Detection and Mitigation of Cyber-Attacks using Game Theory

João P. Hespanha

Kyriakos G. Vamvoudakis
Mission to Cyber Assets

Analysis to get up-to-date view of cyber-assets

Analysis to determine dependencies between assets and missions

Mission Model to Cyber-Assets Model

Sensor Alerts to Correlation Engine

Impact Analysis to Create semantically-rich view of cyber-mission status

Simulation/Live Security Exercises to Analyze and Characterize Attackers

Predict Future Actions to COAs

Observations: Netflow, Probing, Time analysis
Detection in adversarial environments

Multi-agent learning for distributed consensus/agreement under cyber-attack

Online optimization for real-time attack prediction and human-in-the-loop experiments
Detection in adversarial environments

Multi-agent learning for distributed consensus/agreement under cyber-attack

Online optimization for real-time attack prediction and human-in-the-loop experiments
How to interpret & access the reliability of “sensors” that have been manipulated?

“Sensors” relevant to cyber missions?
• Measurement sensors (e.g., determine presence/absence of friendly troops)
• Computational sensors (e.g., weather forecasting simulation engines)
• Data retrieval sensors (e.g., database queries)
• Cyber-security sensors (e.g., IDSs)

Domains
• Deterministic sensors:
  with $n$ sensors, one can get correct answer as long as $m < n/2$ sensors have been manipulated
• Stochastic sensors without manipulation:
  solution given by hypothesis testing/estimation
  $$P(\text{sensor error}) = p_{\text{err}} \quad \Rightarrow \quad P(\text{n sensor error}) \approx \binom{n}{n/2} p_{\text{err}}^{n/2}$$
• Stochastic sensors with potential manipulation: open problem?
Problem formulation

$X$ – binary random variable to be estimated

$$P(X = 0) = P(X = 1) = \frac{1}{2}$$

for simplicity
(papers treats general case)

$Y_1, Y_2, \ldots, Y_n$ – “noisy” measurements of $X$ produced by $n$ sensors

$$P(Y_i \neq X) = p_{err} \quad P(Y_i = X) = 1 - p_{err} \quad \forall i$$

per-sensor error probability
(not necessarily very small)

$Z_1, Z_2, \ldots, Z_n$ – measurements actually reported by the $n$ sensors

$$Z_i = \begin{cases} Y_i & \text{sensor } i \text{ not attacked} \\ ? & \text{sensor } i \text{ attacked} \end{cases}$$

at most $m$ sensors attacked

$p_{\text{attack}}$ – probability that we are under attack (very hard to know!)

interpretation of sensor data should be mostly independent of $p_{\text{attack}}$
Theorem:

\[ P_{\text{attack}} \leq \left( 1 + \frac{1}{n} \left( \frac{n-m}{n-1} \right) \right) \frac{p_{\text{err}}^{n-m} - \frac{n-1}{2} \left( 1 - p_{\text{err}} \right)^{n-1} - p_{\text{err}}^{n-1} (1 - p_{\text{err}})}{p_{\text{err}} (1 - p_{\text{err}})^{n-1} - p_{\text{err}}^{n-1} (1 - p_{\text{err}})} \]

The optimal estimator is largely independent of \( p_{\text{attack}} \) (hard to know)

\[ \begin{align*}
J_0 &:= \sum_{k=0}^{n-1} \binom{n}{k} p_{\text{err}}^n (1 - p_{\text{err}})^k \\
\rho &:= \sum_{k=m}^{n-1} \binom{n-m}{k-m} p_{\text{err}}^n (1 - p_{\text{err}})^{k-m} \\
\gamma &:= \sum_{k=0}^{n-1} \binom{n-m}{k} p_{\text{err}}^n (1 - p_{\text{err}})^k \\
\beta &:= \frac{1 - P_{\text{attack}}}{P_{\text{attack}}} ((1 - p_{\text{err}})^n - p_{\text{err}}^n)
\end{align*} \]

The optimal estimator is

\[ \begin{cases} 
\mu_{\text{majority}} & \text{w.p. } 1 - y_2 \\
\mu_{\text{no-consensus}} & \text{w.p. } y_2
\end{cases} \]

\[ y_2 = \begin{cases} 
\prod_{[0,1]} \left( \frac{\gamma - \rho}{(1 - p_{\text{err}})^{n-m} + p_{\text{err}}^{n-m}} \right) & \beta \leq p_{\text{err}}^{n-m} \\
0 & \beta > p_{\text{err}}^{n-m}
\end{cases} \]

go with the majority, EXCEPT if there is consensus
Result for “small” # of sensors \((n<2/p_{\text{err}})\)

