Stochastic Bayesian Games for the Cybersecurity of Nuclear Power Plants

by

Lee Tylor Maccarone

B.S. Mechanical Engineering, University of Pittsburgh, 2016

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2021

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Lee Tylor Maccarone

It was defended on

June 3, 2021

and approved by

Daniel G. Cole, Ph.D., Associate Professor,

Department of Mechanical Engineering and Materials Science

Mai Abdelhakim, Ph.D., Assistant Professor,

Department of Electrical and Computer Engineering

Nikhil Bajaj, Ph.D., Assistant Professor,

Department of Mechanical Engineering and Materials Science

William W. Clark, Ph.D., Professor,

Department of Mechanical Engineering and Materials Science

Christopher C. Lamb, Ph.D., Principal Cyber-Security Research Scientist,

Sandia National Laboratories

Dissertation Director: Daniel G. Cole, Ph.D., Associate Professor,

Department of Mechanical Engineering and Materials Science

Stochastic Bayesian Games for the Cybersecurity of Nuclear Power Plants

Lee Tylor Maccarone, PhD

University of Pittsburgh, 2021

The goal of this research is to reduce the likelihood of successful attacks on nuclear power plants. Cyber-physical systems such as nuclear power plants consist of interconnected physical processes and computational resources. Because the cyber and physical worlds are integrated, vulnerabilities in both the cyber and physical domains can result in physical damage to the system. Nuclear power plants can be targeted by a variety of adversaries — each with a unique motivation and set of resources. To secure nuclear power plants and other cyber-physical systems, we require an approach to security that also accounts for the interactions of human decision-makers.

This research uses a game-theoretic approach to nuclear cybersecurity. The cybersecurity of the plant can be viewed as a non-cooperative game between a defender and an attacker. The field of game theory provides a mathematical framework to analyze the interactions of the defender and attacker as both players seek to accomplish their objectives. In this research, a stochastic Bayesian game is used to optimize cybersecurity decision-making. A stochastic Bayesian game is a combination of a stochastic game and a Bayesian game. The stochastic elements of the game enable the consideration of uncertainty in the interactions of the attacker and defender. The Bayesian elements of the game enable the consideration of the uncertainty regarding the attacker's characteristics. This combination is useful for the analysis of nuclear power plant cybersecurity because it enables plant defenders to optimize their security decisions in the presence of uncertainty.

Table of Contents

Preface . . . xiv

1.0 Introduction . . . 1

1.1 Research Objectives . . . 2

1.2 Research Approach . . . 3

1.3 Contributions . . . 4

1.4 Broader Impact . . . 5

1.5 Dissertation Overview . . . 6

2.0 State of the Art and Limits of Current Practice . . . 8

2.1 ICS Cybersecurity Methods . . . 8

2.1.1 Expert-Elicited Models . . . 8

2.1.2 Attack Graphs . . . 10

2.1.3 Petri Nets . . . 12

2.2 Game-Theoretic Approaches to Cybersecurity . . . 14

2.2.1 Strategic Form Games . . . 14

2.2.2 Stochastic Games . . . 15

2.2.3 Bayesian Games . . . 16

2.2.4 Limits of Current Practice . . . 17

2.3 Summary of Limits of Current Practice . . . 19

3.0 Stochastic Bayesian Games . . . 20

3.1 Preliminaries . . . 20

3.2 Bayesian Games . . . 23

3.2.1 Bayesian Games in the Extensive Form . . . 25

3.2.2 Bayesian Nash Equilibria . . . 27

3.2.3 Solution Methods . . . 29

3.3 Stochastic Games . . . 29

3.3.1 Nash Equilibrium . . . 31

3.4 Stochastic Bayesian Games . . . 33
3.4.1 Bayesian Learning of the Adversary's Parameters . . . 33
3.4.2 Harsanyi-Bellman Ad Hoc Coordination . . . 35
3.4.3 Application of SBGs to Cybersecurity Decisions . . . 37
4.0 Construction of the Stochastic Bayesian Game . . . 40
4.1 The Residual Heat Removal System . . . 41
4.2 The Players . . . 46
4.2.1 The Defender . . . 46
4.2.2 The Attacker . . . 52
4.2.2.1 Threat Agent Library . . . 52
4.2.2.2 Threat Agent Risk Assessment . . . 55
4.2.3 The Type Distribution . . . 65
4.3 Stochastic State Space . . . 66
4.3.1 System-Theoretic Process Analysis . . . 67
4.4 The Actions . . . 70
4.4.1 Normal and Penetrated States . . . 70
4.4.2 Hazard States . . . 74
4.4.3 Loss States . . . 75
4.5 State Transitions . . . 75
4.5.1 Penetration Transitions . . . 75
4.5.2 Hazard Transitions . . . 78
4.5.3 Loss Transitions . . . 78
4.5.4 Recovery Transitions . . . 79
4.6 Utility Functions . . . 81
4.6.1 Immediate Utility Functions . . . 82
4.6.2 Cumulative Utility Functions . . . 84
4.7 Decision Algorithms . . . 85
4.7.1 The Defender . . . 85
4.7.1.1 Bayesian Learning of Attacker's Parameters . . . 85
4.7.1.2 Estimating the Attacker's Type . . . 87

4.7.1.3 HBA Implementation . . . 88
4.7.2 The Attacker . . . 91
5.0 Simulation Examples . . . 98
5.1 Radical Activist Simulation . . . 98
5.2 Disgruntled Employee Simulation . . . 100
5.3 Government Cyberwarrior Simulation . . . 105
5.4 Terrorist Simulation . . . 111
6.0 Results and Discussion . . . 117
6.1 Security Metrics . . . 117
6.1.1 Mean Time-to-loss . . . 117
6.1.2 Mean Availability . . . 118
6.1.3 The Defender's Cumulative Utility . . . 119
6.1.4 The Attacker's Cumulative Utility . . . 119
6.2 Bayesian Learning of the Attacker's Loss Utility . . . 123
6.3 Estimating the Attacker's Type . . . 125
7.0 Conclusions and Future Work . . . 128
7.1 Summary of Contributions . . . 129
7.2 Limitations . . . 129
7.3 Future Work . . . 130
Appendix A. Observability Attacks and Game Theory . . . 132
A.1 Attacker Controllability and Observability . . . 132
A.2 Stealthy Observability Attacks . . . 135
A.3 Game-Theoretic Approach . . . 137
A.3.1 Game Overview . . . 140
A.3.2 The Defender . . . 141
A.3.3 The Attacker . . . 142
A.4 Balance of Plant Model . . . 143
A.4.1 U-Tube Steam Generator . . . 145
A.4.2 Steam Turbine . . . 145
A.4.3 Condenser . . . 146

A.5 Results and Discussion . . . 146
A.5.1 Iterated Elimination of Dominated Strategies . . . 146
A.5.2 Pure-Strategy Nash Equilibria . . . 149
A.5.2.1 No reinforcement or masking . . . 150
A.5.2.2 No reinforcement and one masking . . . 150
A.5.2.3 No reinforcement and two maskings . . . 151
A.5.2.4 Impossible pure-strategy equilibria . . . 151
A.5.3 Mixed-Strategy Nash Equilibria . . . 152
A.5.3.1 One masking and two maskings . . . 152
A.5.3.2 Zero maskings and one masking . . . 153
A.5.3.3 Zero masking and two maskings . . . 153
A.6 Summary and Conclusions . . . 154
Appendix B. Bayesian Game Examples . . . 156
B.1 Bayesian Game Theory . . . 156
B.1.1 Stackelberg Games . . . 156
B.1.2 Decomposed Optimal Bayesian Stackelberg Solver . . . 157
B.2 Bayesian Game Construction . . . 159
B.2.1 The Defender's Types . . . 159
B.2.2 Type Distributions . . . 162
B.3 The Players' Actions . . . 163
B.3.1 Action Profiles and Consequences . . . 165
B.3.2 The Players' Utility Functions . . . 166
B.4 Results and Discussion . . . 170
B.4.1 Stackelberg Game . . . 170
B.4.2 Simultaneous Game with One Defender Type . . . 171
B.4.3 Simultaneous Game with Four Defender Types . . . 173
B.5 Summary and Conclusions . . . 174
Bibliography . . . 177

List of Tables

1 Prisoners' dilemma in the strategic form. 21

2 Match pennies in the strategic form. 23

3 A Bayesian security game. For each strategy intersection, the defender's payoff and attacker's payoff are provided. 24

4 An example of the Harsanyi transformation. 38

5 U.S. adults' support for constructing nuclear power plants in their area, before and after the Three Mile Island accident in March 1979 [72]. 50

6 Losses and their assigned consequence magnitudes. 51

7 Hostile agents from Intel Corporation's Threat Agent Library [39] 56

8 Threat agents, their motivations, limits, desired outcomes, and desired NPP losses. 59

9 Threat agents, their access, resources, skill levels, visibility, estimated loss likelihoods, and risk. 60

10 The attacker's types and corresponding loss magnitudes. 64

11 The probability distributions of the types. 66

12 Hacked devices and potential hazards. 68

13 States in the SBG. 71

14 The cybersecurity choices available to the defender in the normal and penetrated states. 72

15 The cybersecurity choices available to the attacker in the normal and penetrated states. 72

16 The success rates of the cybersecurity choices available to the defender in the normal and penetrated states. 76

17 The success rates of the cybersecurity choices available to the attacker in the normal and penetrated states. 77

18 The success rates of the attacker's hazard initiation. 79

19 The probability of hazard states transitioning to loss states, given that the attacker has chosen loss initiation and the defender's hazard recovery is unsuccessful. 81
20 Utility ($) given to each player when a device is penetrated. 83
21 Utility ($) given to each player when a hazard occurs. 84
22 Initial loss estimates for each attacker type. 87
23 Parameters of the utility adjustment factor used to evaluate paths in HBA. . 91
24 Type parameters used in the attacker's decision algorithm. 92
25 An example of the observability attack game. The utilities for the attacker and defender are provided for each intersection of defender and attacker strategies. 139
26 The strategies of the attacker and defender. The defender's strategy s_D^i includes a set of reinforced signals of quantity n_D. The attacker's strategy s_A^j includes a set of masked signals of quantity n_A. 147
27 Stealth outcomes of the observability attack game: (O) observable attack; (U) unstealthy attack; (S) stealthy attack. 148
28 Normal form of the reduced observability attack game. The defender's utility is listed in the first row of each cell, followed by the attacker's utility. 149
29 The defender's losses and their possible values. 160
30 The defender's types and corresponding loss magnitudes. 161
31 The probability distributions of the types. 162
32 The choices available to each player. 164
33 The probability of hazards causing losses. 167
34 The expense parameters for each player in the Bayesian game. 169
35 Nash equilibrium of the Bayesian Stackelberg game. 170
36 Nash equilibrium of the Bayesian simultaneous game with one defender type. 172
37 Nash equilibrium of the Bayesian simultaneous game with four defender types. 175

List of Figures

1 An example of an attack graph. 11

2 An example of a Petri net [19]. 13

3 The extensive form of a Bayesian simultaneous game. 26

4 A simple stochastic security game. 38

5 Approximate Bayesian updating example. 39

6 Boiling water reactor (BWR) overview. 42

7 Residual heat removal (RHR) system. 43

8 Network topology of the RHR system. 45

9 U.S. nuclear industry safety accident rate [64]. The industry safety accident rate is the number of accidents resulting in lost work, restricted work, or fatalities per 200,000 worker hours. 49

10 Intel Corporation's Threat Agent Risk Assessment [40]. 57

11 Threat agent risks on logarithmic scale. 62

12 Generalized state space of an SBG. 69

13 Event tree mapping a hazard to a loss. 80

14 Flow chart of SBG simulation. 86

15 Sampling the upper and lower bounds of the paths provides a more consistent approximation of the average path utility than performing a purely stochastic sample. 89

16 State trajectory of the example game played against θ_A^1. 99

17 Average occurrences of each state when HBA is used against each θ_A^1. 99

18 Players' utilities from the example game played against θ_A^1. 100

19 The defender's belief distributions over the range of the attacker's possible utilities assigned to L_2 from the example game played against θ_A^1. 101

20 The defender's estimate of the attacker's utility of L_2 from the example game played against θ_A^1. 102

21 The defender's beliefs regarding the attacker's type from example game played against θ_A^1. 102
22 State trajectory of the example game played against θ_A^2. 103
23 Average occurrences of each state when HBA is used against each θ_A^2. 104
24 Players' utilities from the example game played against θ_A^2. 104
25 The defender's belief distributions over the range of the attacker's possible utilities assigned to L_2 from the example game played against θ_A^2. 106
26 The defender's estimate of the attacker's utility of L_2 from the example game played against θ_A^2. 107
27 The defender's beliefs regarding the attacker's type from example game played against θ_A^2. 107
28 State trajectory of the example game played against θ_A^3. 108
29 Average occurrences of each state when HBA is used against each θ_A^3. 108
30 Players' utilities from the example game played against θ_A^3. 109
31 The defender's belief distributions over the range of the attacker's possible utilities assigned to L_2 from the example game played against θ_A^3. 109
32 The defender's estimate of the attacker's utility of L_2 from the example game played against θ_A^3. 110
33 The defender's beliefs regarding the attacker's type from example game played against θ_A^3. 111
34 State trajectory of the example game played against θ_A^4. 112
35 Average occurrences of each state when HBA is used against each θ_A^4. 113
36 Players' utilities from the example game played against θ_A^4. 113
37 The defender's belief distributions over the range of the attacker's possible utilities assigned to L_2 from the example game played against θ_A^4. 114
38 The defender's estimate of the attacker's utility of L_2 from the example game played against θ_A^4. 115
39 The defender's beliefs regarding the attacker's type from example game played against θ_A^4. 116
40 SBG duration from using HBA against each attacker type. 118

41 Relative time spent in each plant condition when HBA was used against each attacker type. 120
42 The defender's cumulative utility when HBA is used against each attacker type. 121
43 The percentage of simulations where the defender's cumulative utility is positive when HBA is used against each attacker type. 121
44 The attacker's cumulative utility when the defender uses HBA against each attacker type. 122
45 The defender's final estimate of the attacker's loss utility when the defender uses HBA against each attacker type. 124
46 The expected value of each hazard as a function of the utility of L_2. The hazard preferences of θ_A^4 change as a function of L_2. 126
47 The defender's final estimate of the probability of the attacker's true type when the defender uses HBA against each attacker type. 127
48 The balance of plant system with global system inputs and measurements identified. 144
49 The extensive form of a Bayesian Stackelberg game. 157

Preface

I would like to thank my parents for their support and encouragement.

This research was supported in part by the University of Pittsburgh Center for Research Computing through the resources provided.

This material is based upon work supported under an Integrated University Program Graduate Fellowship. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the Department of Energy Office of Nuclear Energy.

This work was supported by Sandia National Laboratories Contract 2084579: Civilian Nuclear Power Plant Systems Game and Risk Modeling. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

1.0 Introduction

The goal of this research is to reduce the likelihood of successful attacks on nuclear power plants (NPPs). Cyber-physical systems (CPSs) such as NPPs consist of interconnected physical processes and computational resources [51]. Because the cyber and physical worlds are integrated, vulnerabilities in both the cyber and physical domains can result in physical damage to the system [79]. A cyber attack that has an adverse effect on physical processes is called a cyber-physical attack. NPPs are subject to federal cyber and physical security regulations, but are still susceptible to attack by various adversaries in an ever-evolving threat landscape. If this research is successful, we should be able to do the following: predict how adversaries might target an NPP, quantify NPP security, and optimally allocate security resources to defend the NPP.

There are many examples of NPPs being targeted by both internal and external threats. In 2003, the safety display of the Davis-Besse NPP was disabled by the Microsoft SQL Server worm [92]. This attack was not directed at the Davis-Besse plant, but was accidentally introduced to the system by a contractor's infected computer. In 2009, a former employee at Energy Future Holdings hacked the Comanche Peak NPP's energy forecast system through a VPN that was not deactivated after his employment was terminated [53]. Perhaps the most famous incident of a successful attack on a nuclear system is Stuxnet [25]. Stuxnet damaged Iranian centrifuges by masking system measurements while causing the centrifuges to operate erratically [48]. While this attack did not target a commercial NPP, it demonstrates the potentially severe consequences of a cyber-physical attack.

The Department of Homeland Security acknowledged that NPPs could be vulnerable to cyber-physical attacks, and has identified two cyber-physical security goals in the Nuclear Reactors, Materials, and Waste Sector-Specific Plan: (1) to reduce physical and cyber risks to nuclear sector assets and (2) to enable a risk-informed approach to security and resiliency enhancements [85]. This research addresses both of these goals by developing an approach to analyze the cyber-physical security of an NPP.

In addition, we must also consider the behavior of the humans who interact with the NPP. NPPs can be targeted by a variety of adversaries such as state agents, hacktivists, and disgruntled employees — each with a unique motivation and set of resources [60]. To secure NPPs and other CPSs, we require an approach to cyber-physical security that accounts for the interactions of human decision-makers. By considering the cyber-physical properties of an NPP and the behavior of the human decision-makers who interact with it, we can develop effective security strategies to protect the plant.

1.1 Research Objectives

We will reduce the likelihood of successful attacks on NPP by achieving the following research objectives:

1. **Predict how an adversary might target a nuclear power plant**

   First, we must be able to assess the impact of a given attack on the NPP. This requires an understanding of both the cyber and physical characteristics of the plant. An example of a cyber characteristic is the NPP's network structure, and an example of a physical characteristic is the dynamics of plant systems. Using this information, we can assess the potential impact of a given cyber-physical attack and determine if the attack could be attractive to an adversary.

