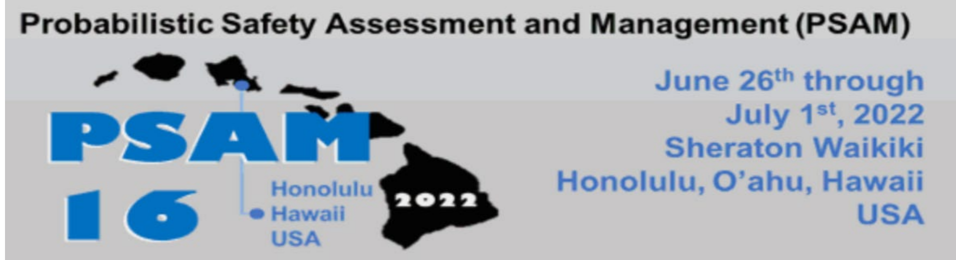
Reinforcement Learning based Autonomous Cyber Attack Response in Nuclear Power Plants

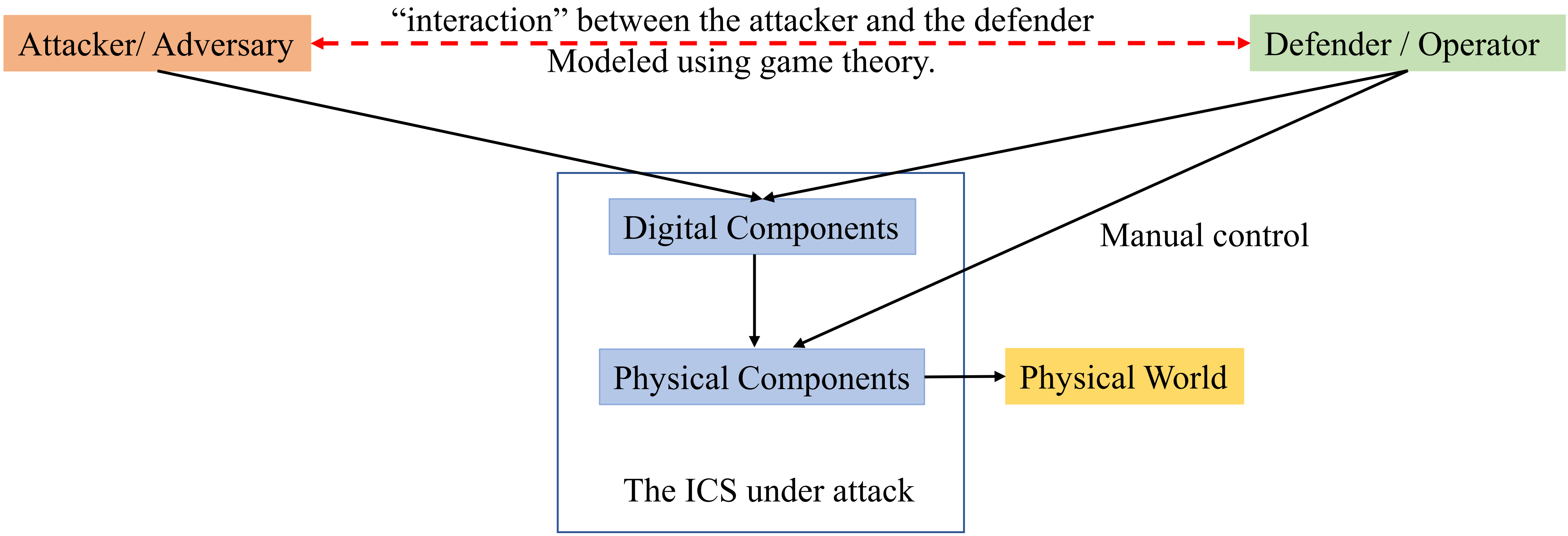
Pavan Kumar Vaddi, Yunfei Zhao, and Carol Smidts
Department of Mechanical and Aerospace Engineering, The Ohio State University.
June 29, 2022.

THE OHIO STATE UNIVERSITY

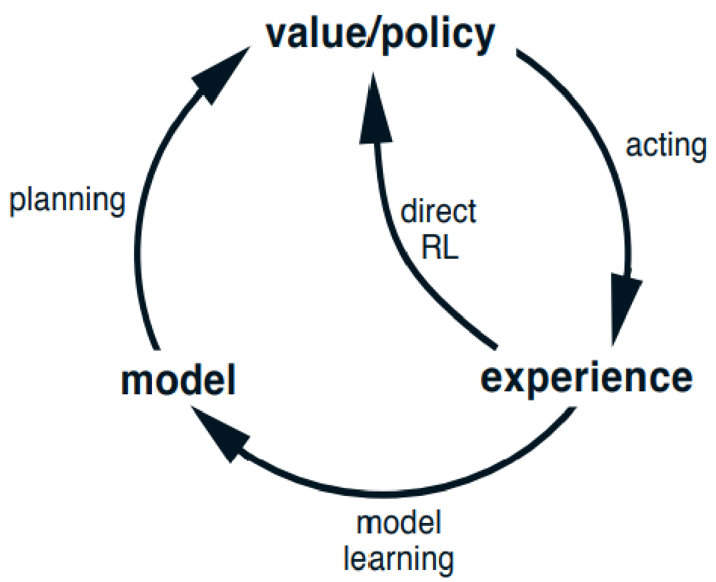


Introduction

- Cyber-Attack response is a sequential decision-making problem that requires consideration of attacker-defender interactions.