Theorem:

\[ p_{\text{attack}} \leq \left( 1 + \frac{1}{n} \left( n - m \right) \frac{p_{\text{err}}^{n-m} - p_{\text{err}}^{n-1}}{(1 - p_{\text{err}})^{n-1} - p_{\text{err}}^{n-1}(1 - p_{\text{err}})} \right)^{-1} \]

The optimal estimator is

- go with the majority of the (potentially manipulated) sensor readings
- go with the majority, EXCEPT if there is consensus

The optimal estimator is largely independent of \(p_{\text{attack}}\) (hard to know)
**Theorem:** For every $p_{\text{err}}, p_{\text{attack}}$ the estimator

\[ \hat{X} = \begin{cases} 1 & \# \text{ 1's} \geq z^* \\ 0 & \# \text{ 1's} < z^* \end{cases} \quad \text{threshold like rule} \]

\[ z^* := \max \left\{ \frac{n}{2} + \frac{\sigma^2}{m + (n - m)(2p_{\text{err}} - 1)} \log \frac{P(X = 1)}{P(X = 0)}, \frac{n - m}{2} + \frac{\sigma^2}{(n - m)(2p_{\text{err}} - 1)} \log \frac{P(X = 1)}{P(X = 0)} \right\} \]

is away from optimal, at most by,

\[ \epsilon \leq \frac{.12 + .18|2p_{\text{err}} - 1|}{(n - m)p_{\text{err}}(1 - p_{\text{err}})} + e^{-\frac{3\sqrt{(n-m)p_{\text{err}}(1-p_{\text{err}})}}{2}} \]

small when we have enough sensors
(needed to overcome large $p_{\text{err}}$)

- All of this is independent of $p_{\text{attack}}$ (hard to know)
- Specific bounds on probability of being fooled by attacker
General Case

Estimate random variable $X$ based on a measurement

$$Z = \begin{cases} 
R & \text{w.p. } 1 - p_{\text{attack}} \\
S + A & \text{w.p. } p_{\text{attack}} 
\end{cases} \quad \text{no attack}$$

$R, S$ have distributions that depend on value of $X$
(reflecting a potentially noisy measurement)
$A$ has a distribution “selected” by the attacker
(limited to a support that reflects some limitation of the attacker)

Goal: Use measurement $Z$ to estimate the value of $X$, minimizing error probability

Game theoretical formulation: attacker wants to maximize error probability

E.g., $R \equiv \#$ sensors reporting $X = 1$, no errors
$R = \begin{cases} 
n & X = 1 \\
0 & X = 0 
\end{cases}$

$S \equiv \#$ non-attacked sensors reporting $X = 1$, no errors
$S = \begin{cases} 
n - m & X = 1 \\
0 & X = 0 
\end{cases}$

$A \in \{0, 1, \ldots, m\} \equiv \#$ of sensors that the attacker decides should report $X = 1$
Estimate random variable $X$ based on a measurement

$$Z = \begin{cases} R & \text{w.p. } 1 - p_{\text{attack}} \\ S + A & \text{w.p. } p_{\text{attack}} \end{cases}$$

no attack

attack

**Corollary:** Suppose $A \in [a, b]$ and $R, S$ are Gaussian with

$$\max \left\{ \frac{b + \sigma_0}{\sigma^2}, \frac{\rho_0}{\sigma^2} \right\} < \min \left\{ \frac{a + \sigma_1}{\sigma^2}, \frac{\rho_1}{\sigma^2} \right\}$$

Then optimal estimator is a threshold rule

$$\hat{X} := \begin{cases} 0 & Z < z^* \\ 1 & Z \geq z^* \end{cases}$$

threshold $z^*$ is an easily computable quantity

- we have a general theorem for arbitrary continuous distributions
- for discrete distributions this leads to $\varepsilon$-saddle points
  (by approximating discrete by continuous distributions)

Detection in adversarial environments (summary of new results)

Multi-agent learning for distributed consensus/agreement under cyber-attack

Online optimization for real-time attack prediction and human-in-the-loop experiments
In complex cyber missions,
- human operators define policies and rules
- computing elements automate processes of distributed resource allocation, scheduling, inventory management, etc.