   Second, we must be able to assess whether an adversary with knowledge of an attack's potential impact would choose to conduct the attack. A range of factors influence whether an adversary will attack a system, and how the adversary will conduct the attack. These factors include the cost of the attack, the benefit gained by the adversary if the attack is successful, and the penalty incurred by the adversary if the attack is unsuccessful. The adversary's decisions are also dependent on the costs, benefits, and penalties of those defending the plant. For example, an adversary may be more likely to attack a component if it is expensive for the defender to protect it. This interaction of adversary and defender cost parameters must be studied to predict how an adversary might target an NPP.

2. **Quantify nuclear power plant security**

   We will provide metrics that quantify the security level of the NPP. These metrics will describe the expected state of the NPP if a given security strategy is selected. For example, the operational capacity of the NPP can be described using metrics such as the mean time-to-failure and mean availability. The economics of the NPP can be described by considering the security costs and the financial consequences of a successful attack, among other factors. Using these metrics, we can analyze the efficacy of defense strategies with respect to the spectrum of attack strategies.

3. **Optimally allocate security resources to defend a nuclear power plant**

   With a quantitative understanding of the attacker's and defender's behavior, we can determine how to most effectively protect the NPP. Security strategies may include plant equipment upgrades such as redundant sensors or cybersecurity upgrades such as improved firmware scanning. Cyber and physical security must both be considered to defend a nuclear power plant from cyber-physical attacks.

By achieving these objectives, we have developed a technique to optimally defend an NPP given our knowledge of the potential attackers. This approach will enable security engineers to enhance the security of NPP with respect to the modern cyber-physical threat landscape. The methods developed in this research will be readily applicable to other critical infrastructure and cyber-physical systems.

1.2 Research Approach

The goal of this research is to reduce the likelihood of successful attacks on nuclear power plants. Cyber-physical systems such as nuclear power plants consist of interconnected physical processes and computational resources. Because the cyber and physical worlds are integrated, vulnerabilities in both the cyber and physical domains can result in physical damage to the system. Nuclear power plants can be targeted by a variety of adversaries

— each with a unique motivation and set of resources. To secure nuclear power plants and other cyber-physical systems, we require an approach to security that also accounts for the interactions of human decision-makers.

This dissertation presents a game-theoretic approach to nuclear cybersecurity. The cybersecurity of the plant can be viewed as a non-cooperative game between a defender and an attacker. The field of game theory provides a mathematical framework to analyze the interactions of the defender and attacker as both players seek to accomplish their objectives. The purpose of game theory is to identify the optimal strategy for each player.

The cyber-physical security of an NPP system is studied using a stochastic Bayesian game (SBG). An SBG is a combination of a stochastic game and a Bayesian game. Stochastic game theory enables us to analyze interactions between players where the outcome of the interactions is uncertain. Bayesian game theory enables us to consider uncertainties regarding the characteristics of the players. This combination of stochastic and Bayesian games is useful for the analysis of NPP cybersecurity because it enables plant defenders to optimize their security decisions in the presence of uncertainty.

The SBG is used to identify an optimal cybersecurity strategy by applying two techniques: Bayesian learning and Harsanyi-Bellman ad hoc coordination (HBA). Using Bayesian learning, the defender can use his observations of the attacker's actions and the game history to update his beliefs about the attacker's characteristics as the game is played. HBA can then be used by the defender to select a cybersecurity strategy in real-time. HBA combines game-theoretic equilibrium concepts with optimal control to identify the optimal strategy.

1.3 Contributions

Although game theory has been applied to address a variety of security challenges, little research has been done to apply game theory to the cybersecurity of nuclear power plants. Some security problems have been cast as stochastic games and as Bayesian games, but combined stochastic Bayesian games have not yet been used to address security challenges.

Stochastic Bayesian games have been used in the fields of artificial intelligence and multi-agent systems, but the application of these games for the cybersecurity of nuclear power plants is new.

This research provides new insight regarding the practical implementation of game theory for nuclear cybersecurity. Mathematicians have developed a significant body of literature surrounding game theory, but literature describing its implementation is scarce. This work demonstrates several techniques to bridge the gap between theory and practice. These techniques rely on tools familiar to cybersecurity experts, but their application to a game-theoretic approach is new.

The main contributions of this work to the field of NPP cybersecurity are:

1. an approach to characterize threats to NPPs and model them as attacker types in a Bayesian game
2. an approach to construct the state space of a stochastic security game
3. an approach to define the transition function of a stochastic security game
4. a novel application of stochastic Bayesian games to cybersecurity challenges
5. methods to approximate Harsanyi-Bellman ad hoc coordination solution methods for stochastic Bayesian games with large action spaces

1.4 Broader Impact

While this research is focused specifically on applications for commercial nuclear power plants, the approach can be generalized to defend other critical infrastructures. For example, the chemical sector consists of several hundred thousand chemical plants that convert raw materials to a variety of chemical products. Similar to commercial nuclear power plants, many chemical plants are CPSs that must operate within strict limitations to safely produce the desired chemical product. A cyber-physical attack on a chemical plant could result in faulty products, damaged machinery, environmental hazards, and unsafe conditions for plant employees and the surrounding public. Other critical infrastructure sectors such as critical

manufacturing, dams, communications, energy, and transportation are susceptible to similar operating restrictions and may suffer similar consequences if attacked by a threat agent [81].

This research is applicable to protect these critical infrastructures from cyber-physical attacks from the spectrum of threat agents. Using the proposed stochastic Bayesian game framework, defenders can identify optimal strategies to protect the system, even when faced with uncertainty about the threat agents. This will enable critical infrastructure sectors to implement defenses that provide adequate protection for the system, and to avoid overspending on superfluous security measures.

This research is also applicable to military systems. A 2013 task force report issued by the Defense Science Board emphasizes the severity of the cyber-threat to critical military and intelligence systems [20]. If the United States enters a conflict with a peer adversary, they could be susceptible to a variety of attacks disrupting communication systems, the supply chain, and offensive capabilities. Many of the postulated attacks could be cyber-physical with severe consequences.

By applying this research, military security analysts could effectively allocate finite security resources to ensure the greatest likelihood of mission success. The stochastic Bayesian game approach enables analysts to leverage intelligence gathered about the adversary, and to account for uncertainty in that intelligence. Resources can then be allocated to adequately defend mission-critical assets without overspending or interfering with the ability to complete the mission.

1.5 Dissertation Overview

This dissertation is structured as follows. Chapter 2 describes the state of the art and limits of current practice. Chapter 3 provides an overview of stochastic Bayesian games — the foundation of the research approach. Chapter 3 also describes how the defender can apply Bayesian techniques to learn about the attacker as the game is played and how the defender can select an optimal security strategy in real-time using Harsanyi-Bellman ad hoc coordination. Chapter 4 describes the construction of the development of our

case study, Chapter 5 provides examples of the game played against each attacker, and Chapter 6 provides the results and discussion. Finally, Chapter 7 concludes the dissertation and summarizes our research contributions.

2.0 State of the Art and Limits of Current Practice

This chapter presents the state of the art and limits of current practice of cybersecurity analysis techniques for industrial control systems (ICSs). First, we broadly discuss various approaches to cybersecurity for ICSs, and their limitations. Second, we discuss game-theoretic approaches to cybersecurity, and their limitations. We conclude with a summary of the key limitations that are addressed by this research.

2.1 ICS Cybersecurity Methods

The importance of securing CPS in critical infrastructure was identified by various researchers in the early 2000's [16, 90, 11], and was addressed by the U.S. Department of Energy (DOE) in 2002 [84]. As part of the President's Critical Infrastructure Board established in 2001, the DOE provided a list of recommendations for organizations to increase the security of supervisory control and data acquisition (SCADA) networks and establish an effective cyber security program. Unfortunately, it was not until the Stuxnet attack of 2010 that the security of CPS became more widely discussed [89, 76]. More recently, a ransomware attack forced the Colonial Pipeline to shut down for five days, thereby causing fuel shortages on the east coast of the United States [88]. This section presents several methods used to secure CPS.

2.1.1 Expert-Elicited Models

Expert-elicited models are computational models derived from expert characterization of the system [36]. Experts characterize the system by considering factors such as the importance of ICS components and functions, and how these resources could be targeted by

an attacker. These characterizations are quantified and used as inputs to a computational model. The computational model is then analyzed to identify effective security strategies to ensure mission success.

The expert-elicited approach consists of three general steps [36]. In the first step, the system is described by its function and components, and the connections between components identified. The vulnerabilities of components and the consequences of their compromise are also identified. In the second step, experts are consulted to estimate numerical parameters describing the system and its components, such as component importance and the consequence of the component being compromised. In the third step, the expert-elicited parameters are used by the mathematical model to characterize risk to the system. The complexity of the model can vary significantly.

The NIST Common Vulnerability Scoring System (CVSS) is an example of an expert-elicited model [28]. The CVSS is used to measure the severity of vulnerabilities in cyber-systems. To use the CVSS, experts first characterize the system using several categorical variables. These variables include descriptions of the exploitability of a vulnerability and the vulnerability's impact on confidentiality, integrity, and availability. Several equations are then used to calculate the vulnerability's score on a scale from zero to ten. A vulnerability with a greater score are considered to be more severe than one with a lower score. The CVSS exploitability parameters are discussed in greater detail in Section 4.5.1.

One limitation of expert-elicited models is the introduction of unintentional bias by the experts [36]. Experts often tend to describe the system in terms of its normal operation and intended uses. Resources are often prioritized by their importance during normal operations and resources that are rarely used are often neglected. Some of these limitations can be mitigated to some degree by using a large group of experts, or by conducting red-team exercises. Cybersecurity approaches for ICSs should be able to address security scenarios outside of normal operations.

2.1.2 Attack Graphs

Attack graphs are a modelling tool used to show the sequence of actions taken by an attacker to achieve a goal [36]. Attack graphs are generally used to assess the intrusion methods and path of attacker given the initial conditions of the system. Attack graphs can be used to gain insight about system features such as the network topology and access control mechanisms. The nodes in an attack graph represent system resources and privileges, and the edges represent the reachability to one resource from another [3].

An example of an attack graph for a network is shown in Figure 1. The attacker begins at the top, and is trying to reach the target at the bottom. The green triangle node represents the initial conditions of the attacker and the red octagon represents the attack target. Initial conditions of the network are represented by the blue rectangles. Based on the attacker's initial capabilities and the conditions of the network, the attacker can conduct exploit 1A or exploit 2 to attempt to reach the target. Exploit 2 leads directly to the target, but exploit 1A results in an intermediate condition that, when combined with another initial condition of the network, enables the attacker to launch exploit 1B to reach the target.

Many graph-based methods are deterministic and do not account for the difficulty of attack actions. These attack graphs provide insight regarding which states are theoretically reachable, but do not provide insight regarding the path of least resistance for the attacker. Consider the example in Figure 1. Although the right path requires fewer steps than left path, if exploit 2 is sufficiently difficult, the attacker might instead choose exploit 1A and 1B. Some graphs do account for path difficulty to provide probabilistic security assessments, such as those in [104, 66].

One limitation of attack graphs is the inability to model dynamic defense strategies. This means that the attacker proceeds through the attack graph given an initial defense strategy that does not change. This is a reasonable assumption for many passive cybersecurity actions such as access control, but does not accurately model active cybersecurity actions such as intrusion detection systems.

Another limitation of attack graphs is the inability to model cyclical behaviors. Attack graphs are generally directed acyclic graphs, meaning that each edge has a direction and

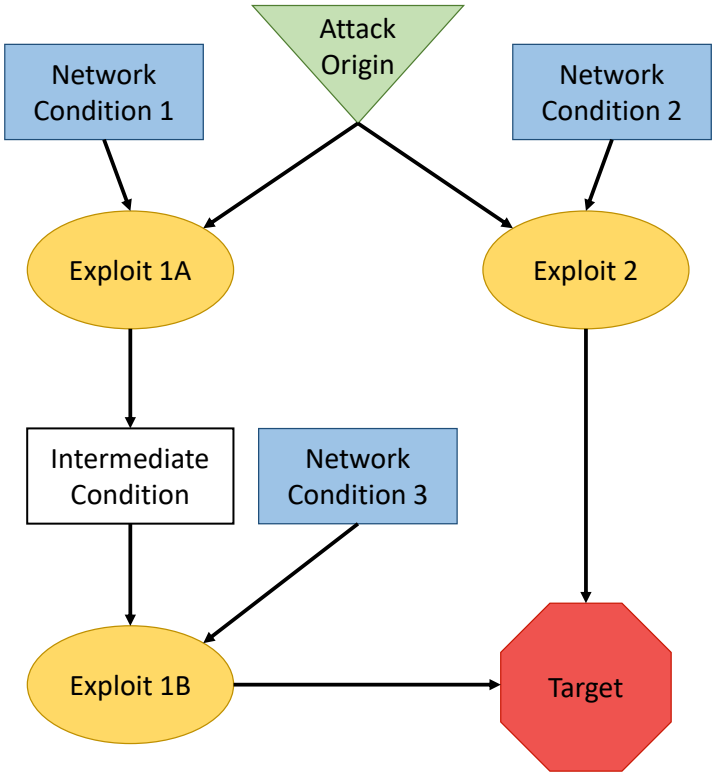


Figure 1: An example of an attack graph.

that following the directions along the edges will not result in a closed loop. This property is useful for analyzing the graph, and is appropriate for many cybersecurity applications. The general justification for this property is that the attacker is not likely to yield access or privileges to ICS devices once he has obtained them. This assumption is often valid, but does not capture the situation where the defender partially expunges the attacker from the system.

2.1.3 Petri Nets

Petri nets are another graphical mathematical tool used to evaluate cybersecurity. They were originally developed to model chemical reactions, but have since been applied to many other systems and processes. Other applications include biological systems [105], cryptography [18], communication protocols [37], and automatic control [31].

Petri nets contain two types of elements: places and transitions [71]. Places are represented by ellipses and model passive components. Transitions are represented by bars or rectangles and model active components. Transitions and places are connected by directional arrows called arcs. Arcs do not connect places to other places or transitions to other transitions.

Places can contain an integer number of tokens, represented by dots inside the place. A particular configuration of tokens over the Petri net is called a marking. A transition can only occur, or be fired, if each of the input places has at least one token. When a transition is fired, one token is withdrawn from each of the input places and one token is added to each of the output places. The progression of the Petri net's markings represents the dynamics of the modelled system [19].

An example of a Petri net is shown in Figure 2 [19]. Places are designated by P and transitions are designated by T. The places P_2, P_4, and P_6 have tokens. The transitions T_2 and T_5 are enabled. The transition T_6 is not enabled because P_7 does not have a token.

There are several methods to analyze Petri nets [103]. One example is reachability analysis. Reachability analysis provides the markings that can be reached from an initial marking. To conduct a reachability analysis, a reachability tree is constructed, where each

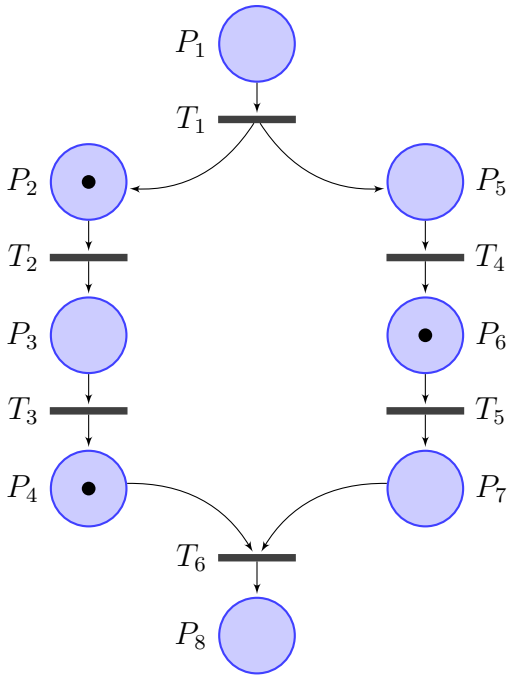


Figure 2: An example of a Petri net [19].

node in the tree is a particular marking, and each arc in the tree is a transition firing. A similar idea to reachability is coverability. Coverability problems examine whether a particular marking is part of a reachable marking.

In the context of cybersecurity, Petri nets can be used to model an attacker's exploitation of system vulnerabilities. Tokens can be used to represent the access or privileges gained by the attacker through an exploit. Transitions can be fired once the attacker has accumulated the necessary privileges and tokens to allow the attack to continue. Given the initial marking of the Petri net, reachability analysis can show which privileges can be obtained by the attacker throughout the course of the attack.

One limitation of Petri nets is that solving complex nets can be computationally expensive [103]. For complex Petri nets, constructing a reachability tree is not feasible and coverability graphs may exhibit poor behavior. Simulation-based approaches are often

used to study reachability for these nets. Large-scale simulation efforts generally involve the initialization of the Petri net with a given marking, and randomly selecting enabled transitions to fire. While this approach is straightforward, it is time-consuming.

2.2 Game-Theoretic Approaches to Cybersecurity

In order to properly allocate security resources in a CPS, one must consider the motivations and resources of all decision-makers who interact with the system. By considering these factors, we can avoid allocating excessive security resources towards low-impact components or allocating insufficient resources towards high-impact components. This prioritization is simple when only one decision-maker interacts with the system, but becomes complex when multiple decision-makers with different priorities must be considered.

The field of game theory enables us to analyze the interactions of multiple rational decision-makers. The security of a CPS can be viewed as a non-cooperative game between a defender and an attacker. The attacker's objectives vary, and may include environmental damage, economic damage, physical damage, or loss of life. The defender's objectives include maximizing profits, maintaining system operations, and avoiding environment, health, or safety incidents. Game theory was first applied to cyber-physical security challenges in the early 2000's to study network intrusion detection and network security [54, 9, 7]. Since then, many researchers have applied game theory to analyze the security of CPSs. The remainder of this chapter discusses different types of games and their application to the security of CPSs.

2.2.1 Strategic Form Games

The most simple type of game is the strategic form game. Strategic form games are used to describe scenarios where the players make decisions simultaneously. These games can provide basic insight towards the interaction between decision-makers and are often used as a starting point before developing more complex games.