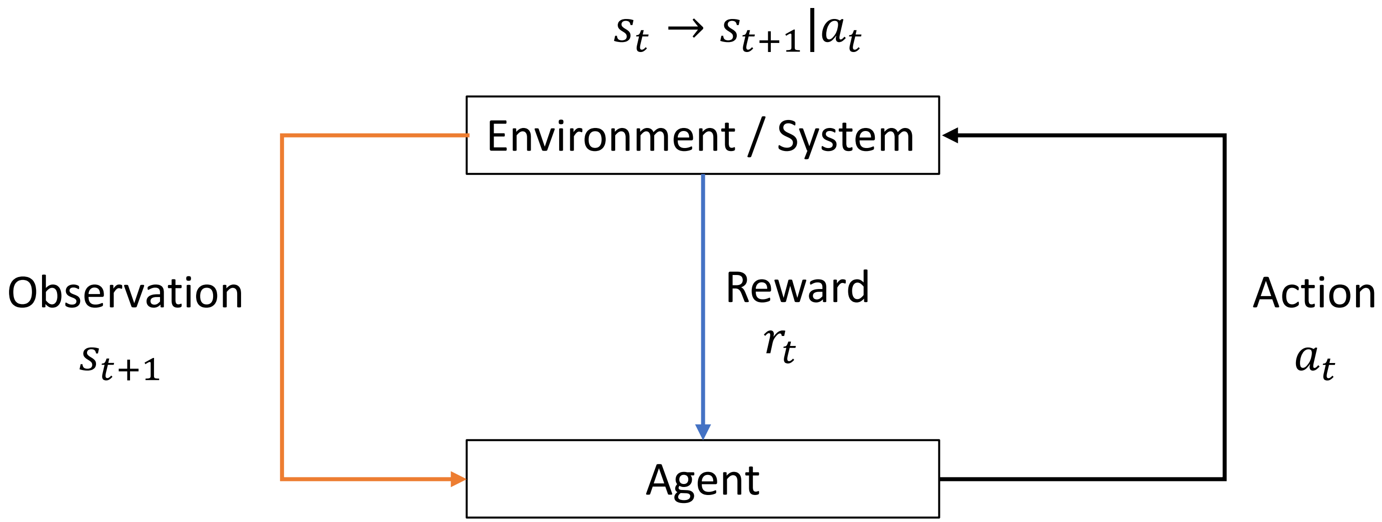
Introduction



- Planning based methods were used to solve game-theory based cyber-attack response problems.
- Planning requires explicitly constructing the models of the players.

R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, Second edition. Cambridge, Massachusetts: The MIT Press, 2018.

Introduction - Reinforcement Learning



THE OHIO STATE UNIVERSITY

R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, Second edition. Cambridge, Massachusetts: The MIT Press, 2018.

Elements of a Reinforcement Learning Problem

$$S, A, \pi\ , P, R, \boldsymbol{\gamma}$$

- S represents the set of all possible states of the system / environment.
- A is the action space of the agent - the set of possible actions of the agent.
- π - A mapping from S to the probabilities of taking different actions.
  - The manner in which an agent behaves is defined by the policy.
  - $\pi(a|s)$ is the probability of taking action a in state s.
- $P: S \times A \times S \rightarrow [0,1]$ is the state transition probability mapping. The agent observes the state of the environment s_t, implements the action $a_t \in A$ on the environment, and the environment transitions to a new state s_{t+1}, with a probability of transition $P(s_{t+1}|s_t, a_t)$
- $R: S \times A \times S \rightarrow \mathbb{R}$ is the reward function.
  - At any timestep t, if the environment is in state s_t, the agent takes action a_t and the environment transitions to state s_{t+1}, the agent receives an immediate reward $r_t = R(s_t, a_t, s_{t+1})$. The reward r_t is a real number.

The Ohio State University

R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, Second edition. Cambridge, Massachusetts: The MIT Press, 2018.

Elements of a Reinforcement Learning Problem

- $\gamma \in [0,1]$ is the discount factor that represents the weight assigned to future rewards.
- The discounted cumulative reward obtained by the agent over the course of time is:

$$G_t = \sum_{j=0}^{\infty} \gamma^j r_{t+j}$$

where r_{t+j} is the reward received j time steps after t.

- The agent's objective is to maximize the expected cumulative reward $\mathbb{E}_\pi[\sum_{j=0}^{\infty} \gamma^j r_{t+j}]$.

R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, Second edition. Cambridge, Massachusetts: The MIT Press, 2018.

Elements of a Reinforcement Learning Problem

- Value function represents the expected cumulative reward obtained starting from state s if the policy π is followed and t is any time step.

$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$$

- Q-value, is defined for every state-action pair (s, a) representing the expected cumulative reward if action a is taken at state s, and then the policy π is followed subsequently from any time step t.