- **self-configuration**: automatic configuration of components
- **self-healing**: automatic discovery and correction of faults
- **self-optimization**: automatic allocation of resources for optimal operation

What is the impact of attacks on this type of automated/optimization process? Can we devise algorithms with built-in attack prediction/awareness capabilities?
Focus: Distributed Consensus/Agreement

Classical problem in distributed computing:
• A group of computing elements must agree on a common scalar value $x$ (e.g., priority, resources allocated, inventory decision, database value)
• Decision done iteratively & distributed using peer-to-peer communication

\[
x_i(k + 1) = x_i(k) + \Delta x_i(k)
\]
\[
\Delta x_i(k + 1) = \Delta x_i(k) + u_i
\]

second-order or double-integrator algorithm:
• computers do not update directly $x_i$
• instead, update the per-iteration adjustment on $x_i$
• slower, but smoother adjustments of the $x_i$

**Goal:** minimize errors between values of agents and their neighbors

\[
e_i := \sum_{j \in \mathcal{N}_i} \left[ \frac{x_i - x_j}{\Delta x_i - \Delta x_j} \right]
\]
Focus: Distributed Consensus/Agreement

Classical problem in distributed computing:
• A group of computing elements must agree on a common scalar value $x$ (e.g., priority, resources allocated, inventory decision, database value)
• Decision done iteratively & distributed using peer-to-peer communication

Goal: minimize errors between values of agents and their neighbors

$$x_i(k+1) = x_i(k) + \Delta x_i(k)$$
$$\Delta x_i(k+1) = \Delta x_i(k) + u_i + v_i$$

Goal: minimize errors between values of agents and their neighbors

$$e_i := \sum_{j \in N_i} \left[ \frac{x_i - x_j}{\Delta x_i - \Delta x_j} \right]$$

Attacker: maximize errors using stealth attacks (small $v_i$)
Classical problem in distributed computing:
• A group of computing elements must agree on a common scalar value $x$ (e.g., priority, resources allocated, inventory decision, database value)
• Decision done iteratively & distributed using peer-to-peer communication

\[
x_i(k + 1) = x_i(k) + \Delta x_i(k)
\]
\[
\Delta x_i(k + 1) = \Delta x_i(k) + u_i + v_i
\]

Nash equilibrium formulation:
\[
J_i = \sum_k \left( \|e_i\|^2 + \sum_{j \in \mathcal{N}_i} \left( \|u_j\|^2 - \gamma_{ij}^2 \|v_j\|^2 \right) \right)
\]
\[
e_i := \sum_{j \in \mathcal{N}_i} \left[ \frac{x_i - x_j}{\Delta x_i - \Delta x_j} \right]
\]

- error \(\min.\) by us, \(\max.\) by attacker
- our updates \(\min.\) by us, \(\max.\) by attacker
- attacker updates \(\min.\) by us, \(\max.\) by attacker
Optimal Solution

Bellman Equation

\[
\frac{\partial V_i}{\partial e_i}^T \left( \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} e_i - \sum_{j \in N_i} (u_i + v_i) \right) + \frac{1}{2} \left( \|e_i\|^2 + \sum_{j \in N_i} (\|u_j\|^2 - \gamma_{ij}^2 \|v_j\|^2) \right) = 0
\]

Optimal Control and Attacker Policies

\[
u_i^* = \frac{d_i}{\gamma_{ii}^2} \begin{bmatrix} 0 & I \end{bmatrix} \frac{\partial V_i}{\partial e_i}
\]

Under appropriate regularity assumptions (smoothness)

\[J_i(u_i^*, u_{-i}^*, v^*) \leq J_i(u_i, u_{-i}^*, v^*) \quad \forall u_i \quad u_i^* \text{ is optimal (minimal) for us}
\]

\[J_i(u^*, v_i^*, v_{-i}^*) \geq J_i(u^*, v_i, v_{-i}^*) \quad \forall v_i \quad v_i^* \text{ is optimal (maximal) for attacker}
\]

Moreover,

- Consensus will be reached asymptotically
- All variables will remain bounded through the transient (in fact, Lyapunov stability)

Theoretical results derived for a continuous-time approximation of the algorithms, more suitable for the asymptotic analysis
Optimal Solution