Several strategic form games with national security applications are discussed in [15]. Applications include arms races, optimal threats, crisis stability, and deterrence. For example, consider the deterrence game. In this game, there are two parties capable of inflicting harm upon the other. Each party believes that the other party will retaliate to an attack with a particular probability. By studying this game, the conditions are identified where neither party takes offensive action and the conflict is avoided. Although these games were developed to describe interactions between nations with nuclear weapons capabilities, there are parallels with the development of offensive cyber-capabilities.

A strategic form game was also used to identify effective security strategies for an ICS in [59, 58]. In this game, the attacker chooses a set of sensors to attack and the defender chooses a set of sensors to defend. If particular sets of sensors are successfully hacked, the attacker can compromise the defender's observability of the system and cause damage unbeknownst to the defender. An effective security strategy was identified by applying a game-theoretic framework to examine the cost and benefit of each attack and defense. This example is discussed in greater detail in Appendix A.

Several strategic form games with network security applications are discussed in [8]. Applications include intrusion detection, malicious behavior on social networks, wireless networks, and vehicular networks. In the vehicular networks example, vehicles can communicate with one another and with roadside units. The roadside units can tunnel data to improve communication between vehicles. The attacker can choose regions of the road to target with communication jamming attacks. A game-theoretic approach is used to determine the optimal distribution of the roadside units to maintain an effective communication network.

2.2.2 Stochastic Games

The field of stochastic game theory is used to study the interactions of decision-makers when the outcome of the decisions are uncertain [27]. As decisions are made by each player, the game traverses a set of states that describe the progress of the game. At each decision point, the players choose an action. The resulting actions provide the probability of

transitioning to each state in the stochastic state space. Each player receives an immediate reward resulting from the actions and state transition. Each player seeks to maximize his cumulative reward earned throughout the game. Using optimization techniques, we can determine the optimal action for each player at each state and determine each player's expected cumulative reward.

Stochastic games are applied to address network security applications in [8]. Applications include intrusion detection, the security of interconnected systems, and malware filter placement. While these systems are not inherently cyber-physical, the stochastic game-theoretic approach can be extended to ICSs and CPSs.

A stochastic game was used to study the cybersecurity of a boiling water power plant [65]. The states were defined by the operational status of the plant and observability of the cyber-attack. The game was conducted in continuous time with an accompanying continuous-time model of the plant. The effects of various parameters such as the penalty to the attacker if caught, attack detection probability, and attack speed were examined to identify favorable cybersecurity scenarios for the plant.

Stochastic games have also been studied for the response to cyber-incidents in nuclear power plants. A case study was given for a digital feedwater system in [108]. The game was characterized as a discrete-time competitive Markov decision process and the states were defined by the operational status of the ICS devices and plant components. Fault tree and event tree analysis were used to describe the transitions between states. A dynamic programming approach was used to find a Nash equilibrium. The Nash equilibrium provided the optimal defense action for each state in the game.

2.2.3 Bayesian Games

Bayesian games are used to study interactions where at least one player has uncertainty about the characteristics of the other players. Within the context of a security game, the defender may be unsure about the parameters describing the attacker. The defender could

then construct several attacker "types", each of which describes the attacker with a different set of parameters. The defender then assigns a probability to each type, and the game can be solved with traditional game theory methods [33, 34, 35].

Bayesian games have been applied to various physical security challenges. For example, consider the approach developed in [78] for airport security patrols. The security challenge is cast as a Bayesian Stackelberg game — a game where one player must commit to an action before the other player. For the airport security problem, the defender must first commit to a patrol schedule before the attacker selects a target. An efficient algorithm was designed to identify the best defense strategy and randomize patrols within that strategy.

Bayesian games are applied to network security application in [8]. Examples include intrusion detection and the security of wireless networks. While these systems are not inherently cyber-physical, the Bayesian game-theoretic approach can be extended to ICSs and CPSs.

Bayesian games are applied to a cyber-physical manufacturing setting in [107]. A medium-sized manufacturing facility with a continuous production line is studied. The facility could be targeted by an insider or a cybercriminal with varying risk attitudes. Using this approach, an effective security strategy can be identified based on the defender's beliefs about the attacker's characteristics.

2.2.4 Limits of Current Practice

Most stochastic games are structured as two-player games. While this approach provides some insight into security problems, it is not representative of the security challenges faced by NPP. NPP face a variety of adversaries with different capabilities and motivations. For example, an NPP must be resilient to threats such as state actors, terrorists, disgruntled employees, and cyber criminals. These threats have different technical abilities, resources, and goals. To accurately use game theory to inform security decisions, these threats and their objectives must be addressed.

It can be difficult to estimate the values of parameters used to describe the adversaries in a game. For example, it is difficult to estimate the reward that the attacker will gain if

an attack is successful. These rewards are functions of the subjective value that the attacker places on non-monetary outcomes such as the loss of life and press coverage. One method of addressing these uncertainties is by applying a Bayesian game structure. Little research exists that combines the benefits of a Bayesian game with the framework of a stochastic game [4].

Another challenge in analyzing stochastic games is managing the size of the stochastic state space. For example, consider a system consisting of n components, each of which has two possible states (e.g. functional and nonfunctional). The resulting stochastic state space will have a dimension of 2^n states to account for each possible combination of component states. A large state space presents computational challenges and challenges in interpreting the results of the game. Methods are desired to manage the size of the stochastic state space so that the analysis of the game can be used to inform practical security decisions.

Many game-theoretic security approaches rely on the Nash equilibrium as the game's solution concept. A Nash equilibrium occurs when each player has selected a strategy that is a best response to the strategies of the other players. The Nash equilibrium is an attractive solution concept because each player can predict the equilibrium, and predict that each other player will predict it, and so on. No player has an incentive to unilaterally deviate from the equilibrium — doing so would only decrease that player's payoff.

While the Nash equilibrium solution method is widely implemented, it is not valid if it cannot be predicted by all players in the game. For complex stochastic games, it is possible that not all players will model the game in the same manner and arrive at the same Nash equilibrium. This becomes increasingly challenging if the game has multiple Nash equilibria.

Another challenge with the Nash equilibrium solution is that it provides a static defense strategy. If both players select the strategies corresponding to the same Nash equilibrium solution, then a static defense strategy is not only appropriate, it is optimal. But, if the players do not arrive at the same Nash equilibrium solution, it would be advantageous to allow the defense strategy to change over time to arrive at a best response to the adversary's strategy.

2.3 Summary of Limits of Current Practice

In summary, this research will address the following key limits of current practice:

1. Lack of research on stochastic Bayesian games for security applications
2. Challenge of determining game-theoretic parameters
3. Lack of formal methods to manage the size of the stochastic state-space
4. Challenge of validating the Nash equilibrium as a solution method
5. Challenge of designing dynamic defense strategies

By addressing these limitations, we will be able to apply stochastic and Bayesian game theory to design holistic cyber-physical defense strategies for NPP that address a variety of adversaries.

3.0 Stochastic Bayesian Games

This chapter provides a theoretical foundation for this work. Stochastic and Bayesian games are used to address different types of uncertainty in a game. In a stochastic game, the outcomes of the players' actions are uncertain. In a Bayesian game, the players are uncertain about the parameters that govern the decision-making of the other players. This chapter will discuss stochastic and Bayesian games in greater detail, and introduce the combination of these two types of games: a stochastic Bayesian game.

3.1 Preliminaries

This section provides an overview of preliminary game theory concepts. These concepts provide the necessary framework to discuss more sophisticated games. We introduce the game theory vernacular within the context of strategic form games. Strategic form games are used to model situations where players select their strategies simultaneously. We discuss the components of a game, and introduce the Nash equilibrium – the fundamental solution method used in game theory.

A strategic form game is defined by three components:

1. Players: the agents who are participating in the game
2. Strategies: the set of choices available to each player
3. Utilities: the numerical payoffs for each player. The utilities are functions of the strategies chosen by the players.

Strategic form games are often represented using matrices. Consider the prisoners' dilemma game shown in Table 1. Utilities for each player are given for each strategy intersection, with Row's utility listed first. In this game, two prisoners are each given an opportunity to reduce their sentences. Each prisoner is spoken to individually, and told that

Table 1: Prisoners' dilemma in the strategic form.

		Column	
		Cooperate (C)	Not cooperate (NC)
Row	Cooperate (C)	-2, -2	10, -5
	Not cooperate (NC)	-5, 10	0, 0

their sentence will be reduced if they cooperate and implicate the other prisoner. If neither prisoner cooperates, their sentences remain unchanged, and if both prisoners cooperate, both prisoners receive worse sentences.

A best response is a player's optimal strategy, given the strategies selected by the other players. For player i with utility function u_i, strategy s_i^* is a best response to the strategies s_{-i} chosen by the other players (denoted by $-i$ subscript) if

$$u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i}) \quad \forall s_i \tag{3.1}$$

In the prisoners' dilemma game, the best response to cooperation is cooperation. The best response to not cooperating is also cooperation. Because cooperating is always more profitable than not cooperating, it is said that not cooperating is strictly dominated by cooperating. A rational player would never select a strategy that is strictly dominated because there is always a more profitable strategy than the dominated strategy. Therefore, in this prisoner's dilemma game, it is always best for each prisoner to cooperate, regardless of the strategy chosen by the other prisoner.

Suppose both prisoners choose to cooperate. Here each prisoner is playing a best response to the strategy of the other prisoner. This is called a Nash equilibrium. A strategy profile (s_i^*, s_{-i}^*) is a Nash equilibrium if

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*) \quad \forall s_i, i \tag{3.2}$$

At a Nash equilibrium, no individual player has an incentive to unilaterally change his strategy. In this game, deviation from the Nash equilibrium of (C, C) to (NC, C) or (C, NC) would reduce the utility of the deviating player from -2 to -5.

Note that although (C, C) is a Nash equilibrium, it is not the most profitable outcome for the players. The strategy profile (NC, NC) yields a greater utility for both players. Why, then, is (C, C) the game's solution rather than (NC, NC)? Noncooperation is not the solution because it is not a stable strategy profile. Although (NC, NC) results in greater payoffs than (C, C), there is incentive for players to unilaterally deviate from the (NC, NC) profile. For example, suppose Row believes Column will not cooperate. Row will then either receive a utility of ten for cooperating, or a utility of zero for not cooperating. Cooperating is Row's best response. Column can reason similarly about Row's actions. Therefore, both players choose to cooperate.

Nash proved that at least one Nash equilibrium exists for every finite strategic form game [63]. There is one nuance to this statement: the Nash equilibrium is not guaranteed to be an equilibrium in pure strategies. The equilibrium may include mixed strategies, i.e. probability distributions over the pure strategies. Given a set of pure strategies, $S_i = \{s^1, s^2, \ldots, s^N\}$, a mixed strategy is defined as a probability vector $\sigma = (p^1, p^2, \ldots, p^N)$ where $0 \leq p^k \leq 1$ for $k = 1, \ldots, N$, and $\sum_{k=1}^{N} p^k = 1$. A pure strategy can be expressed as a mixed strategy where one pure strategy is assigned a probability of one and all other pure strategies are assigned a probability of zero. Pure strategies that are assigned positive probability are said to be in the support of the mixed strategy. We will discuss mixed strategy Nash equilibria with an example.

Consider the game of matching pennies shown in Table 2. In matching pennies, each player has a penny, and each secretly places it in either the heads or tails position. Row receives positive utility if the pennies match, and Column receives positive utility if the pennies do not match.

The best responses for each player are identified by underlined utilities in Table 2. In this game, there is no pure strategy profile where both players are playing a best response, therefore there is not a pure strategy Nash equilibrium. There is a mixed strategy Nash equilibrium where each player assigns a probability of 0.5 to heads and a probability of

Table 2: Match pennies in the strategic form.

		Column	
		Heads (H)	Tails (T)
Row	Heads (H)	$\underline{1}$, -1	-1, $\underline{1}$
	Tails (T)	-1, $\underline{1}$	$\underline{1}$, -1

0.5 to tails. The utility earned by each player at a mixed strategy Nash equilibrium is an expected value taken over the mixed strategies. For a general two-player game played by i and j, the expected utility of a mixed-strategy for i is

$$u_i(\sigma_i, \sigma_j) = \sum_{x \in S_i} \sum_{y \in S_j} \sigma_i(s_x)\sigma_j(s_y)u_i(x, y) \tag{3.3}$$

In the matching pennies game, the expected utility for both players at the Nash equilibrium is zero.

An example of a strategic form game for cybersecurity applications is given in Appendix A. Now that we have defined basic game theory concepts, we can begin our discussion of Bayesian games, stochastic games, and stochastic Bayesian games.

3.2 Bayesian Games

A Bayesian game is a game of incomplete information in which some players do not necessarily know the payoffs of other players. Each uncertain player constructs a set of "types" that describe each possible set of parameters governing the other players. Within the context of NPP security, Bayesian games can be used to address the defender's uncertainty about the attacker's parameters.

Consider the Bayesian game shown in table 3. In this game, an NPP defender is attempting to protect the plant from a threat, but is unsure whether the attacker is a

Table 3: A Bayesian security game. For each strategy intersection, the defender's payoff and attacker's payoff are provided.

Type 1: Disgruntled Employee

Probability = 0.8

	Attack 1	Attack 2
Defense 1	0, 2	-10, 1
Defense 2	-3, 10	0, -2

Type 2: Terrorist

Probability = 0.2

	Attack 1	Attack 2
Defense 1	0, -1	-10, 10
Defense 2	-3, 3	0, 4

disgruntled employee or a terrorist. A separate game matrix is constructed for each attacker type. Within each matrix, the rows correspond to the defender's strategies, the columns correspond to the attacker's strategies, and the utilities of both players are specified for every strategy intersection.

In this game, it can be seen that if the attacker is Type 1, then Attack 1 strictly dominates Attack 2. If the defender knew he were facing a Type 1 attacker, he would then conclude that if the attacker is going to select Attack 1, he should select Defense 1 to maximize his payoff. Similarly, if the attacker is Type 2, then Attack 2 strictly dominates Attack 1. If the defender knew he were facing a Type 2 attacker, he would then conclude that if the attacker is going to select Attack 2, he should select Defense 2 to maximize his payoff. Bayesian game theory provides a method for the defender to choose his strategy given that he is uncertain about the attacker's true type.

A probability distribution is first assigned to the set of player types to describe the player's uncertain belief about each player type's likelihood. With the assumption that the types and type distributions are common knowledge, the Bayesian game can be transformed from a game of incomplete information to a game of imperfect information, and solved using standard Nash equilibrium techniques [29]. The equilibrium of the game in Table 3 occurs when the defender plays Defense 1, a Type 1 attacker plays Attack 1, and a Type 2 attacker plays Attack 2.

The attacker may also be uncertain about the parameters that govern the defender. For example, the attacker may be uncertain whether the defender is risk-averse or risk-embracing. A Bayesian game can be constructed in which both players have uncertainty regarding their opponent's parameters. Again, with the assumption that the types and type distributions are common knowledge, the Bayesian game can be transformed into a game of imperfect information and solved using Nash equilibrium techniques.

3.2.1 Bayesian Games in the Extensive Form

The extensive form is a representation of a game where the actions chosen by the players are represented as a decision tree. The extensive form can include the assumption that the players have a common prior belief regarding the distribution of types. The common prior assumption is enacted by introducing a chance node, Nature, that stochastically assigns the types to each player. Information sets are used to define the knowledge that is available to each player, such as Nature's type assignments and the actions selected by the other players [22].

Consider a Bayesian simultaneous game played by i and j. Let the set $\Theta_i = \{\theta_i^1, \theta_i^2\}$ contain the types of i and let the set $\Theta_j = \{\theta_j^1, \theta_j^2\}$ contain the types of j, where the superscript notation indexes the types within their respective sets. Let the probability that i is type θ_i^1 and j is type θ_j^2 be denoted p_{12} and let the probability of the other types be similarly defined. Let the action sets available to the player be $A_i = \{a_i^1, a_i^2\}$ and $A_j = \{a_j^1, a_j^2\}$. In this game, the players choose their actions simultaneously and without communication. Let the utility functions be $u_i(s_i, s_j, \theta_i, \theta_j)$ and $u_j(s_i, s_j, \theta_i, \theta_j)$, where $s_{i/j}$ is the player's strategy for action selection. The extensive form of this Bayesian game is shown in Figure 3.

The extensive form of a game consists of nodes and branches. Nodes in the tree represent a point where a decision is to be made, and branches represent each of the possible decisions. Nodes that are owned by players are shaded and chance nodes are not shaded. At a player's node, that player must choose from the set of actions available to determine the progress of the game. At a chance node, the progress of the game is dependent on a probability distribution over the branches.