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$$

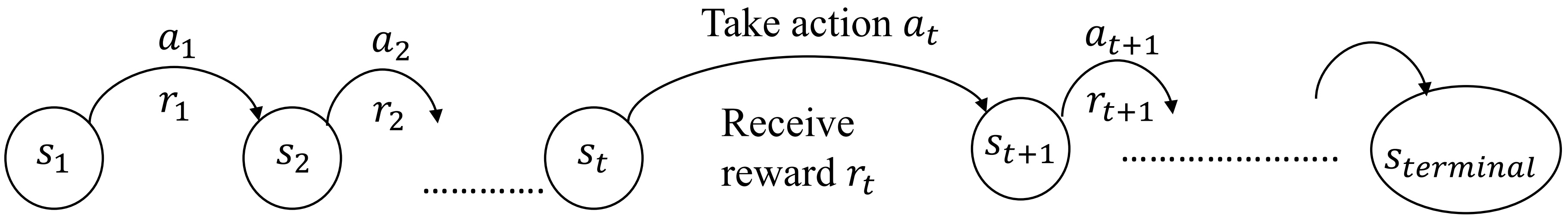
- Bellman equation for the Q-value function

$$Q(s_t, a_t) = \sum_{s_{t+1} \in S} \left(p(s_{t+1} | s_t, a_t) \times \left[R(s_t, a_t, s_{t+1}) + \gamma \times \sum_{a_{t+1} \in A} (\pi(a_{t+1} | s_{t+1}) \times Q(s_{t+1}, a_{t+1})) \right] \right)$$

The Ohio State University

R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, Second edition. Cambridge, Massachusetts: The MIT Press, 2018.

Q-Learning



- In an episode i, the agent's action a_t at system state s_t is chosen according to the Q-values learned by the agent up to the episode $i-1$.
- Greedy policy:

$$a_t = \arg\max_a Q_{i-1}(s_t, a)$$

- ε-greedy policy: A random action is chosen with a probability ε. Exploration - agent can explore actions that are different from those dictated by previous experience.
- Assume that the action a_{t+1} at state s_{t+1} is chosen such that, $Q_{i-1}(s_{t+1}, a_{t+1})$ is maximum
- Q-update equation based only on current sample of $(s_t, a_t, s_{t+1}, a_{t+1})$ and $Q_{i-1}(s_t, a_t)$:

$$Q_i(s_t, a_t) \leftarrow R(s_t, a_t, s_{t+1}) + \gamma \max_a Q_{i-1}(s_{t+1}, a)$$

- Q-value update equation, which combines the Q-values learned previously, with the current updates using a learning parameter $\alpha \in (0,1]$.

$$Q_i(s_t, a_t) \leftarrow (1-\alpha) \times Q_{i-1}(s_t, a_t) + \alpha \times [R(s_t, a_t, s_{t+1}) + \gamma \max_a Q_{i-1}(s_{t+1}, a)]$$

R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, Second edition. Cambridge, Massachusetts: The MIT Press, 2018.

Elements of Multi-Agent RL

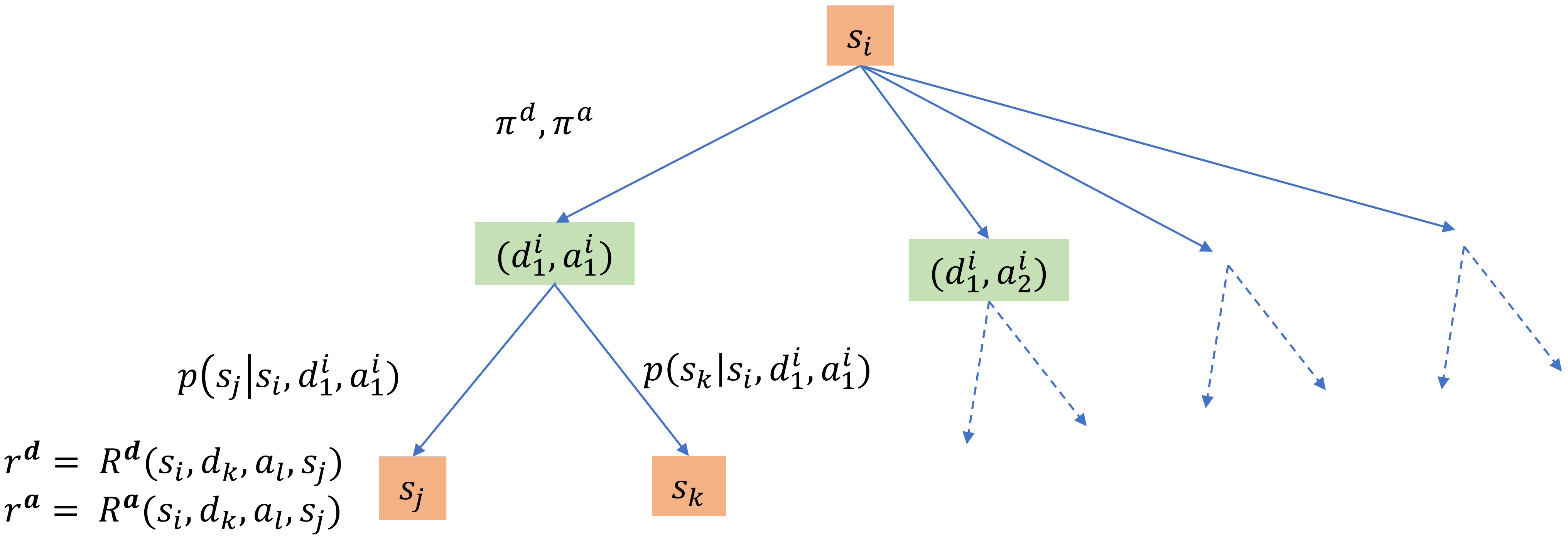
- Cyber-attacks, involve multiple agents acting on the environment simultaneously and trying to maximize their individual rewards.
- The system is affected by the actions of all the agents.
- The corresponding reward received by every individual agent is dependent on actions of all other agents.
- Markov game framework – Two players: attacker and defender.