Bellman Equation

\[
\frac{\partial V_i}{\partial e_i}^T \left( \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} e_i - \sum_{j \in \mathcal{N}_i} (u_i + v_i) \right) + \frac{1}{2} \left( \|e_i\|^2 + \sum_{j \in \mathcal{N}_i} \left( \|u_j\|^2 - \gamma_{ij}^2 \|v_j\|^2 \right) \right) = 0
\]

Optimal Control Policies

But…

Bellman equation difficult to solve (curse of dimensionality)

Under appropriate regularity assumptions (smoothness)

\[
J_i(u_i^*, u_{-i}^*, v^*) \leq J_i(u_i, u_{-i}^*, v^*) \quad \forall u_i \quad u_i^* \text{ is optimal (minimal) for us}
\]
\[
J_i(u^*, v_i^*, v_{-i}^*) \geq J_i(u^*, v_i, v_{-i}^*) \quad \forall v_i \quad v_i^* \text{ is optimal (maximal) for attacker}
\]

Moreover,

• Consensus will be reached asymptotically
• All variables will remain bounded through the transient (in fact, Lyapunov stability)

Theoretical results derived for a continuous-time approximation of the algorithms, more suitable for the asymptotic analysis
Critic = (distributed) algorithm to evaluate the current algorithm & estimate attacker actions
Actor = (distributed) algorithm to enact optimal decisions (based on critic’s findings)

Actor & Critic based on Approximate Dynamic Programming
(Neural Network approximation of value function & optimal control laws)
**Learning Structure**

**Critic Neural Networks** to approximate the costs

\[ \hat{V}_i = \hat{W} e_i^T \phi_i(e_i) \]

unknown weights to be learned
basis sets, with local state information

**Actor Neural networks** to approximate the control and adversarial inputs

\[ \hat{u}_i = -d_i \left[ \begin{array}{l} 0 \\ I \end{array} \right]^T \frac{\partial \phi_i}{\partial e_i} \hat{W} u_i; \quad \hat{v}_i = \frac{d_i}{\gamma_{ii}^2} \left[ \begin{array}{l} 0 \\ I \end{array} \right]^T \frac{\partial \phi_i}{\partial e_i} \hat{W} v_i. \]

**Approximate Bellman equations**

\[
H_i(e_i, \hat{W} c_i, \hat{u}_i, \hat{u}_{N_i}, \hat{v}_i, \hat{v}_{N_i}) = \\
\hat{W} c_i^T \frac{\partial \phi_i}{\partial e_i} \left( \begin{array}{ll} 0 & I \\ 0 & 0 \end{array} \right) \left( \begin{array}{l} s_i + d_i \left[ \begin{array}{l} 0 \\ I \end{array} \right] (\hat{u}_i + \hat{v}_i) - \sum_{j \in N_i} \left[ \begin{array}{l} 0 \\ I \end{array} \right] (\hat{u}_j + \hat{v}_j) \right) + r_i \equiv E_i.
\]

**Critic Tuning Update Law**
Learn the value
\[ \dot{\hat{W}} c_i = -\alpha_i \frac{\omega_i}{(\omega_i^T \omega_i + 1)^2} E_i^T \]

**Control Input Tuning Update Law**
Learn the optimal control
\[ \dot{\hat{W}} u_i = \mathcal{P} r \left[ \sigma_{ui}(\hat{W} c_i - \hat{W} u_i) + \ldots \right] \]

**Attacker Tuning Update Law**
Learn the attacker
\[ \dot{\hat{W}} v_i = \mathcal{P} r \left[ \sigma_{vi}(\hat{W} c_i - \hat{W} v_i) + \ldots \right] \]
The scheme shown is implemented inside each agent, it uses an approximation of the attacker behavior.
Learning under attacks key result

**Assuming**
- The signals are **persistently exciting**.
- The graph is **strongly connected**.
- Number of basis functions is **large**.

**Then**
- All the signals are **Uniformly Ultimately Bounded**
- The policies converge to an **approximate Nash equilibrium**

Baseline linear consensus algorithm
Under attack, consensus is not reach

Proposed consensus algorithm
Consensus is reach, even under attack
Outline...

- Detection in adversarial environments (summary of new results)
- Multi-agent learning for distributed consensus/agreement under cyber-attack
- Online optimization for real-time attack prediction and human-in-the-loop experiments
**Motivation**

**Features of Cyber Missions:**
- cyber assets shared among missions
- need for cyber asset change over time
- missions can use different configurations of resources
- complex network of cyber-asset dependencies

**Cyber Awareness Questions:**
- When & where is an attacker most likely to strike?
- When & where is an attacker be more damaging to mission completion?
- How will the answer depend on attacker resources? attacker skills? attacker knowledge?