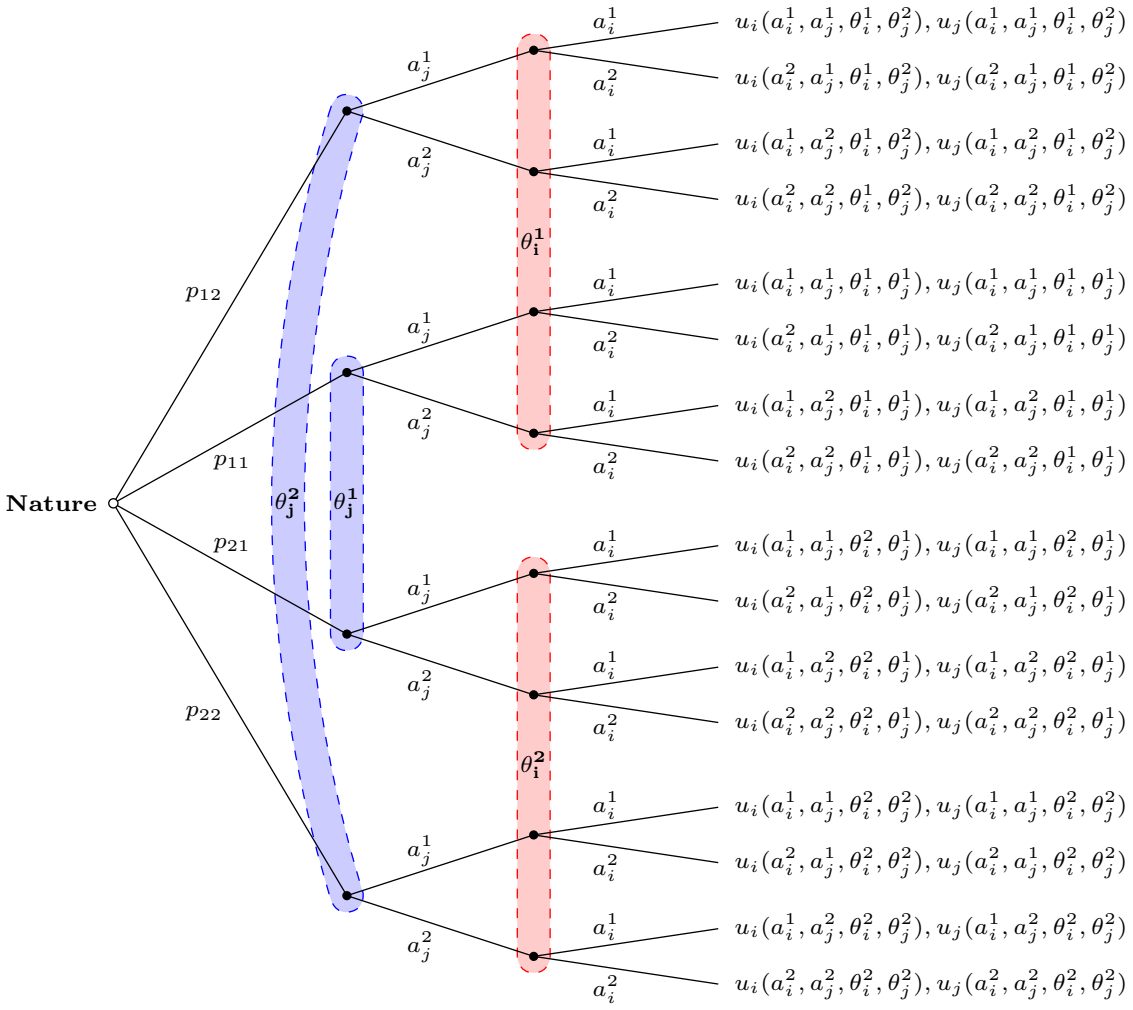


Figure 3: The extensive form of a Bayesian simultaneous game.

In the game in Figure 3, the only chance node is the Nature node that assigns the players their types. Chance nodes can also be used to model scenarios where the outcome of the players' actions is uncertain. For example, consider a scenario where an attacker has caused a device in an NPP to fail. For the device failure to cause a larger system failure, redundant plant systems and safety systems must also fail. The outcome of the attack is therefore dependent not only on the actions selected by the players, but also on the occurrence of external events. These external events can be represented by chance nodes, and their probabilities can be determined using risk analysis tools such as event tree analysis [91] or Bayesian networks [26].

Information sets are used to model the players' knowledge of their types and the actions chosen by the other players. In the game in Figure 3, i's information sets are shown by the red rounded rectangles, and j's information sets are shown by the blue rounded rectangles. All of the nodes contained in a given information set are indistinguishable to that player. Consider the information set for i of type θ_i^1. This information set is preceded by both types of j because i does not know whether Nature has assigned j type θ_j^1 or θ_j^2. It is also preceded by the full action set of j for both types because while i is choosing an action, i does not know the action chosen by j. The other information sets are similarly constructed.

3.2.2 Bayesian Nash Equilibria

The solution method of Bayesian games is called a Bayesian Nash equilibrium. The Bayesian Nash equilibrium is a strategy profile where each type of each player has selected a best response to the strategies of the other players. Every finite Bayesian game has at least one Bayesian Nash equilibrium [33, 34, 35].

Before discussing the Bayesian Nash equilibrium, we must define notation. The components of a Bayesian game are the players, their actions, their strategies, their types, their utility functions, and the probability distribution over the types.

Players Let $\mathcal{I}$ denote the set of players. Let $i \in \mathcal{I}$ denote an individual player i. Let I be the total number of players. For notational convenience, let the subscript notation "$-i$" denote all players except player i.

Types Let Θ_i denote the set of types belonging to player i. Let $\theta_i \in \Theta_i$ be a type of i.

Actions Let $A_{i,\theta}$ denote the set of actions available to player i of type θ. Let $a_{i,\theta} \in A_{i,\theta}$ be an action selected by player i of type θ.

Strategies Let s_i, denote a strategy chosen by player i. A player's strategy specifies the actions selected by every type of that player. Let a strategy profile be denoted $s = (s_1, \ldots, s_I)$. A strategy profile defines the action chosen by every type of every player.

Utility functions Let $u_i(s_1, \ldots, s_I, \theta_1, \ldots, \theta_I)$ be the utility function of player i. Generally, the utility is a function of the strategies selected by the players and their types.

Probability distribution Let $p(\theta_1, \ldots, \theta_I)$ be the probability distribution over the types.

The expected utility of player i with type θ_i is

$$\mathbb{E}\left[u_i(s_i, s_{-i}, \theta_i, \theta_{-i})\right] = \sum_{\theta_{-i} \in \Theta_{-i}} p(\theta_{-i} \mid \theta_i) u_i(s_i, s_{-i}(\theta_{-i}), \theta_i, \theta_{-i}) \tag{3.4}$$

The best response of a player to the strategies of the other players is the strategy that maximizes the player's expected utility.

A Bayesian Nash equilibrium is a strategy profile where all types of all players have selected best responses to the other players' strategies. By this definition, no type has incentive to unilaterally deviate from its equilibrium strategy. A strategy profile $s^* = (s_1^*, \ldots, s_I^*)$ is a Bayesian Nash equilibrium if

$$\mathbb{E}\left[u_i(s_i^*, s_{-i}^*, \theta_i, \theta_{-i})\right] \geq \mathbb{E}\left[u_i(s_i, s_{-i}^*, \theta_i, \theta_{-i})\right] \tag{3.5}$$

for all $i \in \mathcal{I}$, for all $s_i \in S_i$, and for all θ_i that have a positive probability.

3.2.3 Solution Methods

A Bayesian game can be transformed into a normal-form game using the Harsanyi transformation. The Harsanyi transformation uses the common prior assumption to transform the incomplete information game to an imperfect information game. An example of the Harsanyi transformation is shown in Table 4. The row player has one type and the column player has two types. In both the original and the transformed game, the row player has two pure strategies. In the original game, the column player has two pure strategies, but in the transformed game the column player has four strategies. A single pure strategy in the transformed game includes a pure strategy for each of the column player's types in the original game (strategy YZ indicates that Type 1 plays Y and Type 2 plays Z). In general, if the column player has α pure strategies and β types, the column player in the normal-form game has α^{β} pure strategies. The payoffs for the players are calculated as a weighted sum of the payoffs from the Bayesian game, where the weighting factors come from the type distribution. Once the game is in normal form, several solution methods can be used [102].

The Nash equilibria of Bayesian games in the extensive form can be found using software such as Gambit [61]. Gambit is open-source software used to analyze finite non-cooperative games. Gambit has both a command line interface and a graphical user interface, and is particularly useful for visualizing game theory problems. Analyzing large games in Gambit is not feasible. Using Gambit, we can identify each player's Nash equilibrium strategy and expected utility for games of a reasonable size. Large Bayesian games require specialized algorithms to be solved [67, 68, 78]. An example application is given in Appendix B.

3.3 Stochastic Games

A stochastic game is a dynamic system that evolves as the players take action. As the game progresses through time, it traverses a finite set of states, $\mathcal{S}$, that describe the environment of the players' interaction. Consider the simple stochastic game played by an

NPP defender and an attacker in Figure 4. In this example, the stochastic state space includes three states. In the normal operating state, the plant is operating as expected. In the penetrated state, the attacker has breached the plant's defenses. Finally, in the damaged state the attacker has caused damage to the plant. In this game, the damaged state is an absorbing state — the game concludes when this state is reached.

Stochastic games may be modelled in continuous or discrete time. This work considers discrete-time games. At each discrete time step in the game, each player selects from a finite set of actions. The defender's action set could include physical defenses such as maintaining guard stations, and cyber defenses such as maintaining antivirus software. Similarly, the attacker's action set could include both cyber and physical attacks.

The actions selected by each player affect the probability of the state transitioning to each of the other states in the stochastic state space. Within the context of a security game, these transition probabilities are dependent on a variety of factors, including the skill of the players and the complexity of their actions. Consider the transitions available in the game's normal state. If an attacker attempts to exploit a well-known vulnerability but the defender has opted to address that vulnerability, then the game is likely to transition to the normal state rather than the penetrated state.

Let the set of actions available to player i/j at state s be denoted $a_{i/j}(s)$. When both players select their actions, the resulting action profile $a_{i,j} = (a_i(s), a_j(s))$ determines the probability of transitioning from state s to every other state in the stochastic state space. After an action profile has been selected, the transition to the next state is determined stochastically by a transition vector given by a function

$$T(s, a_i, a_j) = \big(p(s_1|s, a_i, a_j), p(s_2|s, a_i, a_j), \dots, p(s_N|s, a_i, a_j)\big) \tag{3.6}$$

For this work, given N states, $\sum_{s'}^{N} p(s'|s, a_i, a_j) = 1$ for all s. This is not required in general, and the value $1 - \sum_{s'}^{N} p(s'|s, a_i, a_j)$ gives the stopping probability in cases where the sum does not equal one. The stopping probability is the probability that the game ends in current state given the action profile.

After each decision point, each player receives an immediate utility that is a function of the current state, the actions selected by all players, and the state to which the game

transitions. Within the context of a security game, the immediate utility of the defender is generally a function of the cost of the selected defense action and the stochastic outcome of the defense and attack action profile. The immediate utility for the attacker is similarly defined. A cumulative utility function is defined to assess the immediate utilities earned by the players throughout the game. An example of a cumulative utility function is

$$u_i(s^0, \sigma_i, \sigma_j) = \sum_{t=0}^{\infty} \beta_i^t \mathbb{E}[r_i(s^t, a_{i,j}^t, s^{t+1})] \tag{3.7}$$

The cumulative utility is dependent on i's and j's strategies, σ_i and σ_j, and the initial state, s^0. The discount factor $\beta \in (0, 1)$ describes i's preferences for utility earned earlier in the game relative to utility earned later in the game [27, 75]. The superscript t on the discount factor is an exponent, and the superscript on the state and action profile variables is a time index. The function $\mathbb{E}[\cdot]$ denotes the expectation over the states and strategies for the immediate utility function, r_i.

Strategies in stochastic games are rules that determine the actions selected by a player at any point in the game. There are four types of strategies: Markov strategies, semi-Markov strategies, stationary strategies, and behavior strategies [27]. Markov strategies provide a decision rule, f_t for every time $t = 0, 1, 2, \ldots$, where f_t is determined by t and s^t. Semi-Markov strategies are Markov strategies that are also dependent on the initial state s^0. Stationary strategies provide a decision rule that is only a function of the current state (i.e. a Markov strategy without time dependence). Finally, behavior strategies are the most general type of strategy. They provide a decision rule f_t that is a function of t and the history, $H^t = (s^0, a_{i,j}^0, s^1, a_{i,j}^1, \ldots, a_{i,j}^{t-1}, s^t)$, where superscripts are time indices. Behavior strategies are used in this work.

3.3.1 Nash Equilibrium

Here we present the ε-Nash equilibrium of a two-player stochastic game with semi-Markov strategies [108]. The existence of Nash equilibria in more complicated stochastic games and the solution methods to find them are both areas of ongoing research [69, 62].

First we define the value, $v_i(s^0)$, of a game with initial state s^0 for player i as the quantity $u_i(s^0, \pi_i, \pi_{-i})$ such that

$$\inf_{\pi_2} \sup_{\pi_1} u_i(s^0, \pi_i, \pi_{-i}) = \sup_{\pi_1} \inf_{\pi_2} u_i(s^0, \pi_i, \pi_{-i}) \tag{3.8}$$

The strategy profile (π_1, π_2) is an ε-Nash equilibrium if there are no unilateral deviations that result in profit greater than or equal to ε. That is,

$$v_i(s^0) \geq u_i(s^0, \pi_i, \pi^*_{-i}) - \varepsilon, \quad \forall \pi_i, i \in \{1, 2\} \tag{3.9}$$

If ε is zero, the ε-Nash equilibrium is a Nash equilibrium.

An ε-Nash equilibrium can be found by solving the optimization problem given in Equation 3.10, where the objective function is given by Equation 3.11 and Equation 3.12 defines a variable for notational convenience [108].

$$\begin{aligned}
\underset{v_1, v_2, \pi_1, \pi_2}{\text{minimize}} \quad & \psi(v_1, v_2, \pi_1, \pi_2) \\
\text{subject to} \quad & \sum_{a_i \in A_i} \pi_i(s, a_i) h_{-i}(s, a_i, a_{-i}) \leq v_{-i}(s), \quad \forall s \in \mathcal{S}, i \in \{1, 2\}, a_i \in A_i(s) \\
& \sum_{a_i \in A_i} \pi_i(s, a_i) = 1, \quad \forall s \in \mathcal{S}, i \in \{1, 2\} \\
& \pi_i(s, a_i) \geq 0, \quad \forall s \in \mathcal{S}, i \in \{1, 2\}, a_i \in A_i(s)
\end{aligned} \tag{3.10}$$

$$\psi(v_1, v_2, \pi_1, \pi_2) = \sum_{i=1}^{2} \sum_{s \in \mathcal{S}} \left[v_i(s) - \sum_{a_1 \in A_1(s)} \sum_{a_2 \in A_2(s)} \pi_1(s, a_1) \pi_2(s, a_2) h_i(s, a_1, a_2) \right] \tag{3.11}$$

$$h_i(s, a_1, a_2) = \sum_{s' \in \mathcal{S}} p(s' | s, a_1, a_2) \big[r_i(s, a_1, a_2, s') + \beta v_i(s') \big] \tag{3.12}$$

The first constraint is obtained from the definition of the Nash equilibrium. The second and third constraints are required to ensure that the mixed strategies are well-defined discrete probability distributions. The nonlinear characteristics of this optimization problem present challenges in finding the global optimum rather than a local optimum [108].

3.4 Stochastic Bayesian Games

A stochastic Bayesian game (SBG) combines the features of stochastic games and Bayesian games. The stochastic elements of the SBG enable the consideration of uncertainty in the interactions of the attacker and defender. As in a stochastic game, a set of states define the environment for the players' interactions, and the state transitions occur stochastically as a function of the players' actions. The Bayesian elements of the SBG enable the consideration of the uncertainty regarding the attacker's characteristics. As in a Bayesian game, a set of types are defined to describe possible behaviors of at least one of the players. This combination is useful for the analysis of NPP security because it enables plant defenders to optimize their security decisions in the presence of uncertainty. The remainder of this section will discuss the application of an SBG to the security of a nuclear system.

3.4.1 Bayesian Learning of the Adversary's Parameters

As the SBG is played, the defender can learn the characteristics of the attacker. As the players interact through the SBG, the defender can use the attacker's actions to draw conclusions about the attacker's game-theoretic parameters. The defender can draw conclusions about not only the parameters of a single type of attacker, but about the parameters governing the behavior of all potential attackers in the attacker's type space [6].

The defender first hypothesizes a set of possible attacker types, Θ_A. For every $\theta_j \in \Theta_A$, the defender has an initial belief, $P(\theta_j|H^0)$, which defines the probability that the attacker has type θ_j given the initial information H^0 available to the defender. The defender also has an initial estimate, $p_j^0 \in [p_j^{min}, p_j^{max}]$, of the game-theoretic parameters governing each type θ_j. At each time $t > 0$, a new estimate of the parameters, p_j^t, can be calculated based on the updated history, H^{t-1}, available to the defender. Here we use approximate Bayesian updating to estimate p_j^t. Approximate Bayesian updating is summarized in Algorithm 1.

Algorithm 1 Approximate Bayesian updating [6]

1: Represent the belief $P(p_j^t|H^{t-1},\theta_j)$ as $\hat{p}$: a polynomial of degree d

2: Represent the belief $P(a_j^{t-1}|H^{t-1},\theta_j,p)$ as $\hat{f}$: a polynomial of degree d

3: Compute the polynomial product $\hat{g}=\hat{f}\cdot\hat{p}$

4: Collect samples $D=(p^{(l)},\hat{g}(p^{(l)}))$

5: Fit polynomial $\hat{h}$ of degree d to D

6: Compute $I=\int_{p_{\min}}^{p_{\max}}|\hat{h}(p)|dp$

7: Set new belief $P(p_j^t|H^t,\theta_j)=\hat{h}/I$

8: Extract new estimate p^t from $P(p_j^t|H^t,\theta_j)$

A demonstration of approximate Bayesian updating is shown in Figure 5. Figure 5a shows the prior belief over p given H^{t-1} and type θ_j, and the true value of p. This belief is shown as a polynomial of degree d. Figure 5b shows the belief regarding the action given H^{t-1} and θ_j as a function of p. This belief is sampled, and fit with a polynomial, $\hat{f}$. Figure 5c shows the polynomial product of the prior belief and $\hat{f}$. This polynomial is sampled, and represented as polynomial $\hat{h}$ with degree d. This belief is normalized to obtain the posterior belief. The posterior belief is shown in Figure 5d. The updated estimate of p is selected as the value with the greatest belief density.

The use of polynomial approximations in approximate Bayesian updating does present one challenge — the approximations may result in negative values of the belief within the range of p values [6]. Two countermeasures must be taken to address this challenge. The first countermeasure is to take the absolute integral of $\hat{h}$ to obtain I. The second countermeasure is to only sample points from $\hat{g}$ that have positive values. This helps prevent propagation of negative minima.

