nash equilibrium profiles supported by non-credible threats off the equilibrium path.
- *DEFINITION 1*. A pair of strategies (δ_Θ, F_Ψ) form a Strong Stackelberg Equilibrium (SSE) if they satisfy the following:
  - The leader plays a best-response: $\Omega_\Theta(\delta_\Theta, F_\Psi(\delta_\Theta)) \geq \Omega_\Theta(\delta'_\Theta, F_\Psi(\delta'_\Theta))\ \forall\ \delta'_\Theta \in \Delta_\Theta$.
  - The follower plays a best-response: $\Omega_\Psi(\delta_\Theta, F_\Psi(\delta_\Theta)) \geq \Omega_\Psi(\delta_\Theta, \delta_\Psi)\ \forall\ \delta_\Theta \in \Delta_\Theta, \delta_\Psi \in \Delta_\Psi$.
  - The follower breaks ties optimally: $\Omega_\Theta(\delta_\Theta, F_\Psi(\delta_\Theta)) \geq \Omega_\Theta(\delta_\Theta, \delta_\Psi)\ \forall\ \delta_\Theta \in \Delta_\Theta, \delta_\Psi \in \Delta^*_\Psi(\delta_\Theta)$, where $\Delta^*_\Psi(\delta_\Theta)$ is the set of follower best-responses, as above.

A COMPACT REPRESENTATION OF RESOURCES

- Let $T = \{t_1, \ldots, t_n\}$ be a set of targets that may be attacked, corresponding to pure strategies for the attacker.
- The defender has a set of resources available to "cover" these targets, $R = \{r_1, \ldots, r_m\}$.
- Associated with each target are four payoffs defining the possible outcomes for an attack on the target, as shown in Table 1.

Table 1: Example payoffs for an attack on a target.

	Covered	Uncovered
Defender	5	–20
Attacker	–10	30

A COMPACT REPRESENTATION OF RESOURCES

- There are two cases, depending on whether or not the target is covered by the defender.
  - The defender's payoff: Uncovered attack = $U^u_\Theta(t)$, Covered attack = $U^c_\Theta(t)$.
  - The attacker's payoff: Uncovered attack = $U^u_\Psi(t)$, Covered attack = $U^c_\Psi(t)$.
- A crucial feature:
  - Payoffs only depend on the identity of the attacked target and whether or not it is covered by the defender.
  - Does not matter whether or not any un-attacked target is covered or not.

A COMPACT REPRESENTATION OF RESOURCES

- Exploited by summarizing the payoff-relevant aspects:
  - Defender: coverage vector, C, that gives the probability that each target is covered, c_t.
  - Attacker: attack vector, A, that gives the probability that each target is attacked, a_t.

$$
\begin{aligned}
U_\Theta(C,A) &= \sum_{t\in T} a_t \cdot (c_t \cdot U_\Theta^c(t) + (1-c_t)U_\Theta^u(t)) \quad (1)\\
U_\Theta(t,C) &= c_t U_\Theta^c(t) + (1-c_t)U_\Theta^u(t) \quad (2)\\
\Gamma(C) &= \{t : U_\Psi(t,C) \geq U_\Psi(t',C)\ \forall\, t' \in T\}. \quad (3)
\end{aligned}
$$

$$
\begin{aligned}
\delta_\Theta^{1,2} + \delta_\Theta^{1,3} + \delta_\Theta^{1,4} &= c_1\\
\delta_\Theta^{1,2} + \delta_\Theta^{2,3} + \delta_\Theta^{2,4} &= c_2\\
\delta_\Theta^{1,3} + \delta_\Theta^{2,3} + \delta_\Theta^{3,4} &= c_3\\
\delta_\Theta^{1,4} + \delta_\Theta^{2,4} + \delta_\Theta^{3,4} &= c_4 \quad (4)
\end{aligned}
$$

- In a strong Stackelberg equilibrium:
  - Attacker selects the target in the attack set with maximum payoff for the defender.
  - Let t^* denote optimal target.
  - The expected SSE payoff for the defender is $\hat{U}_\Theta(C) = U_\Theta(t^*, C)$, and $\hat{U}_\Psi(C) = U_\Psi(t^*, C)$ for the attacker.

"ERASER" SOLUTION ALGORITHM

- The **ERASER** algorithm (**E**fficient **R**andomized **A**llocation of **SE**curity **R**esources) takes as input a security game in compact form and solves for an optimal coverage vector corresponding to a SSE strategy for the defender.
- The algorithm is a mixed-integer linear program (MILP).
- C and A are mutual best-responses in any optimal solution.

$$
\begin{aligned}
\max \quad & d && && (5)\\
a_t \in \quad & \{0,1\} && \forall t \in T && (6)\\
\sum_{t \in T} a_t = \quad & 1 && && (7)\\
c_t \in \quad & [0,1] && \forall t \in T && (8)\\
\sum_{t \in T} c_t \leq \quad & m && && (9)\\
d - U_\Theta(t,C) \leq \quad & (1-a_t)\cdot Z && \forall t \in T && (10)\\
0 \leq k - U_\Psi(t,C) \leq \quad & (1-a_t)\cdot Z && \forall t \in T && (11)
\end{aligned}
$$

EXPLOITING PAYOFF STRUCTURE

- Consider a class of security games in which the defender always benefits by having additional resources covering an attacked target, while the attacker is always worse off attacking a more heavily defended target.
- Formally, we restrict payoff functions so that:
  - $U^u_\Theta(t) < U^c_\Theta(t)$ and $U^u_\Psi(t) > U^c_\Psi(t)$ for all t.