(Real-time what-if analysis)
• Mission requires multiple services
• Mission reliance on services varies with time

**Damage equation:** 
\[ P D^s_t \approx a^s_t + b^s_t \ AR^s_t \]

- Potential damage
- Attack resources

**Uncertainty equation:** 
\[ p^s_t \approx \prod_{[0,1]} \left( c^s_t - d^s_t \ AR^s_t \right) \]

- Probability of realizing damage
- Attack resources

**Optimal attacks:**

\[
TD = \sum_t \sum_s f^s_t(AR^s_t)g(AR^s_t)
\]

- Total damage to mission

\[
\sum_s AR^s_t \leq TR_t, \ \forall t
\]

- Total attack resources at time \( t \)

\[
\sum_t \sum_s AR^s_t \leq TR
\]

- Total attack resources
2011 iCTF

- Distributed, wide-area security exercise to test the security skills of the participants, organized yearly by Prof. Giovanni Vigna (UCSB)
- The 2011 iCTF involved 89 teams from around the world (1000+ participants)

- Cyber assets shared among missions
- Mission needs (in terms of cyber assets) change over time
- Attackers (teams)
  - receive cues about importance of cyber assets (*payoff* & *cut*)
  - receive cues about uncertainty in inflicting damage (*risk*)
  - must decide how to efficiently use resources (*money*)
- Team that produces most damage wins (*points*)

[Diagram showing the setup of the iCTF exercise with teams and cyber assets]
What did the teams do?

• teams got right the top services to attack
• teams ignored the lowest yielding services (even when “profitable”)
• top teams correctly identified the best 4 services to attack & invested resources more judiciously

But...

• Our analysis ignored:
  • which services are more difficult to attack
  • attackers’ capabilities

Actual damage to mission due to attacks on different services, by top 10 teams

Actual damage to mission due to attacks on different services, by all teams

Optimal actual damage to mission due to attacks on different services
Integrated Attack Predictions

maximize

\[ TD = \sum_t \sum_s f_t^s(A_R_t^s)g(A_R_t^s) \]

constrained by:

\[ \sum_s A_R_t^s \leq T_R, \quad \forall t \quad \sum_t \sum_s A_R_t^s \leq T_R \]

Optimal attack predictions capture solely mission & required cyber assets

• historical data
• live security exercises
• simulation (e.g., CyMiR, see Giovanni’s talk)

Predictions capturing
• mission & required cyber assets
• Simulation, exercises, historical data
Integrated Attack Predictions

Mission Model

Cyber-Assets Model

maximize

\[ TD = \sum_t \sum_s f_t^s(AR_t^s)g(AR_t^s) \]

constrained by:

\[ \sum_s AR_t^s \leq TR_t, \quad \forall t \]

\[ \sum_t \sum_s AR_t^s \leq TR \]

\[ PL^s(t + 1) := \frac{SP^s(0 \cdots t)}{SP^s(0 \cdots t) + FP^s(0 \cdots t)} \in [0, 1] \]

- Plausibility of next attack prediction for service s
  (based on historical data)
- Informs security officer of attackers capabilities
- Integrated measure of threat to different services

Joint work with T. Hollerer and R. Choudhury

more on this shortly…
Summary of Accomplishments (Y4)

Estimation and detection in adversarial environments
- Extension of results to large # sensors, prob. of sensor errors, prob. of attack
- Generalization to formulation that can cover a much wider class of adversarial estimation problems

Distributed consensus/agreement under cyber-attack
- Developed Dynamic Programming-based solution to obtain resilient algorithms with attack estimates. Used online learning approach to overcome complexity issues
- Developed set-based solution to detect attack (see report).

Online optimization for real-time attack prediction and human-in-the-loop experiments
- Optimization framework to predict maximally damaging attacks based on the mission model
- Plausability metric to integrate simulation data, life security exercises, historical data
- Results incorporated in visualization tools (more on Tobias Hollerer’s presentation)

Focus for next review period
- Generalize the learning framework to other distributed computing problem that arise in cyber missions
- Develop numerical algorithms for fast (real-time) attack prediction into iCTF 2012 data (preliminary work under way, not reported today).
- Further integration with visualization tools and human-subject testing of tools developed
Published last year or in press


Submitted