Using the new estimate p_j^t for type θ_j, the current belief regarding the type distribution is updated by

$$P(\theta_j|H^t)\propto P(a_A^{t-1}|H^{t-1},\theta_j,p_j^t)P(\theta_j|H^{t-1}) \tag{3.13}$$

The probability functions used to calculate $P(\theta_j|H^i)$ are dependent on the information available to the defender, the attacker's action, the type definitions, and the estimates of the type's parameters.

3.4.2 Harsanyi-Bellman Ad Hoc Coordination

The typical solution method applied in game theory is the Nash equilibrium. At a Nash equilibrium, each player's strategy is the best response to the strategies of the other players. Because of this property, no player has an incentive to unilaterally deviate from the Nash equilibrium. For this solution method to predict the play of a real game, the equilibrium must be identified by all players and that fact must be common knowledge. For a complex stochastic security game, this assumption may not be valid. Instead, we opt for a solution method that informs security decisions in real-time as the game is played.

The solution method we select for the SBG is Harsanyi-Bellman Ad Hoc coordination (HBA) [5, 4]. In contrast to the predictive nature of the Nash equilibrium, HBA is a tool used to select actions in real-time as the game evolves. HBA combines the concepts of the Bayesian Nash equilibrium [34] and Bellman optimal control [13] to calculate optimal actions based on the action history observed by the players. Given a postulated set of types describing the opponent, HBA first uses the action history to calculate a discrete probability distribution over the type set. Using this updated type distribution, HBA then calculates the optimal strategy to maximize that player's expected cumulative reward.

Mathematically, HBA is defined as $a_D^t \sim \arg\max_{a_D} E_{s^t}^{a_D}(H^t)$, where $E_{s^t}^{a_D}(H^t)$ is the expected cumulative reward for the defender, D, after history H^t, including taking action a_D in state s at time t. For a two-player game, the expected cumulative reward is given by

$$E_s^{a_D}(\hat{H}) = \sum_{\theta_j \in \Theta_A} P(\theta_j|\hat{H}) \sum_{a_i \in A_j} \pi_j(\hat{H}, a_i, \theta_j) Q_s^{a_D,i}(\hat{H}) \tag{3.14}$$

The set of hypothesized attacker types is given by Θ_A and θ_j is a type within the set. The posterior $P(\theta_j|\hat{H})$ is the probability of type θ_j given the history, $\hat{H}$. The set of actions available to the attacker is denoted by A_j and an action within the set is given by a_i. The mixed-strategy of the attacker is given by π_j and returns the probability of the attacker

selecting action a_i if he is type θ_j. The term $Q_s^{a_{D,i}}(\hat{H})$ is the expected cumulative reward for the defender after the occurrence of action profile $a_{D,i}$ in state s and is given by

$$Q_s^{a_{D,i}}(\hat{H}) = \sum_{s' \in \mathcal{S}} p(s'|s, a_{D,i}) \left[r_D(s, a_{D,i}, s') + \beta \max_{a_k \in A_D} E_{s'}^{a_k} \left(\hat{H}, a_{k,i}, s' \right) \right] \tag{3.15}$$

The transition function is given by $p(s, a_{D,i}, s')$ and defines the probability of transitioning from state s to state s' as a result of action profile $a_{D,i}$. The set of all states is denoted by $\mathcal{S}$. The immediate reward function of the defender is given by r_D and the discount factor, β, is applied to the expected cumulative reward.

Assuming the independence of types, we can define the posterior $P_j(\theta_j|H^t)$ from Equation 3.14 as

$$P(\theta_j|H^t) = \frac{L(H^t|\theta_j)P(\theta_j)}{\sum_{\hat{\theta}_j \in \Theta_j} L(H^t|\hat{\theta}_j)P(\hat{\theta}_j)} \tag{3.16}$$

where $L(H^t|\theta_j)$ is given by the product posterior

$$L(H^t|\theta_j) = \prod_{\tau=0}^{t-1} \pi_j(H^\tau, a_j^\tau, \theta_j) \tag{3.17}$$

Using this formulation for a pure type distribution, HBA will make correct predictions given sufficient time [4]. There are two caveats to this guarantee. The first caveat is that no types can be ruled out a priori, i.e., all types must have positive prior beliefs. The second caveat is that although HBA is guaranteed to make correct predictions, it is not guaranteed to learn the true type distribution. This is because some types may not be distinguishable by their strategies. An example of such a type distribution can be found in [4].

Using HBA can be computationally expensive for complicated SBGs. The computation time for implementing the recursive HBA algorithm is exponentially related to the number of types, actions, and states in the game [4]. One way to address this challenge is by implementing stochastic sampling methods in the HBA algorithm [5]. HBA with stochastic sampling is given by $a_D^t \sim \arg\max_{a_D} E_{s^t}^{a_D}(H^t)$, where

$$E_s^{a_D}(\hat{H}) = \frac{1}{n_W} \sum_{\theta_j \in \Theta_A} P(\theta_j|\hat{H}) \sum_{w_i \in W} Q_s^{a_{D,i}}(\hat{H}, w_i) \tag{3.18}$$

$$Q_s^{a_{D,i}}(\hat{H}, w_i) = r_D(s, a_{D,i}, s') + \beta E_{s'}^{a_k} \left(\hat{H}, a_{k,i}, s' \right) \tag{3.19}$$

A path is denoted by w_i, the set of all sampled paths is denoted by W, and the number of sampled paths is n_W. A path is the same structure as H; it is defined by a sequence of states and the action profiles occurring in those states. The states and actions used in Q are defined by the path. The values $P_j(\theta_j|H^t)$ and $L(H^t|\theta_j)$ are given by Equations 3.16 and 3.17, respectively.

3.4.3 Application of SBGs to Cybersecurity Decisions

In this work, we use an SBG to analyze a cybersecurity scenario for an NPP system. The players are a defender and an attacker. The defender is unsure about the characteristics of the attacker, and constructs a set of types to model the attacker's behavior. The defender makes cybersecurity decisions and the attacker chooses offensive actions to damage the NPP. As the players take action, the SBG transitions among a set of states that describe the state of the NPP and its ICS devices. Over time, the defender makes inferences about the attacker's characteristics using approximate Bayesian updating. Given his updated beliefs about the attacker's characteristics, the defender uses HBA to select the optimal cybersecurity strategy to maximize his cumulative utility,

Table 4: An example of the Harsanyi transformation.

Column Type 1

Probability $= p$

	Y	Z
W	a, A	b, B
X	c, C	d, D

Column Type 2

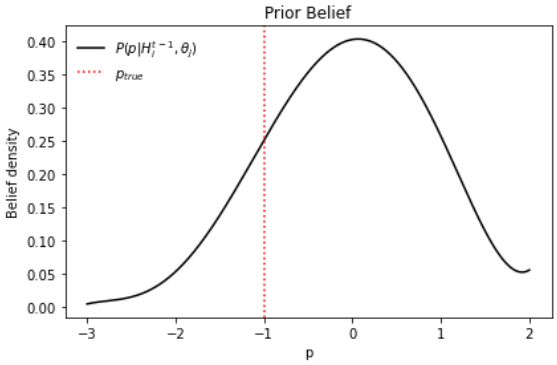
Probability $= 1 - p$

	Y	Z
W	e, E	f, F
X	g, G	h, H

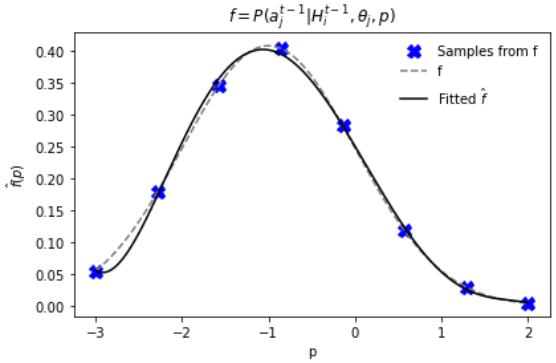
Harsanyi-Transformed Game

	YY	YZ	ZY	ZZ
W	$pa + (1-p)e,$	$pa + (1-p)f,$	$pb + (1-p)e,$	$pb + (1-p)f,$
	$pA + (1-p)E$	$pA + (1-p)F$	$pB + (1-p)E$	$pB + (1-p)F$
X	$pc + (1-p)g,$	$pc + (1-p)h,$	$pd + (1-p)g,$	$pd + (1-p)h,$
	$pC + (1-p)G$	$pC + (1-p)H$	$pD + (1-p)G$	$pD + (1-p)H$

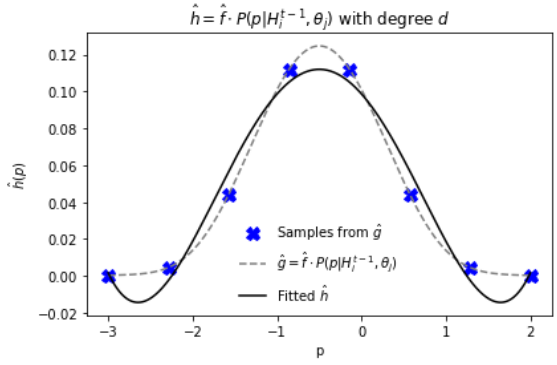
Figure 4: A simple stochastic security game.



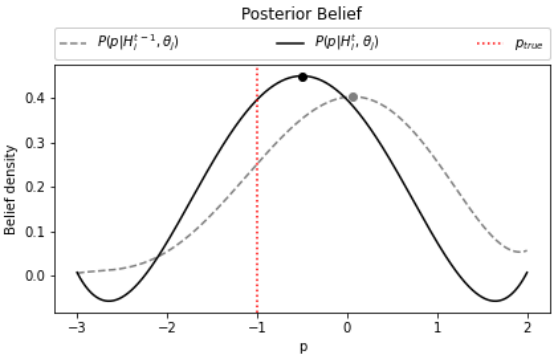
(a) Prior belief over p.



(b) Belief regarding the action as a function of p.



(c) Polynomial product of the prior and the action belief.



(d) The posterior belief over p.

Figure 5: Approximate Bayesian updating example.

4.0 Construction of the Stochastic Bayesian Game

Constructing an SBG is not a trivial task. In this chapter, we discuss several tools that can be used by cybersecurity teams to construct an SBG. These are existing tools in the fields of nuclear engineering, cybersecurity, and risk analysis, but their application to the construction of an SBG is new.

First, the system under consideration must be defined and understood. In this work, we study the residual heat removal system of a boiling water reactor. We present an overview of a boiling water reactor, the residual heat removal system, and their functional requirements.

Second, we identify the players who interact with the system. We define the plant defender and identify the priorities of the defender. We also characterize the threats against the NPP using Intel Corporation's Threat Agent Risk Assessment. This methodology is a tool used to identify the threat agents who pose the greatest risk to a computer system. The threat agents are modelled as attacker types in the SBG.

Third, we define the state space of the game. The states describe the environment for the players' interactions. To define the states we use System-Theoretic Process Analysis. System-Theoretic Process Analysis is a risk assessment tool that examines the functional interactions between components and the environment to prevent losses.

Fourth, we define the actions available to each player at each state in the game. For the defender, actions include defensive cybersecurity controls such as implementing a firewall or disabling a device's wireless capabilities. For the attacker, actions include cyber attacks such as malicious code injection via a USB drive. Each attacker type may have a different set of available actions at some or all of the states.

Fifth, we define the transition probabilities between the states. These probabilities are dependent on the actions chosen by the players. To define the transition probabilities, we use the NIST Common Vulnerability Scoring System and event tree analysis. The NIST Common Vulnerability Scoring System is used to quantify the risk posed by a given vulnerability. Event tree analysis is used to identify the probability of several outcomes given the occurrence of an initiating event such as a cyber attack.

Sixth, we define the utility functions for both players. The immediate utility functions define the rewards earned by the players after each time step of the game. Immediate utilities are a function of the actions chosen by both players and the state history. A cumulative utility function aggregates the immediate utilities over the course of the game.

Seventh, we define the decision algorithms used by the players. There are three components to the defender's decision-making: Bayesian learning of the attacker's parameters, estimation of the attacker's true type, and HBA to select an action. The attacker uses a single decision algorithm to select an action.

4.1 The Residual Heat Removal System

The SBG approach will be demonstrated on a subsystem of a boiling water reactor (BWR). A diagram of a BWR is shown in Figure 6. Coolant is pumped through the reactor vessel to remove heat from the nuclear fuel in the reactor core. The coolant is passed through moisture separators and the resulting steam is passed through a turbine to generate electricity. The steam is condensed by a heat sink such as a cooling tower and pumped back through the reactor vessel. Control rods are used to control the power of the core. The reactor vessel is surrounded by a containment building to prevent the release of radioactive material.

An SBG will be used to study the cybersecurity of the BWR's residual heat removal (RHR) system. The RHR system is used for both cooling and reactor vessel coolant inventory control. An overview of the RHR system is shown in Figure 7. The RHR system consists of two interconnected systems that are nearly identical. System I is on the left side of Figure 7 and System II is on the right. Motor-operated valves (MOVs) are implemented in both systems. MOVs that are normally closed are shaded and MOVs that are normally open are not shaded. Each system contains two pumps, a heat exchanger, and the necessary piping, valves, and instrumentation [83, 30].

The RHR has six operational modes, but for this case study we analyze the low pressure coolant injection (LPCI) mode [30]. The design goal of LPCI mode is to prevent the fuel

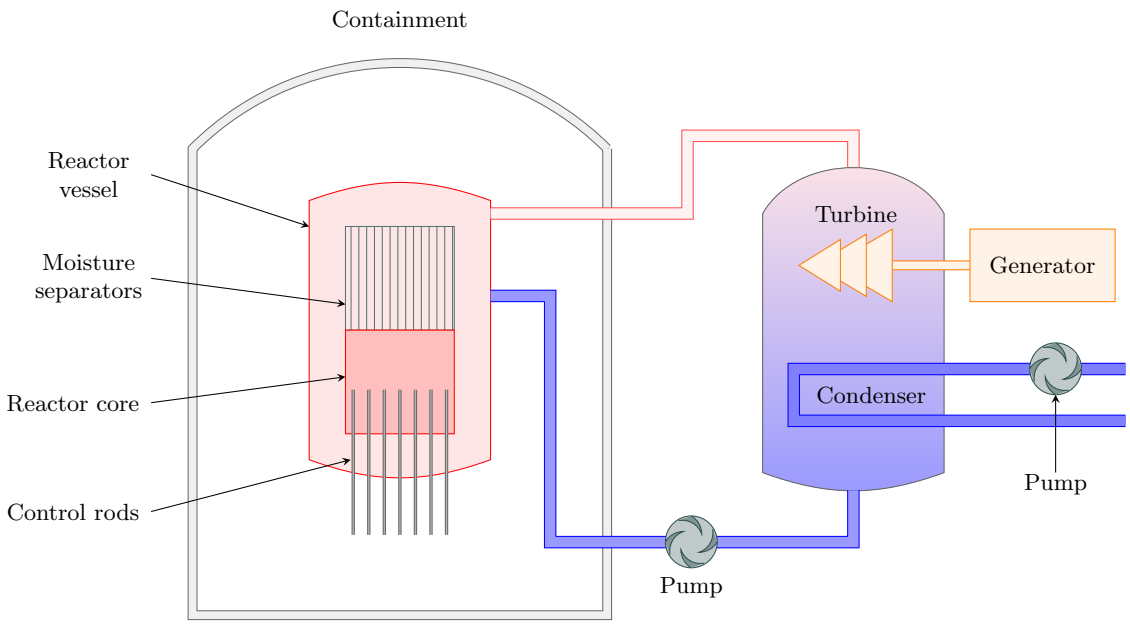


Figure 6: Boiling water reactor (BWR) overview.