$$S, \{A, D\}, \left\{\pi^{a}, \pi^{d}\right\}, P, \left\{R^{a}, R^{d}\right\}, \gamma$$

- $D = \{d_1, d_2, d_3 \ldots\}$ is the defender's action space and $A = \{a_1, a_2, a_3 \ldots\}$ is the attacker's action space.
- $\pi^{\boldsymbol{d}}$ and $\pi^{\boldsymbol{a}}$ are the action policies of the defender and the attacker.
- $P: S \times D \times A \times S \rightarrow [0, 1]$ is the state transition probability mapping.
- $R^{\boldsymbol{d}}: S \times D \times A \times S \rightarrow \mathbb{R}$ is the reward function of the defender.
- $R^{\boldsymbol{a}}: S \times D \times A \times S \rightarrow \mathbb{R}$ is the attacker's reward function.

The Ohio State University

Elements of Multi-agent RL



- $Q^{\boldsymbol{d}}$ and $Q^{\boldsymbol{a}}$ the action-value functions of the defender the attacker.
- functions of defender and attacker action pairs.

Elements of Multi-agent RL

		Q^d – defender Q-values				Q^a – Attacker Q-values		
Attacker Actions		1	2	3		1	2	3
Defender Actions	1	3.72	-4.27	3.5		-5.75	-4.8	-3.67
	2	-7.5	-2.25	-2.75		4.52	-3.61	-2.50
	3	-2.94	-7.6	1.67		-3	-2.54	4.57

Q-update equations:

$$Q_i^d(s_t, d_t, a_t) = (1-\alpha) \times Q_{i-1}^d(s_t, d_t, a_t) + \alpha \times \left[r_t^d + \gamma \times Optimal\left(Q_{i-1}^d(s_{t+1}, d_{t+1}, a_{t+1})\right)\right]$$

$$Q_i^a(s_t, d_t, a_t) = (1-\alpha) \times Q_{i-1}^a(s_t, d_t, a_t) + \alpha \times [r_t^a + \gamma \times Optimal\,(Q_{i-1}^a(s_{t+1}, d_{t+1}, a_{t+1}))]$$

How to choose optimal actions? – Game theory*

Stackelberg Equilibrium

- In a two player Stackelberg game one of the players acts as the leader and the other is a follower.
- Used in security games – with the defender as the leader and the attacker as the follower.
- Leader: can enforce their strategy (action).
- Follower: responds to leader's strategy in a rational manner, i.e., in a manner that optimizes their reward.
- The procedure to calculate Stackelberg equilibrium involves a two-step backward calculation.
- In the first step the follower's optimal response to every one of leader's actions is identified.
- In the second step the leader's action that generates the optimal reward given that the follower responds with the actions identified in the first step is obtained.
- The leader needs to know all Q-functions, while it is sufficient for the follower to know just their Q-functions.

Stackelberg Equilibrium

Step – 1:

- Identify the attacker's (follower's) action that generates the maximum reward (in this case Q-value) for every possible defender action.

$$a_S(d_i) = \underset{a, d_i \in D}{\arg\max}\, Q^a(s, d_i, a)$$

where D is the defender's (leader's) action space,
$d_i \in D$ is the defender's (leader's) action, and
$a_S(d_i)$ is the optimal response by the attacker (follower) for defender's (leader's) action d_i.

- $a_S(d = 1) = 3$, $a_S(d = 2) = 1$ and $a_S(d = 3) = 3$

		Q^d – defender Q-values				Q^a – Attacker Q-values		
Attacker Actions		1	2	3		1	2	3
Defender Actions	1	3.72	-4.27	3.5		-5.75	-4.8	-3.67
	2	-7.5	-2.25	-2.75		4.52	-3.61	-2.50
	3	-2.94	-7.6	1.67		-3	-2.54	4.57

Stackelberg Equilibrium

Step – 2:

- Identify the defender's (leader's) action that generates the maximum reward, for the attacker's (follower's) actions calculated in step – 1.