OBSERVATION 1. *All else equal, increasing c_t for any target not in $\Gamma(C)$ has no effect on $\hat{U}_\Theta(C)$ or $\hat{U}_\Psi(C)$.*

OBSERVATION 2. *If $\Gamma(C) \subset \Gamma(C')$ and $c_t = c'_t$ for all $t \in \Gamma(C)$ then $\hat{U}_\Theta(C) \leq \hat{U}_\Theta(C')$.*

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 $U^u_\Psi(t) > x$.*

"ORIGAMI" SOLUTION ALGORITHM

- Observations 1-3 are exploited in the **ORIGAMI** algorithm (**O**ptimizing **R**esources **I**n **GA**mes using **M**aximal **I**ndifference).
  - Directly compute the attack set for the attacker, using the indifference equation in Observation 3.
- Starting with a target that has maximal U^u_Ψ (t), the attack set is expanded at each iteration in order of decreasing U^u_Ψ(t).
- Each time the attack set is expanded, the coverage of each target is updated to maintain the indifference of attacker payoffs within the attack set.

"ORIGAMI" SOLUTION ALGORITHM

- Two termination conditions:
  - 1 - Occurs when adding the next target to the attack set requires more total coverage probability than the defender has resources available.
  - 2 - Occurs when any target t is covered with probability 1.

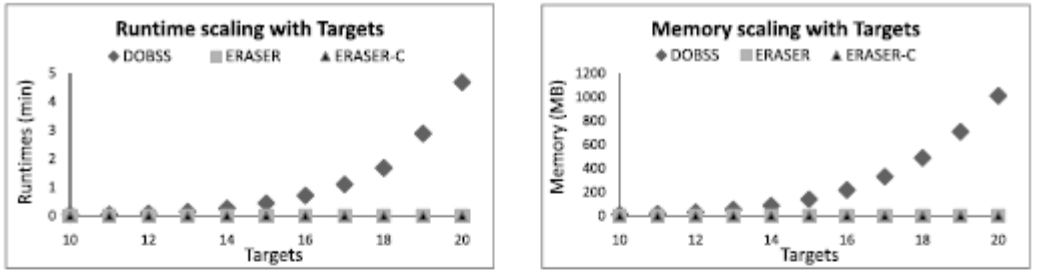
Algorithm 1 ORIGAMI

targets $\leftarrow T$ sorted by $U_\Psi^u(t)$
payoff[t] $\leftarrow U_\Psi^u(t)$, coverage[t] $\leftarrow 0$
left $\leftarrow m$, next $\leftarrow 2$
covBound $\leftarrow -\infty$
while next $\leq n$ **do**
  addedCov[t] $\leftarrow \frac{payoff[next]-U_\Psi^u(t)}{U_\Psi^c(t)-U_\Psi^u(t)}$ - coverage[t]
  if coverage[t] + addedCov[t] ≥ 1 **then**
    covBound $\leftarrow$ Max(covBound, $U_\Psi^c(t)$)
  end if
  if covBound $\geq -\infty$ OR $\sum_{t\in T}$ addedCov[t] $\leq$ left **then**
    BREAK
  end if
  coverage[t] += addedCov[t]
  left -= $\sum_{t\in T}$ addedCov[t]
  next++
end while
ratio[t] $\leftarrow \frac{1}{U_\Psi^u(t)-U_\Psi^c(t)}$
coverage[t] += $\frac{ratio[t]\cdot left}{\sum_{t\in T} ratio[t]}$
if coverage[t] ≥ 1 **then**
  covBound $\leftarrow$ Max(covBound, $U_\Psi^c(t)$)
end if
if covBound $\geq -\infty$ **then**
  coverage[t] $\leftarrow \frac{covBound-U_\Psi^u(t)}{U_\Psi^c(t)-U_\Psi^u(t)}$
end if

OTHER ALGORITHMS

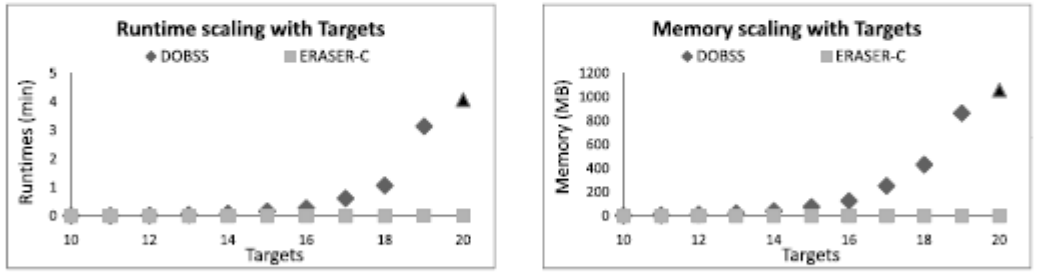
- ERASER-C
  - An extension of ERASER which adds the capability to represent certain kinds of resource and scheduling constraints, motivated by the real example domains described previously.
- ORIGAMI-MILP
  - Similar to ERASER, but does not optimize the defender's payoff. Instead, it minimizes the attacker's payoff, and adds a constraint that restricts c_t for any t not in $\Gamma(C)$ to 0, consistent with Observation 1. This constraint forces the attack set to include the maximal number of targets.