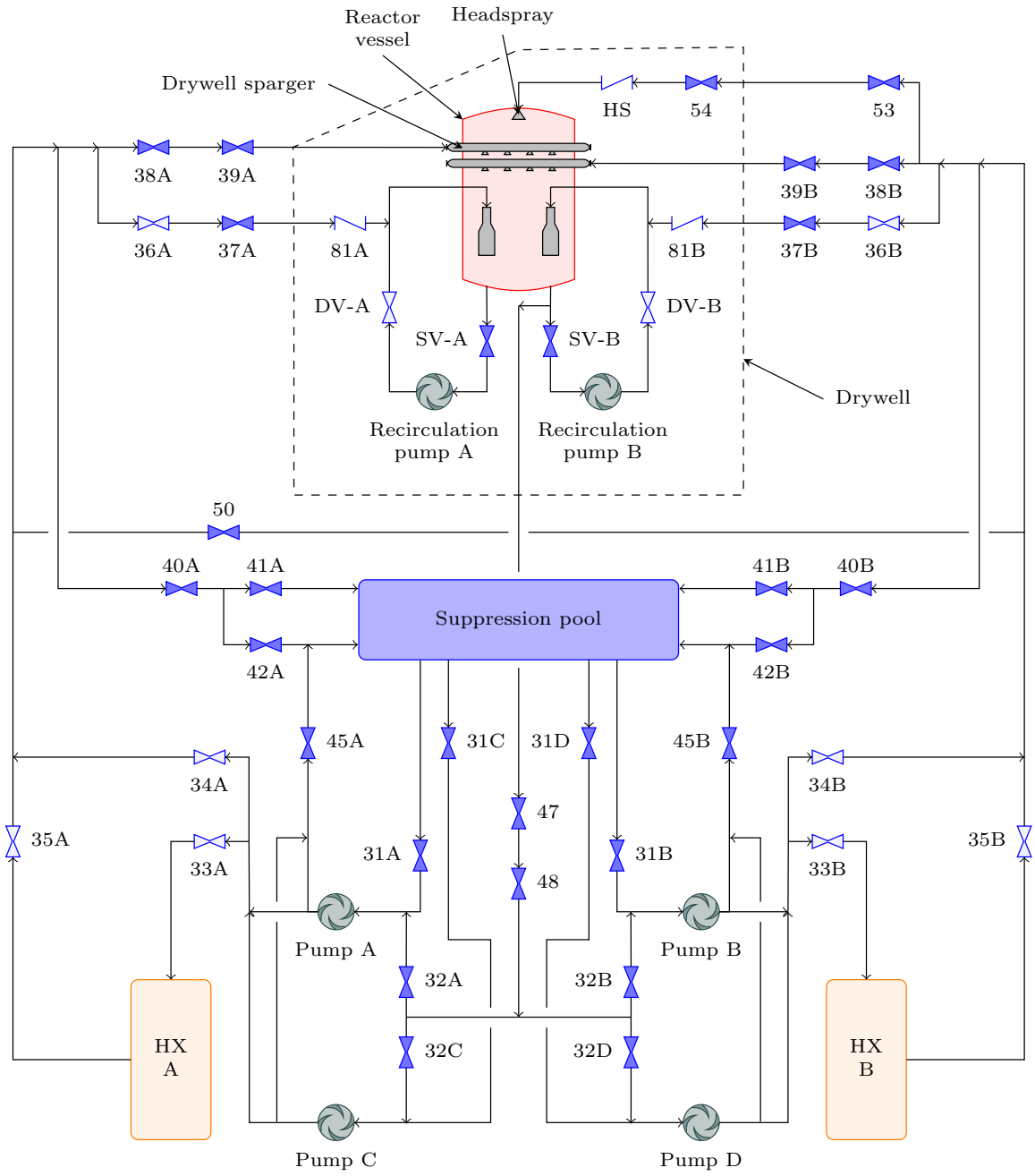


Figure 7: Residual heat removal (RHR) system.

cladding temperature from exceeding 1,204 °C during a loss-of-coolant accident (LOCA). Following a LOCA, LPCI mode cools the fuel by maintaining water level in the reactor vessel. RHR pumps draw water from the suppression pool and feed it to the reactor vessel through the recirculation system piping. Two out of four RHR pumps must inject after a design-basis LOCA to provide sufficient cooling [30].

LPCI mode can be initiated manually or automatically. LPCI mode is initiated automatically by one-out-of-two-twice logic for either low reactor water level or high drywell pressure. This means that water level and drywell pressure are each measured by four independent sensors. The four sensors are configured as two pairs. To automatically initiate LPCI mode, one sensor from each pair must measure the initiating signal [30].

The sequence of LPCI mode operations are:

1. LPCI mode initiation signal is generated by low reactor water level or high drywell pressure.
2. RHR pumps A, B, and C start two seconds after the LPCI initiation signal.
3. RHR pump D starts seven seconds after the LPCI initiation signal to prevent overloading of the bus.
4. The valves in the suction path between the RHR pumps and suppression pool are normally open, so no action is required. These valves are 31A, B, C, and D.
5. The containment spray and test isolation valves close so that the RHR pumps discharge to the recirculation system. These valves are 38A and B, 39A and B, 40A and B, 41A and B, and 42A and B.
6. Minimum flow valves close when LPCI injection flow exceeds a threshold. These valves are 45A and B.
7. The heat exchanger bypass valves open and cannot be closed for three minutes. These valves are 34A and B.
8. The LPCI injection valves open when the reactor vessel pressure drops below a threshold. These valves are 37A and B. Valves 36A and B are normally open, so no action is required.
9. The recirculation pump discharge valves close once reactor pressure has dropped below a threshold. These valves are DV-A and DV-B.
10. The LPCI system delivers water to the reactor vessel to cool the fuel.

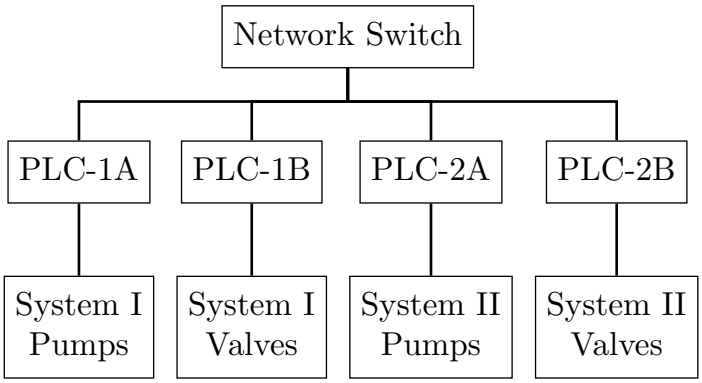


Figure 8: Network topology of the RHR system.

Hazards are conditions that can cause losses under certain circumstances. Hazards are discussed in greater detail during our discussion of System-Theoretic Process Analysis in Section 4.3.1. We consider the following hazards for the RHR system in LPCI mode.

H_1: Loss of flow path alignment capability to RHR subsystems

H_2: Damage to (or loss of) RHR pumps

H_3: Excessive removal of suppression pool inventory

H_4: Reactor trip

H_5: RHR does not initiate

H_6: Inadequate flow for intended operation

H_7: Cooling provided via RHR while reactor is at-power

The network topology used in this case study is shown in Figure 8. In this topology, the RHR system is controlled by four programmable logic controllers (PLCs) that communicate over a network switch. For each RHR system, one PLC controls the pumps and one PLC controls the valves.

To accomplish their goals, the threat agents could choose to target one or more of the devices shown in the RHR network topology. After deriving the goals and capabilities of the threat agents, the SBG approach will be used to select an optimal defense strategy.

4.2 The Players

The SBG is played by two players: a defender and an attacker. The defender is a multidisciplinary cybersecurity team at the NPP. Although the attacker is modelled as a single player, the construction of the SBG enables the analysis of the full spectrum of adversaries. The following sections describe each player in greater detail.

4.2.1 The Defender

The defender is a nuclear power plant cybersecurity team. The cybersecurity team should be multidisciplinary and be involved in the design of the SBG. Relevant fields include system theory, risk analysis, nuclear engineering, industrial control systems, security engineering, financial engineering, threat analysis, artificial intelligence, and game theory. Each of these fields provides essential input to the game-theoretic approach.

Expertise in threat analysis, cybsersecurity, and security engineering is useful in the definition of the attacker. NPPs face a variety of hostile threat agents and an understanding of the threats is necessary to define attacker types in the SBG. In this work, we use Intel Corporation's Threat Agent Risk Assessment to identify the threats to the NPP and to select the threat agents who pose the greatest risk to the NPP.

Expertise in system theory, industrial control systems, and risk analysis is useful in the construction of the stochastic state space. System-Theoretic Process Analysis will be used to define the states of the SBG. System-Theoretic Process Analysis is a risk assessment tool that examines the functional interactions between components and the environment to prevent damage to the system. This tool enables us to identify ways that the attacker might damage the NPP and allows us to construct a state space that includes those scenarios.

Expertise in risk analysis, nuclear engineering, and mechanical engineering is useful in the definition of state transition functions. Some state transition functions will be defined using event tree analysis. Event tree analysis defines the probability of several outcomes given the occurrence of an initiating event. In this case, the initiating event is a cyber attack and the outcome is damage to the NPP.

Expertise in industrial control systems, security engineering, and cybersecurity is useful in the definition of action sets for the players. At each state in the SBG, both players have a set of actions from which to choose. For the defender, these actions may include actions such as configuring firewalls or installing antivirus software. For the attacker, these actions can include eavesdropping attacks or modifying functions on programmable logic controllers. Expertise in these fields is necessary to understand what actions are available to the attacker and how we might defend against them.

Expertise in financial engineering, cybersecurity, nuclear engineering, and threat analysis is useful in the definition of utility functions for both players. As the players take action in the SBG, they earn immediate utilities that are a function of the action taken and the resulting state transition. Those immediate utilities are then aggregated by a cumulative utility function that guides the long-term behavior of the players throughout the game. Expertise in these fields is necessary to accurately quantify the costs and benefits of the players' actions.

Expertise in the fields of artificial intelligence and game theory is required to construct the SBG and optimize the defender's actions. The construction of the SBG involves elements of both stochastic and Bayesian game theory. Rather than use traditional Nash equilibrium solution methods, we use Harsanyi-Bellman ad hoc coordination. Harsanyi-Bellman ad hoc coordination is an artificial intelligence method that combines the Bayesian Nash equilibrium and Bellman optimal control to optimize the defender's actions as the game is played.

The goal of the defender is to maintain normal operation of the plant and to prevent the occurrence of losses. Losses are events that are unacceptable to plant stakeholders. Losses are discussed in greater detail during our discussion of System-Theoretic Process Analysis in Section 4.3.1. The defender's losses are:

L_1**: Loss of power generation** Since 1990, nuclear energy has generated approximately 20% of the net electricity generation in the United States [87]. To continue meeting this demand and to maintain a high capacity factor, outages at NPP must be minimized. Some power outages are necessary for the NPP to refuel and perform inspection and maintenance activities. Refueling occurs every 18 to 24 months. Planned outages normally occur in the fall and spring when there is lower demand for electricity [86].

Some guidelines have been published to optimize planned NPP outages [45]. Unplanned outages should be minimized. Both planned and unplanned outages have a significant financial impact on the NPP. Revenue is lost because the NPP is not generating power. The NPP must also buy power to replace the power that it would typically generate. The NPP also continues to pay employees and usually hires additional specialized workers to perform maintenance activities. A cyber attack that causes a loss of power generation at an NPP would have a severe financial impact.

L_2**: Environmental damage** Accidents at NPPs have the potential to cause significant environmental damage, despite defense in depth design. Defense in depth is a design approach that includes multiple redundant and independent safety features to prevent and mitigate accidents [95]. The main benefit of defense in depth approaches is that no one safety feature is solely responsible for ensuring NPP safety. For example, there are multiple layers of protection to prevent the release of radioactive materials from the NPP. These layers are the fuel matrix, the fuel cladding, the reactor vessel, and the containment system. Even with a defense in depth approach, it is possible for radioactive materials to be released during an accident. A cyber attack on an NPP that defeats defense in depth safety systems could cause similar environmental damages.

L_3**: Personnel injury or death** Personnel safety is highly valued by the nuclear power industry. Several organizations such as the Institute of Nuclear Power Operations, the U.S. Nuclear Regulatory Commission, and the International Atomic Energy Agency have published guidelines for developing a safety culture at NPPs [38, 74, 46, 42]. The injury or death of an NPP employee is a rare and significant accident. The industrial safety accident rate for the U.S. nuclear industry is shown in Figure 9 [64]. The injury or death of an NPP employee due to a cyber attack would be a significant event.

L_4**: Damaged public opinion** According to a 2019 poll, 49% of U.S. adults support nuclear power, 49% oppose nuclear power, and 47% believe nuclear power is safe [70]. Polls have shown that public opinion of nuclear energy can be negatively related to the occurrence of accidents at NPPs. Table 5 shows U.S. adults' support for constructing nuclear power plants in their area, before and after the Three Mile Island accident in March 1979 [72]. Although no deaths occurred as a result of the accident, public opinion

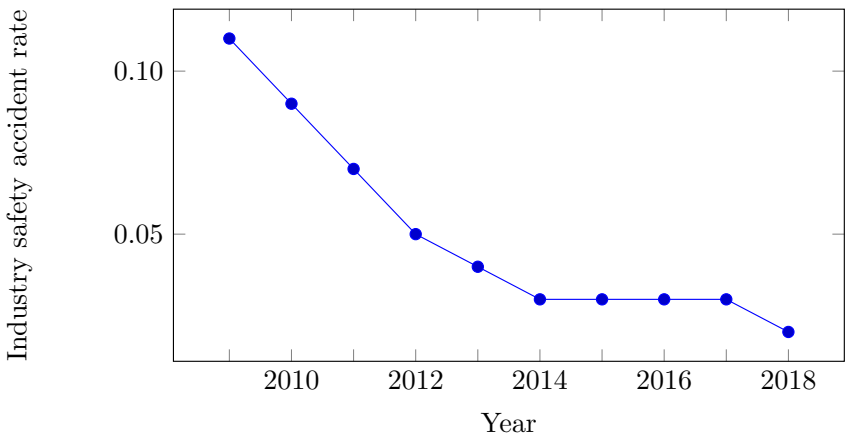


Figure 9: U.S. nuclear industry safety accident rate [64]. The industry safety accident rate is the number of accidents resulting in lost work, restricted work, or fatalities per 200,000 worker hours.

of nuclear energy worsened. A successful cyber attack on an NPP may have a similar affect on public opinion. Damaged public opinion could be costly to the nuclear power industry because public opinion is connected to government policy. Negative public opinion could lead to over-regulation or a reduction of the nuclear industry's share of total energy generation.

L_5**: Major equipment damage** Major equipment damage can have severe financial consequences for an NPP. Several NPP have shut down because of equipment repair costs [10]. For example, Crystal River 3 shut down in February 2013 after 36 years of operation because of the high cost of containment repairs. San Ofre 2 and San Ofre 3 shut down in June 2013 after 30 and 29 years of operation because of the high cost of replacing new steam generators. Equipment damage caused by a cyber attack could have a similar financial impact on an NPP.

L_6**: Core damage** Core damage is a serious event that can be coupled with the release of radioactive materials [43]. An example of an NPP accident causing core damage is the

Table 5: U.S. adults' support for constructing nuclear power plants in their area, before and after the Three Mile Island accident in March 1979 [72].

Opinion	Jun. 11-14, 1976	Apr. 6-9, 1979
Against	45%	60%
Not against	42%	33%
No opinion	13%	7%

Three Mile Island (TMI) accident [106, 82]. High fuel temperatures and oxidation of the fuel cladding caused much of the fuel in the core to melt. Because of severe core damage, the disassembling and defueling procedures had to be significantly modified [23]. Around 100,000 kg of damaged fuel had to be removed from the reactor vessel [106]. The clean-up of the damaged reactor station cost required more than 1,000 workers, took nearly 12 years, and cost approximately $973 million [106]. Core damage caused by a cyber attack could have a similar impact.

L_7**: Loss of sensitive data** The NPP is responsible for protecting various data from unauthorized disclosure. This data is pertinent to national security interests, the protection of radioactive materials, and other sensitive information. Requirements for controlling this data are specified in various federal regulations. Three types of data are protected by the U.S. Nuclear Regulatory Commission [94].

Classified information Classified information pertinent to NPP is usually one of two types. The first type is national security information (NSI). NSI is information that is classified by an executive order and must be protected to preserve national security. The second type is restricted data (RD). RD is information that is classified by the Atomic Energy Act [93] and must be protected to prevent the development or use of nuclear weapons. Requirements for access to this data and information safeguarding are given in 10 CFR §25 and 10 CFR §95 [98, 96].

Table 6: Losses and their assigned consequence magnitudes.

Loss	Description	Consequence ($)
L_1	Loss of power generation	-2×10^6
L_2	Environmental damages	-1×10^{11}
L_3	Personnel injury	-3×10^6
L_4	Damaged public opinion	-7×10^5
L_5	Major equipment damage	-4×10^7
L_6	Core damage	-1×10^8
L_7	Loss of sensitive data	-1×10^6

Safeguard information (SGI) This is a type of sensitive unclassified information required to be protected under Section 147 of the Atomic Energy Act [93]. It is pertinent to the physical protection of operating power reactors and other radioactive material. Performance requirements for the protection of SGI are specified in 10 CFR §73.21 [99], and specific requirements for the protection of safeguards information are specified in 10 CFR §73.22 and 10 CFR §73.23 [100, 101].

Sensitive unclassified non-safeguards information (SUNSI) This is a type of information that is not publicly available and is not related to nuclear safeguards. Some examples are personal and private information and proprietary information. Any information related to the protection or accounting of special nuclear material that is not designated NSI, RD, or SGI must be protected as specified in 10 CFR §2.390 [97].

The consequence magnitude for each loss is given in Table 6. This is a number that captures the impact of the loss to the defender. We define the unit of the loss magnitude to be dollars. In this work we assume that environmental damages and personnel injury are the most severe losses and we assume that damaged public opinion and loss of sensitive data are the least severe losses.

It is important to clearly define the defender's losses before defining the attacker types. Different attacker types may have different capacities to cause losses, and each may care about causing different losses. In the following section, we select the attacker types by considering their capacity and desire to cause losses for the NPP.

4.2.2 The Attacker

The attacker's types can be defined using a threat taxonomy. Several taxonomies have been created to describe cyber-threats. First, the International Atomic Energy Agency created a list of internal and external threats to nuclear facilities and defined their resources, time span of attack, tools used, and motivations [44]. Second, the United States Defense Science Board has created a tiered threat taxonomy system that describes threats in terms of their financial resources, skill, and potential impact [20]. Third, Intel Corporation compiled a Threat Agent Library (TAL) that defines 21 unique threat agents [39]. In this work, we use Intel's TAL.

After selecting a threat taxonomy, we must assess the risk each threat agent poses to the NPP. Each threat agent has specific motivations, goals, and capabilities defined by the TAL. Additionally, each defender type also has specific priorities regarding consequence prevention. Using this information, we estimate the risk posed by each threat agent to the NPP and each defender type, and identify the agents who pose the greatest risk for each type. In this work, we use Intel's Threat Agent Risk Assessment (TARA) methodology to select the critical threat agents. Each of the critical threat agents are modelled as a type in the SBG.

4.2.2.1 Threat Agent Library The hostile threat agents in Intel's TAL are given in Table 7. Each threat agent is defined by nine attributes: intent, access, outcome, limits, resource, skill level, objective, visibility, and motivation [39, 41].

Intent This attribute defines whether the threat agent is malicious.

Hostile An agent with hostile intent deliberately intends to harm the NPP. We only consider hostile threat agents in this work.

Non-hostile An agent with non-hostile intent accidentally harms the NPP. The non-hostile agents in the TAL are reckless employees, untrained employees, and information partners.