$$d_S = \underset{d \in D}{\arg\max}\, Q^d(s, d, a_S(d))$$

where D is the defender's (leader's) action space,

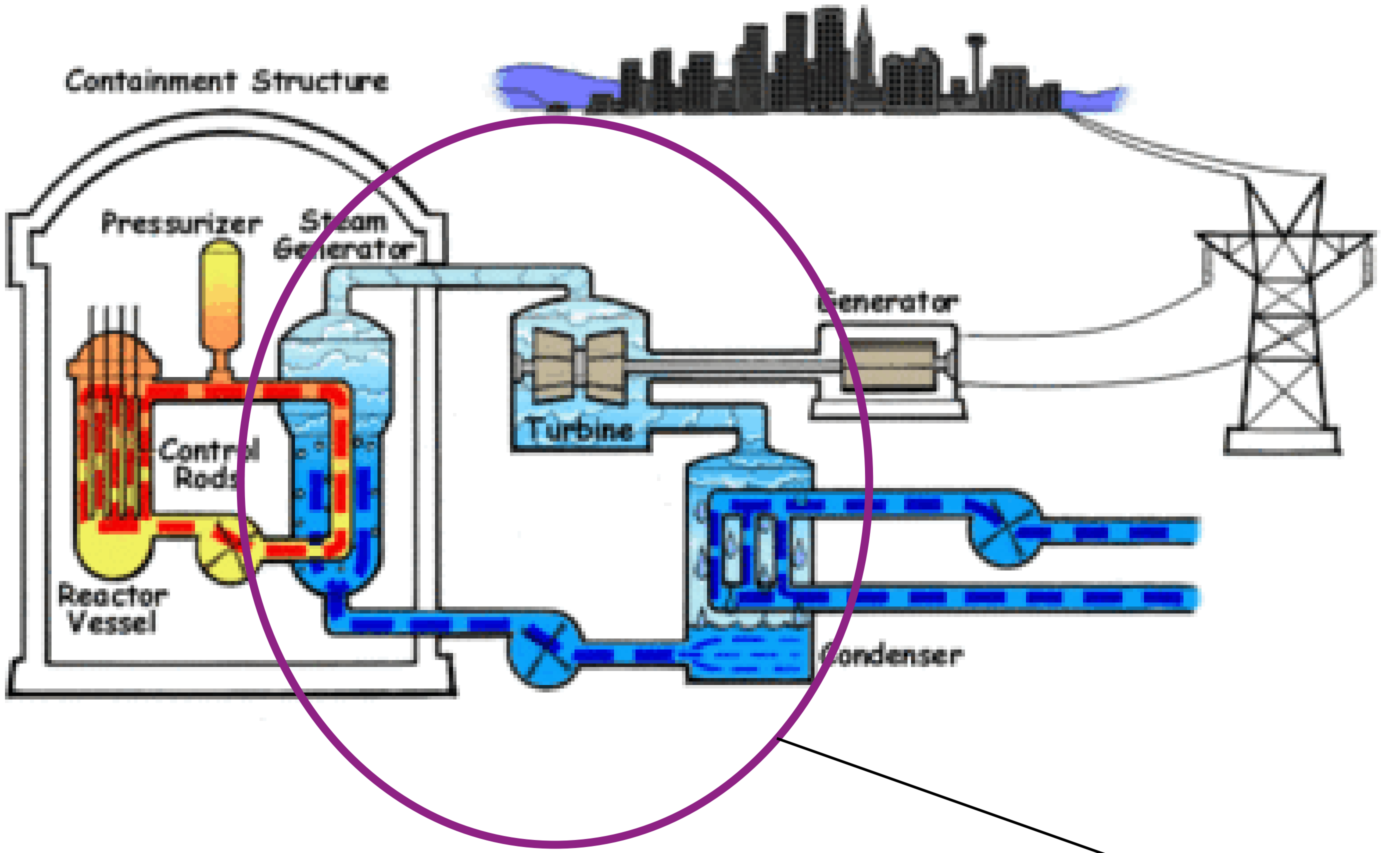
$a_S(d)$ is the optimal response by the attacker (follower) for defender action d, and

d_S is the optimal defender action.

- $(d_S, a_S(d_S)) = (1,3)$ is the pure strategy Stackelberg equilibrium.

		Q^d – defender Q-values				Q^a – Attacker Q-values		
Attacker Actions		1	2	3		1	2	3
Defender Actions	1	3.72	-4.27	3.5		-5.75	-4.8	-3.67
	2	-7.5	-2.25	-2.75		4.52	-3.61	-2.50
	3	-2.94	-7.6	1.67		-3	-2.54	4.57