EXPERIMENTAL RESULTS



(a) Runtimes for DOBSS, ERASER, and ERASER-C

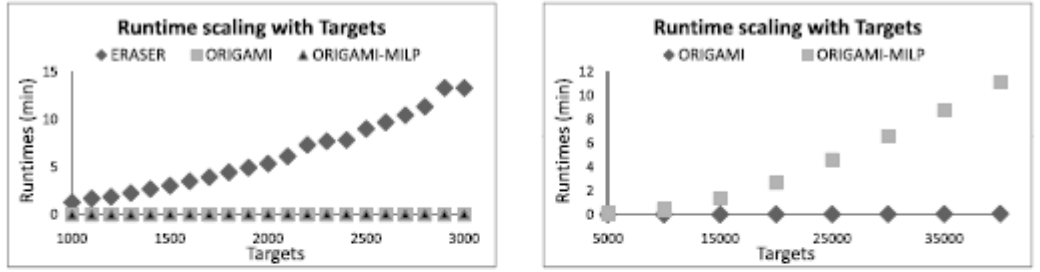
(b) Memory use of DOBSS, ERASER, and ERASER-C



(c) Runtimes for DOBSS and ERASER-C

(d) Memory use of DOBSS and ERASER-C

Figure 1: Runtime and memory scaling



(a)

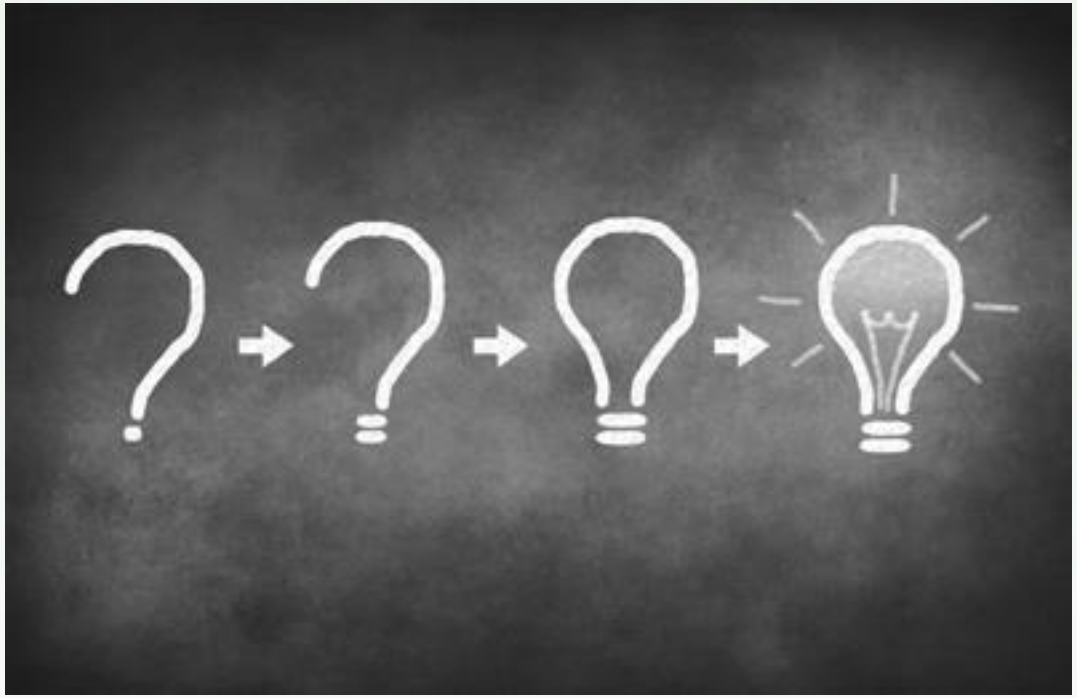
(b)

Figure 2: Runtime scaling of ERASER, ORIGAMI, and ORIGAMI-MILP

	Actions	DOBSS	ERASER (-C)
LAX (6 canines)	784	0.94s	0.23s
FAMS (small)	~6,000	4.74s	0.09s
FAMS (large)	~85,000	435.6s*	1.57s

Table 2: Runtimes on real data.

QUESTIONS & DISCUSSION



REFERENCES

1. Kiekintveld, Christopher & Jain, Manish & Tsai, Jason & Pita, James & Ordóñez, Fernando & Tambe, Milind. (2009). Computing optimal randomized resource allocations for massive security games. AMAAS-2009 Conf. 2. 689-696. 10.1145/1558013.1558108.