Access This attribute defines the scope of the threat agent's access to the NPP.

Internal These threat agents have internal access to the NPP.

External These threat agents have external access to the NPP.

Outcome This attribute defines the primary goal of the agent.

Theft An agent with the theft outcome seeks to steal assets from the NPP.

Business advantage An agent with the business advantage outcome seeks to develop a competitive advantage over the NPP in the marketplace.

Damage An agent with the damage outcome seeks to cause harm to NPP employees, NPP assets, or the general public.

Embarrassment An agent with the embarrassment outcome seeks to portray the NPP negatively to the public.

Technical advantage An agent with the technical advantage outcome seeks to develop technical capability using resources developed or implemented by the NPP.

Limits This attribute defines the limitations that constrain the threat agent.

Code of conduct An agent with code of conduct limits follows a code of conduct that exceeds relevant laws and statutes. This limit only applies to non-hostile threat agents, therefore it is not used in this work.

Legal An agent with legal limits follows relevant laws and statutes.

Extra-legal (minor) An agent with minor extra-legal limits may commit minor or non-violent crimes.

Extra-legal (major) An agent with major extra-legal limits may commit major crimes resulting in significant damage.

Resource This attribute defines the resources available for the threat agent to conduct an attack on the NPP.

Individual An agent with individual resources acts independently.

Club An agent with club resources is a part of a social group.

Contest An agent with contest resources is acting as a part of a short-term event with a particular goal.

Team An agent with team resources is part of a formally organized group.

Organization An agent with organization resources is a part of a group larger than a team with greater resources.

Government An agent with government resources has control over public assets and is highly resourced.

Skill level This attributes defines the abilities of the threat agent.

None An agent with no skill has no expertise regarding the NPP or cybersecurity.

Minimal An agent with minimal skill can use existing techniques to target the NPP.

Operational An agent with an operational skill level understands the technical domain of the NPP and and can create new attacks.

Adept An agent with an adept skill level is an expert in the technical domain of the NPP and can create sophisticated new attacks.

Objective This attribute defines the method by which the agent intends to achieve his outcome. An agent can have multiple objectives.

Copy This objective is to replicate an NPP asset.

Deny This objective is to prevent the use of an NPP asset.

Destroy This objective is to destroy an NPP asset, rendering it unusable.

Injure This objective is to damage an NPP asset, thereby reducing its functionality.

Take This objective is to steal an NPP asset.

Don't care The "don't care" objective means that the agent either makes opportunistic decisions or does not have a preference for which objectives are achieved. The cyber vandal, irrational individual, radical activist, and sensationalist were assigned the "don't care" objective in the TAL. Rather than creating a separate category of "don't care" for these agents, we represent them as having all five of the remaining objectives.

Visibility This attribute defines the agent's efforts to conceal his identity.

Overt An overt agent does not attempt to conceal the attack or his identity.

Covert A covert agent does not attempt to conceal the attack, but does attempt to conceal his identity.

Clandestine A clandestine agent attempts to conceal both the attack and his identity.

Don't care An agent who doesn't care either does not have a plan or does not have a preference regarding secrecy.

Motivation This attribute defines both the agent's cause for targeting the NPP and the agent's drive. Each agent has a defining motivation, but several agents also have a co-motivation, subordinate motivation, binding motivation, or personal motivation. In this work, we only consider the defining motivation for each threat agent.

Accidental An agent with this motivation does not intend to damage the NPP. This motivation applies to non-hostile threat agents and is not used in this work.

Coercion An agent with this motivation is manipulated to target the NPP on behalf of another party.

Disgruntlement An agent with this motivation seeks revenge against the NPP for prior perceived mistreatment.

Dominance An agent with this motivation seeks to establish superiority over the NPP.

Ideology An agent with this motivation targets the NPP to express a set of core beliefs.

Notoriety An agent with this motivation targets the NPP to achieve fame.

Organizational gain An agent with this motivation targets the NPP for the benefit of the agent's business or organization.

Personal financial gain An agent with this motivation targets the NPP for monetary compensation.

Personal satisfaction An agent with this motivation targets the NPP for personal pride or fulfillment.

Unpredictable An agent with this motivation targets the NPP without an identifiable cause or structure.

4.2.2.2 Threat Agent Risk Assessment We select the relevant threat agents from the TAL using Intel Corporation's Threat Agent Risk Assessment (TARA) methodology [40]. The TARA methodology is shown in Figure 10 and described below.

Table 7: Hostile agents from Intel Corporation's Threat Agent Library [39]

Threat Agent	Description
Anarchist	Individual who rejects structure
Civil activist	Non-violent supporter of a cause
Competitor	Business rival
Corrupt government official	Abuser of political power
Cyber vandal	Thrill-seeker without strong agenda
Data miner	External individual who gathers data
Disgruntled employee	Current or former employee with malicious intent
Government cyberwarrior	State-sponsored attacker with significant resources
Government spy	State-sponsored trusted insider
Internal spy	Insider gathering data for profit
Irrational individual	Individual without a rational plan
Legal adversary	Opponent in legal proceedings
Mobster	Participant in organized crime
Radical activist	Destructive supporter of a cause
Sensationalist	Fame-seeker
Terrorist	Person using violence to advance a socio-political agenda
Thief	Intends to steal for profit
Vendor	Business partner seeking competitive advantage

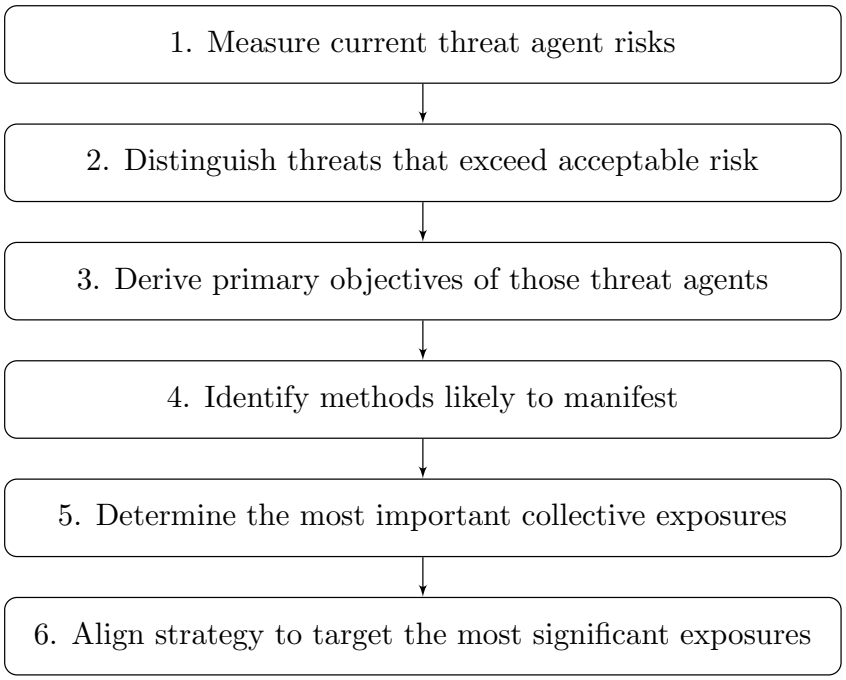


Figure 10: Intel Corporation's Threat Agent Risk Assessment [40].

The first step of TARA is to measure current threat agent risks to the NPP. This step is an initial assessment of the risk of each threat agent to the NPP. This assessment is based on expert opinion and is conducted for each threat agent in the TAL. TARA does not specify the method to determine each agent's risk. To determine a threat agent's risk, we must first determine which losses are desired by the agent. To determine each agent's desired losses, we consider their motivations, limits, visibility, and desired outcomes. Table 8 summarizes this process for each threat agent.

TARA does not specify the method to determine each agent's risk. To calculate the risk posed by a threat agent, we use the traditional definition of risk as the product of consequence and likelihood. The total risk of a threat agent is given by equation 4.1.

$$\text{Risk} = \sum_{\text{Losses}} \text{Consequence} \times \text{Likelihood} \tag{4.1}$$

The consequences are the defender's losses. The assumed loss consequences used in this work are given in Table 6.

The likelihood is the probability that the threat agent in question could cause that loss and is based on expert opinion. To estimate these likelihoods, we consider the threat agent's access, resources, skill level, access, and visibility. Agents with internal access, organization and government resources, operational and adept skill level, and overt visibility receive the greatest likelihoods. The risk posed by each threat agent is calculated using equation 4.1, the estimated likelihoods, and loss magnitudes given in Table 6. Table 9 summarizes this process for each threat agent.

An asterisk superscript in Table 9 indicates that the parameter differs from the original definition in the TAL. The disgruntled employee, government cyberwarrior, and irrational individual were assigned visibility of "Multiple/Don't Care" in the TAL. For simplicity, we have assigned one visibility category for each agent. The disgruntled employee and government cyberwarrior are assigned a visibility of "Covert" and the irrational individual is assigned a visibility of "Overt". In this work, the club and contest resources are consolidated into one club category. This change only affects the cyber vandal.

The second step of TARA is to distinguish threat agents that exceed baseline acceptable risks. First, an acceptable risk baseline must be defined for the NPP. This baseline is based on

Table 8: Threat agents, their motivations, limits, desired outcomes, and desired NPP losses.

Threat Agent	Motivation	Limits	Theft	Bus. Advantage	Damage	Embarrassment	Tech. advantage	L_1: Loss of power	L_2: Env. Damage	L_3: Injury/death	L_4: Public opinion	L_5: Equip. damage	L_6: Core damage	L_7: Data loss
Anarchist	Ideology	Extra-legal, major			X			X	X	X	X	X	X	
Civil activist	Ideology	Extra-legal, minor				X		X			X			X
Competitor	Org. gain	Extra-legal, minor		X			X	X			X			X
Corrupt gov. official	Financial gain	Extra-legal, minor		X			X	X			X			X
Cyber vandal	Dominance	Extra-legal, minor			X			X			X	X	X	X
Data miner	Org. gain	Extra-legal, minor		X			X				X			X
Disgruntled employee	Disgruntlement	Extra-legal, major			X	X		X	X	X	X	X	X	X
Gov. cyberwarrior	Dominance	Extra-legal, major			X	X		X	X	X	X	X	X	X
Gov. spy	Ideology	Extra-legal, major		X			X	X			X	X	X	X
Internal spy	Financial gain	Extra-legal, minor	X				X							X
Irrational individual	Unpredictable	Extra-legal, major			X	X		X	X	X	X	X	X	X
Legal adversary	Dominance	Legal	X		X						X			X
Mobster	Org. gain	Extra-legal, major	X											X
Radical activist	Ideology	Extra-legal, minor			X	X		X	X	X	X	X	X	X
Sensationalist	Notoriety	Extra-legal, minor			X	X		X	X	X	X	X	X	X
Terrorist	Ideology	Extra-legal, major			X			X	X	X	X	X	X	X
Thief	Financial gain	Extra-legal, minor	X											X
Vendor	Org. gain	Legal	X			X					X			X

Table 9: Threat agents, their access, resources, skill levels, visibility, estimated loss likelihoods, and risk.

Threat Agent	Access	Resources	Skill	Visibility	L_1: Loss of power	L_2: Env. Damage	L_3: Injury/death	L_4: Public opinion	L_5: Equip. damage	L_6: Core damage	L_7: Data loss	Risk ($)
Anarchist	Extern.	Club	None	Overt	0.000	0.000	0.000	0.010	0.000	0.000	—	7.00×10^3
Civil activist	Extern.	Org.	Adept	Covert	0.006	—	—	0.700	—	—	0.010	5.12×10^5
Competitor	Extern.	Org.	Adept	Clandestine	0.001	—	—	0.600	—	—	0.009	4.31×10^5
Corrupt gov. official	Extern.	Gov.	Adept	Overt	0.010	—	—	0.950	—	—	0.015	7.00×10^5
Cyber vandal	Extern.	Club*	Oper.	Covert	0.001	—	—	0.030	0.001	0.001	0.005	1.17×10^5
Data miner	Extern.	Team	Adept	Clandestine	—	—	—	0.400	—	—	0.008	2.88×10^5
Disgruntled employee	Intern.	Indiv.	Oper.	Covert*	0.005	0.005	0.100	0.200	0.100	0.005	0.700	5.06×10^8
Gov. cyberwarrior	Extern.	Gov.	Adept	Covert*	0.007	0.005	0.007	0.900	0.007	0.005	0.012	5.01×10^8
Gov. spy	Intern.	Gov.	Adept	Clandestine	0.002	—	—	0.850	0.020	0.005	0.800	2.70×10^6
Internal spy	Intern.	Org.	Adept	Clandestine	—	—	—	—	—	—	0.750	7.50×10^5
Irrational individual	Extern.	Indiv.	None	Overt*	0.000	0.000	0.000	0.010	0.000	0.000	0.000	7.00×10^3
Legal adversary	Extern.	Org.	Adept	Overt	—	—	—	0.850	—	—	0.010	6.05×10^5
Mobster	Extern.	Org.	Adept	Covert	—	—	—	—	—	—	0.010	9.50×10^3
Radical activist	Extern.	Org.	Adept	Overt	0.009	0.007	0.007	0.850	0.007	0.005	0.010	7.01×10^8
Sensationalist	Extern.	Club	Minimal	Overt	0.000	0.000	0.001	0.015	0.001	0.000	0.004	3.60×10^4
Terrorist	Extern.	Org.	Adept	Covert	0.006	0.005	0.006	0.700	0.006	0.004	0.010	5.01×10^8
Thief	Intern.	Indiv.	None	Clandestine	—	—	—	—	—	—	0.050	5.00×10^4
Vendor	Intern.	Team	Oper.	Clandestine	—	—	—	0.350	—	—	0.750	9.95×10^5

expert opinion and threat intelligence. One way to determine the acceptable risk baseline is to consider only the loss with the greatest consequence. Next, identify the maximum allowable probability for that loss. Because the worst loss was assigned a consequence magnitude of one, the acceptable risk baseline is then the product of the loss consequence and maximum allowable probability.

Alternative approaches to distinguishing the high-risk threat agents involve measuring their relative risk. The preferable selection method is defining a risk threshold, but these methods can be used if a risk threshold cannot be determined. Some examples are:

- Select the threat agents with risk greater than a given percentage of the risk of the most significant threat agent.
- Calculate the total risk posed by all threat agents. Select the threat agents with the highest scores that make up a given percentage of the total risk.
- Approximate the total risk posed by all threat agents as the sum of the risk posed by all agents except the least significant agent. Calculate the percent error between the approximated total risk and the total risk including all threat agents. If the percent error is below a given threshold, repeat for the next least significant threat agent and calculate the percent error between the new approximated total and the total from the previous iteration. Repeat until the percent error exceeds the threshold.

The risk of each threat agent is plotted in Figure 11. In this case, one group of threat agents clearly poses the greatest risk. The risks of the radical activist, disgruntled employee, government cyberwarrior, and terrorist exceed the next threat agent by approximately three orders of magnitude. An acceptable risk baseline between $\$2.7 \times 10^6$ and $\$5.0 \times 10^8$ would result in the selection of these threat agents.

Although the four threat agents have varying motivations, they all seek the outcome of causing damage, and three of the four also seek to cause embarrassment. Three of the threat agents also have major extra-legal limits. This combination of outcomes and limitations leads all four of the selected threat agents to desire all of the defender's losses. This alone does not lead to their high risk levels. The skill and resources of these threat agents are also significant. The radical activist, government cyberwarrior, and terrorist all have adept skill

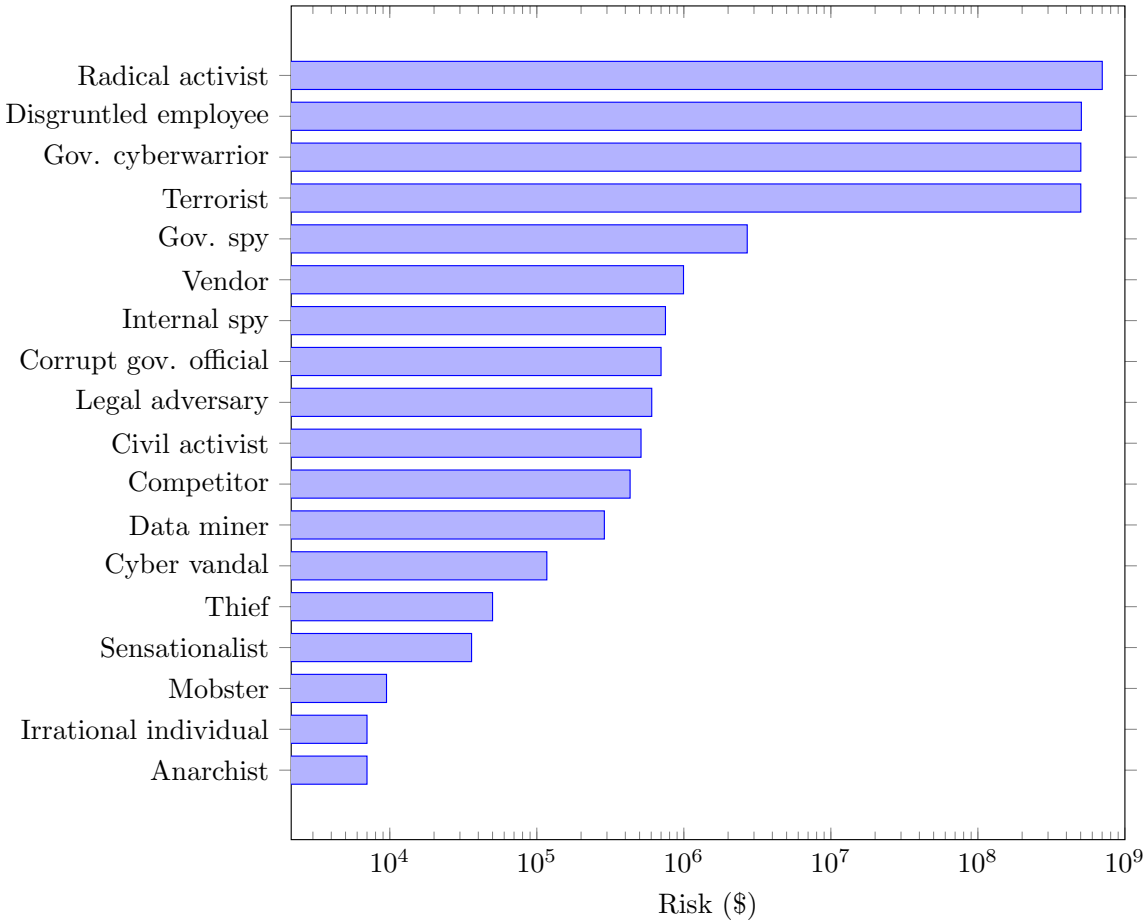


Figure 11: Threat agent risks on logarithmic scale.

levels, and at least organization resources. In comparison, the disgruntled employee has an operational skill level and individual resources, but still poses a significant risk because of his internal access.