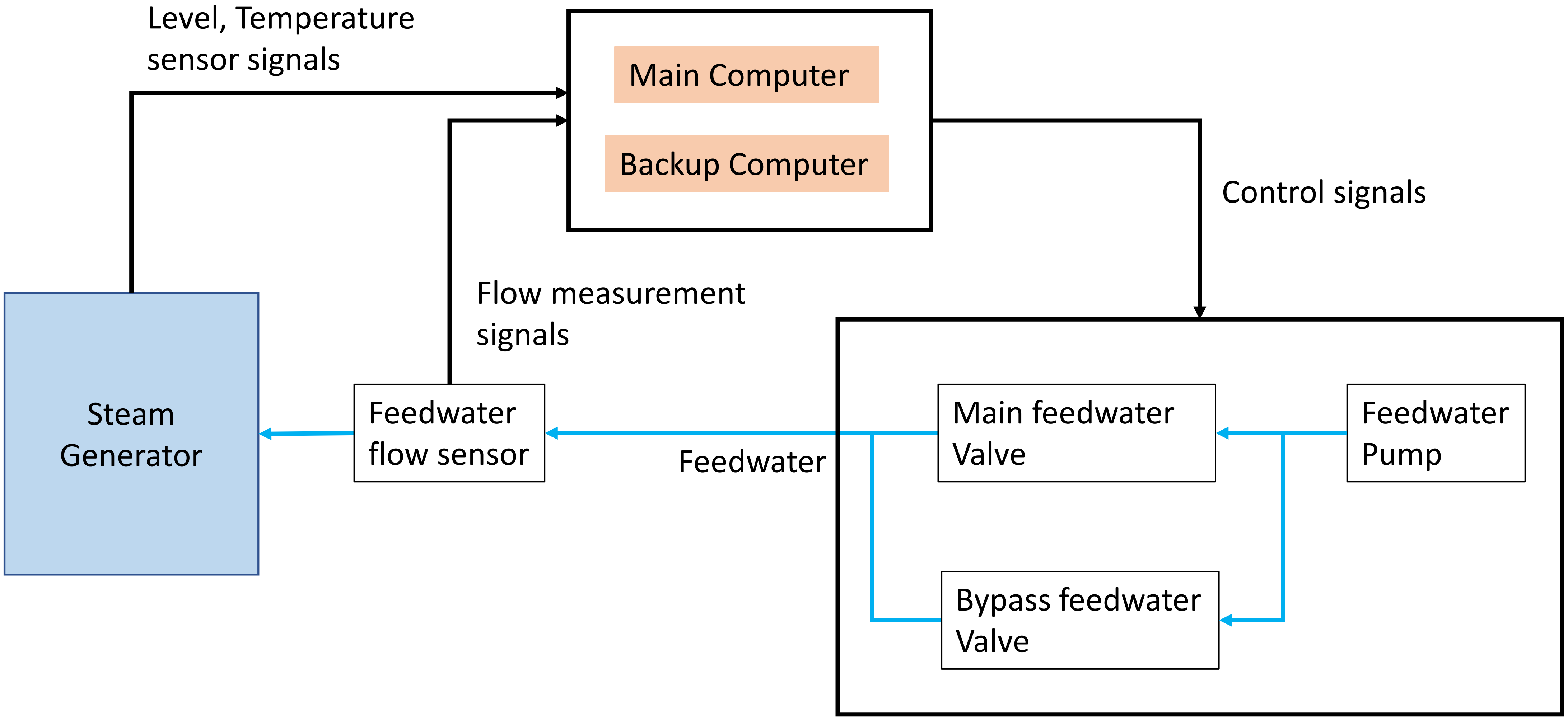
Case Study – PWR



Feedwater system

Source: https://www.nrc.gov/reading-rm/basic-ref/students/animated-pwr.html

Digital Feedwater Control System (DFWCS)



T. Aldemir *et al.*, "NUREG/CR-6942: Dynamic Reliability Modeling of Digital Instrumentation and Control Systems for Nuclear Reactor Probabilistic Risk Assessments," 2007.
Zhao, Y., Huang, L., Smidts, C. and Zhu, Q., 2020. Finite-horizon semi-Markov game for time-sensitive attack response and probabilistic risk assessment in nuclear power plants. *Reliability Engineering & System Safety*, *201*, p.106878.

System Component states and Modes

Digital Components:
1. Sensors
2. Main Computer (MC)
3. Backup Computer (BC)

→ States:
1. Normal and in Use.
2. Normal and in standby (not used)
3. Compromised and in Use
4. Compromised and not used.

4. Control Mode → States:
1. Automatic
2. Manual

5. Sensing Mode → States:
1. Sensors are used.
2. Approximate model is used.

6. Reactor Core → States:
1. Normal
2. Damaged

Zhao, Y., Huang, L., Smidts, C. and Zhu, Q., 2020. Finite-horizon semi-Markov game for time-sensitive attack response and probabilistic risk assessment in nuclear power plants. *Reliability Engineering & System Safety*, *201*, p.106878.

Attacker and Defender actions

Attacker Actions:

1. Compromise the Sensors
2. Compromise the Main Computer (MC)
3. Compromise the Backup Computer (BC)
4. Do nothing.

Defender Actions:

1. Switch from the sensors to using approximate model.
2. Switch control from MC to BC.
3. Switch control from BC to manual control.
4. Do nothing.

Zhao, Y., Huang, L., Smidts, C. and Zhu, Q., 2020. Finite-horizon semi-Markov game for time-sensitive attack response and probabilistic risk assessment in nuclear power plants. *Reliability Engineering & System Safety*, *201*, p.106878.

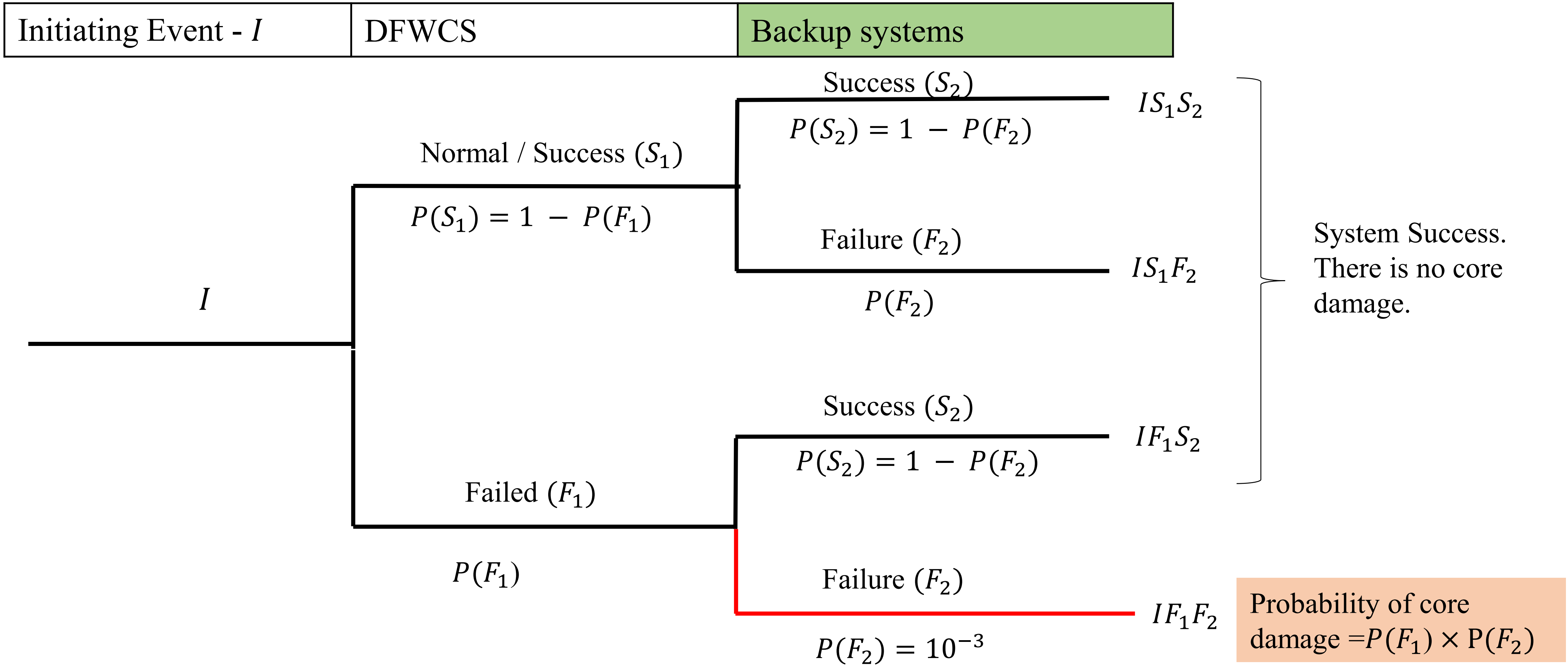
Physical system states

State	Vector	Description	SAFE or not
1	[1 1 2 1 1 1]	Auto with normal MC and normal sensors	SAFE
2	[3 1 2 1 1 1]	Auto with normal MC, compromised sensors	UNSAFE
3	[1 3 2 1 1 1]	Auto with compromised MC, normal sensors	UNSAFE
4	[1 NU 1 1 1 1]	Auto with normal BC, and normal sensors	SAFE
5	[3 3 2 1 1 1]	Auto with compromised MC and sensors	UNSAFE
6	[3 NU 1 1 1 1]	Auto with normal BC and compromised sensors	UNSAFE
7	[NU 1 2 1 2 1]	Auto with normal MC and approximate model	SAFE
8	[1 NU 3 1 1 1]	Auto with compromised BC and normal sensors	UNSAFE
9	[NU 3 2 1 2 1]	Auto with compromised MC and approximate model	UNSAFE
10	[3 NU 3 1 1 1]	Auto with compromised BC and sensors	UNSAFE
11	[1 NU NU 2 1 1]	Manual with normal sensors	SAFE
12	[NU NU 1 1 2 1]	Auto with normal BC and approximate model	SAFE
13	[3 NU NU 2 1 1]	Manual with compromised sensors	UNSAFE
14	[NU NU 3 1 2 1]	Auto with compromised BC and approximate model	UNSAFE
15	[NU NU NU 2 2 1]	Manual with approximate model.	SAFE
16	[X X X X X 2]	Core damaged - END	UNSAFE (end)

NU – Not in use. X –Of no consequence.

Attacker and Defender Action Space

	Physical System State	Description	Attacker actions	Defender actions
1	[1 1 2 1 1 1]	Auto with normal MC and normal sensors	1, 2, 4	1, 2, 4
2	[3 1 2 1 1 1]	Auto with normal MC, compromised sensors	2, 4	1, 2, 4
3	[1 3 2 1 1 1]	Auto with compromised MC, normal sensors	1, 4	1, 2, 4
4	[1 NU 1 1 1 1]	Auto with normal BC, and normal sensors	1, 3, 4	1, 3, 4
5	[3 3 2 1 1 1]	Auto with compromised MC and sensors	4	1, 2, 4
6	[3 NU 1 1 1 1]	Auto with normal BC and compromised sensors	3, 4	1, 3, 4
7	[NU 1 2 1 2 1]	Auto with normal MC and approximate model	2, 4	2, 4
8	[1 NU 3 1 1 1]	Auto with compromised BC and normal sensors	1, 4	1, 3, 4
9	[NU 3 2 1 2 1]	Auto with compromised MC and approximate model	4	2, 4
10	[3 NU 3 1 1 1]	Auto with compromised BC and sensors	4	1, 3, 4
11	[1 NU NU 2 1 1]	Manual with normal sensors	1, 4	1, 4
12	[NU NU 1 1 2 1]	Auto with normal BC and approximate model	3, 4	3, 4
13	[3 NU NU 2 1 1]	Manual with compromised sensors	4	1, 4
14	[NU NU 3 1 2 1]	Auto with compromised BC and approximate model	4	3, 4
15	[NU NU NU 2 2 1]	Manual with approximate model.	4	4
16	[DM DM DM DM DM 2]	Core damaged - END	4	4



If the attack succeeds i.e., if the system is in an unsafe state, Probability of core damage = $P(F_2) = 10^{-3}$

If the system is in a completely safe state, Probability of core damage = $P(F_1) \times P(F_2) = 10^{-5}$

Transition probabilities to terminal state

States	Probability of transition to core damage state.
16 – terminal.	(Already in core damage state)
5, 10 – Both controller and sensors are compromised and in use. 2, 3, 6, 8, 9, 13, 14 – one component is compromised.	10^{-3}
1 – initial state.	10^{-5}
4 – Automatic mode with normal BC and sensors. 7 – Automatic mode with normal MC and approximate model.	3.34×10^{-5}
11 – Manual control with normal sensors. 12 – Automatic mode with normal BC and approximate model.	6.67×10^{-5}
15 - Manual mode with approximate model	10^{-4}

Rewards functions

$$r = r_{action} + r_{transition}$$

r_{action} = Cost of taking an action.

- Cost of taking any action i.e., any one of actions 1, 2 and 3 is $ 10,000 for the attacker
- The defender incurs no cost to take any action .

$r_{transition}$ $(r^{d} = R^{d}(s_i, d_k, a_l, s_j);\ r^{a} = R^{a}(s_i, d_k, a_l, s_j))$

- the attacker receives an immediate positive reward of $10,000 when they compromise a single component (when there is a transition to states 2, 3, 6, 8, 9, 13 and 14) due to their actions.
- Similarly, a positive reward of $20,000 when there is a transition to the states 5 and 10 - states in which two components are compromised.
- The defender receives equivalent negative rewards.

THE OHIO STATE UNIVERSITY

Zhao, Y., Huang, L., Smidts, C. and Zhu, Q., 2020. Finite-horizon semi-Markov game for time-sensitive attack response and probabilistic risk assessment in nuclear power plants. *Reliability Engineering & System Safety*, *201*, p.106878.

Results

	Defender as the Leader		Attacker as the leader	
State	Defender's action	Attacker's action	Defender's action	Attacker's action
1	1	2	2	2
2	1	2	2	2
3	2	1	2	1
4	3	1	3	3
5	1	4	2	4
6	1	4	1	3
7	4	2	2	4
8	3	1	1	1
9	2	4	2	4
10	3	4	1	4
11	4	1	4	1
12	3	4	3	4
13	1	4	1	4
14	3	4	3	4
15	4	4	4	4
16	4	4	4	4

Discussion

- As the leader, the defender is initially prioritizing the use of approximate model which cannot be subjected to cyber-attacks.
- It is not possible to compromise both the main computer and the backup computer at the same time. So, when the defender is switching to the approximate model, it is impossible to reach the states in which two components are compromised at the same time.
- When the attacker is the leader, the priority is on compromising the main computer initially, which will lead the defender to switch to backup computer, thereby providing the attacker with additional opportunities to compromise multiple components.

Conclusions and Future Work

- We presented the use of a multi-agent reinforcement learning approach, specifically the multi-agent q-learning algorithm with Stackelberg equilibrium to compute the defender's optimal response strategy against cyber-attacks.
- We assumed that the leader is aware of the follower's rewards. It is also assumed that the attacker (follower) can always observe the strategy enforced by the defender (leader). Future work will be focused towards relaxing these assumptions.

Acknowledgements

This research is being performed using funding received from the DOE Office of Nuclear Energy's Nuclear Energy University Program.

Thank You

Stackelberg Equilibrium

- For the defender (leader) to compute and enforce their strategy, they should have knowledge of the attacker's (follower's) Q-values to estimate the attacker's optimal response for every one of their actions.

- This is the result of the assumption that the defender (leader) is aware of the attacker's (follower's) rewards.

- It is also assumed that the attacker (follower) can always observe the strategy enforced by the defender (leader).

- It is realistic to expect that the defender is unaware of the attacker's rewards and attacker (follower) cannot completely observe the defender's strategy.