The attacker's types and their loss magnitudes are given in Table 10. All of the attacker's types seek to cause each of the NPP losses, so all of the attacker's loss magnitudes are greater than zero. Note that the attacker's loss magnitudes do not need to be directly correlated with the defender's loss magnitudes. We have assumed these values based on the motivations and desired outcomes of the threat agents as defined by the TAL. These values are assumed for demonstration purposes. In practice, additional threat intelligence should also be used.

The third step of TARA is to derive primary objectives of those threat agents. The primary objectives are a combination of the threat agent's motivations and capabilities that are defined by the TAL. This step was completed in the first step when we identified each threat agent's desired losses in Table 8.

The fourth step of TARA is to identify methods likely to manifest. A method is a combination of threat agent objectives and the means by which the threat seeks to accomplish them. These methods are used to define the actions available to the threat agents at each state in the SBG. This step is not necessary for the selection of attacker types for the SBG, but will be done in conjunction with System Theoretic Process Analysis as the states of the SBG are defined.

The fifth step of TARA is to determine the most important collective exposures. The exposures are the intersection of attack vectors and the methods likely to manifest. This step is not necessary for the selection of attacker types for the SBG, but it is mentioned because its purpose is addressed later in game construction using System Theoretic Process Analysis.

The sixth step of TARA is to align strategy to target the most significant exposures. This is the decision-making portion of TARA. This step is not necessary for the selection of attacker types, but it is mentioned because its purpose is addressed by applying Harsanyi-Bellman ad hoc coordination to the SBG.

Each selected threat agent is modelled as a type in the SBG. Recall that a type is a construction in a Bayesian game that represents a player's belief about the parameters

Table 10: The attacker's types and corresponding loss magnitudes.

Type	Name	$\lvert L_1 \rvert$ ($)	$\lvert L_2 \rvert$ ($)	$\lvert L_3 \rvert$ ($)	$\lvert L_4 \rvert$ ($)	$\lvert L_5 \rvert$ ($)	$\lvert L_6 \rvert$ ($)	$\lvert L_7 \rvert$ ($)
θ_A^1	Radical activist	3×10^9	2×10^6	2×10^6	5×10^{10}	4×10^7	5×10^8	3×10^7
θ_A^2	Disgruntled employee	3×10^{10}	2×10^4	3×10^6	5×10^8	3×10^9	3×10^{10}	2×10^8
θ_A^3	Gov. cyberwarrior	3×10^{12}	1×10^6	1×10^6	1×10^6	1×10^9	2×10^{11}	5×10^7
θ_A^4	Terrorist	2×10^9	5×10^{11}	7×10^7	1×10^6	1×10^7	3×10^8	1×10^6

governing another player. The types characterize the attributes of the threat as defined by the TAL. At each state in the SBG, both players choose from a set of actions. For the defender, these strategies include sets of cyber components to be defended, and the means by which they are defended. For the attacker, these actions include sets of cyber components to be attacked, and the means by which they are attacked. The actions available to the players are defined using System Theoretic Process Analysis in Section 4.3. As a result of the actions that are selected and the resulting state transition, each player receives an immediate reward. The immediate rewards are aggregated over the course of the game by a cumulative reward function. The reward functions of each type are a function of the threat agent's resources, capabilities, and motivations. The reward functions for each type are defined in Section 4.6.

In this section, we identified the players of the SBG. The defender's capabilities were defined and the defender's losses were identified and prioritized. Portions of Intel Corporation's TARA methodology were used to define the attacker types. The threat agent attributes defined in the TAL were used to identify losses of interest to each agent and to calculate the risk posed to the NPP by each agent.

The disgruntled employee, government spy, radical activist, government cyberwarrior, terrorist, and corrupt government official were as the critical threat agents. The majority of these threat agents seek to cause all of the potential losses to the system, thus increasing their estimated risk. The majority of these threat agents have major extra-legal limits, organization or government resources, and adept skill level. The visibility of the agents varies. It is also noteworthy that the two agents with the greatest risk have internal access to the NPP. This resulted in a greater estimated probability of those agents to successfully cause a loss. Each of the selected threat agents are modelled as types in the SBG.

4.2.3 The Type Distribution

Now that the attacker's types have been defined, we must define their probability distributions. In practice, these distributions should be estimated from threat intelligence data. Data of importance include the global political climate, known threat agent activity, insider information, and recent threats to the NPP.

Table 11: The probability distributions of the types.

θ_A^j	θ_A^1	θ_A^2	θ_A^3	θ_A^4
$p(\theta_A^j)$	0.15	0.35	0.30	0.20

For this case study, we assume the type distributions in Table 11. We assume the defender assigns the greatest probability to the disgruntled employee type, followed by the government cyberwarrior, terrorist, and radical activist.

This distribution is the defender's initial estimate of the probability distribution over the attacker's types. As the SBG is played, the defender can use Bayesian learning to update the type distribution. The updated distribution is then used in HBA to select the optimal cybersecurity strategy given the defender's beliefs about the attacker.

4.3 Stochastic State Space

In an SBG, the states define the setting of the players' interactions. The stochastic state space must characterize the plant over the operating range of consideration and throughout the course of all postulated attacks. For complex systems with many digital devices, the unaltered state space may be too large for the security game to be tractable. For example, consider a system with 25 components, each of which has two operational states: operational or nonoperational. For this system, there would be a minimum of $2^{25} = 33,554,432$ states corresponding to each possible combination of the individual components' operational states. A large state space presents challenges both in computation and in interpretation of results. Thus, a method is needed to constrain the size of the stochastic state space.

4.3.1 System-Theoretic Process Analysis

System-theoretic process analysis (STPA) can be used to manage the size of the stochastic state space. STPA is a hazard analysis technique that considers both component failures and unsafe interactions of system components to model accident causation [52]. STPA has been applied to study a variety of cyber-physical systems including offshore supply vessel positioning systems [2], buoy tender control systems [77], and railroad systems [21]. STPA has also been applied to nuclear systems as part of Hazards and Consequences Analysis of Digital Systems (HAZCADS) [24].

The application of STPA to the construction of an SBG is summarized in the following four steps.

Step 1: Define the Purpose of Analysis This step involves three sub-tasks: identifying losses, identifying system-level hazards, and identifying system-level constraints.

Losses are consequences that are unacceptable to stakeholders. These losses may be aligned with the goals of the attacker. Losses may involve environmental conditions that cannot be controlled. We have already defined the losses in Section 4.2.1.

Hazards are system states that will lead to at least one loss if a specific set of environmental conditions are met. A system boundary should be defined to separate the system from its environment. The system is typically defined to include all aspects over which the system designers can exert control. Hazards specify an overall system state and do not refer to specific system components. The RHR hazards that we will consider in this work are listed in our description of the RHR system in Section 4.1.

Constraints are conditions that must be met to prevent hazards. Each constraint must be traceable to at least one hazard. Constraints are often the inverse of the hazard. An example of a constraint for a nuclear power plant is to initiate the RHR system at the correct time. This constraint prevents the hazard of the RHR failing to initiate. The other RHR constraints follow a similar structure. For brevity, we do not list all of the RHR constraints here.

Step 2: Model the control structure The control structure is a functional model of the interactions of the controllers with the controlled process through feedback and control

Table 12: Hacked devices and potential hazards.

	PLC-1A	PLC-1B	PLC-2A	PLC-2B	Switch	PLC-1A & PLC-1B	PLC-2A & PLC-2B
H_1		X		X	X		
H_2	X	X	X	X	X		
H_3	X		X		X		
H_4					X	X	X
H_5	X		X		X		
H_6	X	X	X	X	X		
H_7					X	X	X

actions. In this case, the network topology in Figure 8 is an adequate model of the control structure. A more detailed model may be required for more complex control structures.

Step 3: Identify the unsafe control actions Unsafe control actions (UCAs) are control actions that can lead to a hazard under certain conditions. A UCA describes the source of the control action, the type of control action, and the context of the control action. STPA is used primarily as a risk analysis tool to study unintended malfunctions of digital control systems. In contrast, we are applying STPA to study the deliberate attack of a digital control system by a malicious actor. We recognize that the purpose of defining UCAs is to understand the events that may lead to hazards. To meet this goal, we list the sets of penetrated devices that may cause hazards under certain conditions.

Table 12 shows which hazards can occur as a consequence of different combinations of hacked devices. We do not list every action profile leading to each hazard because there are multiple ways some devices can be hacked. For example, loss of flow path alignment (H_1) may occur if the defender does not disable wireless on PLC-1B, and the attacker chooses to conduct a wireless exploit on PLC-1B. This action profile reflects the attacker's capability to cause a failure, but does not specify the specific method used to cause the failure. For example, an attacker with administrator privileges on PLC-1B could choose to open or close different combinations of valves in System I to cause H_1.

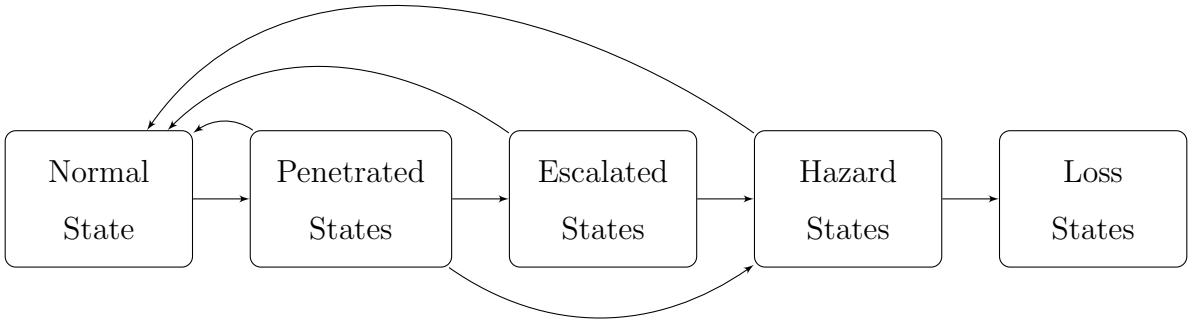


Figure 12: Generalized state space of an SBG.

Step 4: Identify loss scenarios. A loss scenario summarizes the events that can lead to hazards. These scenarios capture the evolution of NPP operation from a normal state to a loss state. The stochastic state space is developed based on the loss scenarios.

The generalized stochastic state space is shown in Figure 12. The game originates in a normal state where the plant is operating as intended. From the normal state, the attacker may be able to breach the plant's defenses and cause the game to enter a penetrated state. From a penetrated state, the attacker may be able to cause a hazard, or the attacker may need to escalate his privileges within the plant before causing a hazard. If certain environmental conditions occur, the game may transition from a hazard state to a loss state. Most losses are represented by absorbing states; when they are reached, the game concludes. Some losses may not necessitate the end of the game, in which case, they are not assigned a separate state. At all of the non-absorbing states, the defender may be able to cause the game to transition back to the normal state.

There are a total of 45 states in the SBG. A full list of states is given in Table 13. We use one normal state to represent standard operation of the RHR system. There are 31 penetrated states where the attacker has compromised various sets of ICS devices. There are 31 penetrated states because there are five ICS devices and we assume that each device is either penetrated or operating normally ($2^5 - 1 = 31$). There are no escalated states in this

game. There are seven hazard states representing the hazards listed in Section 4.1. There are six absorbing loss states. Loss of sensitive data (L_7) is considered to be a non-terminal loss, so it are not designated its own states.

4.4 The Actions

Defining the players' actions requires an understanding of the industrial control system devices, the RHR system, and the players. An understanding of the control system devices is required to identify device vulnerabilities, malicious actions that can exploit those vulnerabilities, and cybersecurity control actions that can address those vulnerabilities. Knowledge of the RHR system is needed to understand how exploits can lead to negative consequences in the system. Finally, knowledge about the players is needed to understand their capabilities and goals. Many considerations about the players have been addressed using TARA and the type definitions. This sections lists the actions available to the players at each state in the SBG.

4.4.1 Normal and Penetrated States

The attacker and defender each have several choices to make regarding each component in the RHR system. For the defender, these choices address the configurations of the industrial control system devices. For the attacker, these choices address the attack vector for circumventing the defender's cybersecurity controls. These choices are available to the players in the normal and penetrated states. Each choice comes with a cost for that player. The defender's choices are given in Table 14 and the attacker's choices are given in Table 15.

We assume that the default configuration of each PLC is that authentication is off and wireless communication is enabled. The defender can choose to enable authentication on each PLC and can choose to disable wireless communication on each PLC. If the defender has properly enabled authentication, the attacker will not be able to connect to the PLC. If the defender has properly disabled wireless, the attacker will not be able to conduct the

Table 13: States in the SBG.

		Penetrated Devices				
Index	Description	PLC-1A	PLC-1B	PLC-2A	PLC-2B	Switch
0	Normal					
1	Penetrated					X
2	Penetrated				X	
3	Penetrated				X	X
4	Penetrated			X		
5	Penetrated			X		X
6	Penetrated			X	X	
7	Penetrated			X	X	X
8	Penetrated		X			
9	Penetrated		X			X
10	Penetrated		X		X	
11	Penetrated		X		X	X
12	Penetrated		X	X		
13	Penetrated		X	X		X
14	Penetrated		X	X	X	
15	Penetrated		X	X	X	X
16	Penetrated	X				
17	Penetrated	X				X
18	Penetrated	X			X	
19	Penetrated	X			X	X
20	Penetrated	X		X		
21	Penetrated	X		X		X
22	Penetrated	X		X	X	
23	Penetrated	X		X	X	X
24	Penetrated	X	X			
25	Penetrated	X	X			X
26	Penetrated	X	X		X	
27	Penetrated	X	X		X	X
28	Penetrated	X	X	X		
29	Penetrated	X	X	X		X
30	Penetrated	X	X	X	X	
31	Penetrated	X	X	X	X	X
32	Hazard: H_1	—	—	—	—	—
33	Hazard: H_2	—	—	—	—	—
34	Hazard: H_3	—	—	—	—	—
35	Hazard: H_4	—	—	—	—	—
36	Hazard: H_5	—	—	—	—	—
37	Hazard: H_6	—	—	—	—	—
38	Hazard: H_7	—	—	—	—	—
39	Loss: L_1	—	—	—	—	—
40	Loss: L_2	—	—	—	—	—
41	Loss: L_3	—	—	—	—	—
42	Loss: L_4	—	—	—	—	—
43	Loss: L_5	—	—	—	—	—
44	Loss: L_6	—	—	—	—	—

Table 14: The cybersecurity choices available to the defender in the normal and penetrated states.

PLCs		**Switch**	
Action	Cost ($)	Action	Cost ($)
Enable authentication	3×10^3	Enable authentication	3×10^3
Disable wireless	2×10^2	Enable firewall	1×10^5
		Access control	6×10^5

Table 15: The cybersecurity choices available to the attacker in the normal and penetrated states.

	PLCs			
Action	θ_A^1 Cost ($)	θ_A^2 Cost ($)	θ_A^3 Cost ($)	θ_A^4 Cost ($)
Connect	6.0×10^5	1.0×10^2	4.0×10^4	1.0×10^5
Wireless exploit	8.0×10^7	7.0×10^3	3.0×10^7	4.0×10^7

	Switch			
Action	θ_A^1 Cost ($)	θ_A^2 Cost ($)	θ_A^3 Cost ($)	θ_A^4 Cost ($)
Connect	6.0×10^5	1.0×10^2	4.0×10^4	1.0×10^5
Cyber attack	4.0×10^4	2.0×10^5	2.0×10^4	3.0×10^4
Physical attack	2.5×10^6	5.0×10^3	1.5×10^6	2.5×10^6

wireless exploit. The attacker requires local access for the wireless exploit, but does not require local access to connect with an unsecured PLC. The defender can also choose to implement access control to restrict who is able to physically access the PLC. If the defender has properly implemented access control, only approved personnel can access the PLC.

We assume that the default configuration of the switch is that authentication is off, the firewall is off, and there is no access control. If the defender has properly enabled authentication, the attacker will not be able to connect to the PLC. The defender can choose to enable the firewall. In practice, there are many possibilities for firewall configuration, but here we assume a binary decision to either enable or not enable the firewall. The attacker can choose whether to attempt an attack. If the defender has enabled the firewall, the attacker will not be able to conduct the attack. The defender can also choose to implement access control to restrict who is able to physically access the switch. If the defender has properly implemented access control, only approved personnel can access the switch.

A complete action for a player consists of selecting an option for each available choice. This is the most secure action for the defender in the normal state or a penetrated state:

- enable authentication on all of the PLCs
- disable wireless on all of the PLCs
- enable authentication on the switch
- enable the firewall on the switch
- implement access control for the switch

Note that the players are not constrained to make the same choice for all of the PLCs. For example, the defender could choose to enable authentication on PLC-1A and not enable authentication on the other PLCs.

In the penetrated states, the attacker and defender also have choices to make that are not directly related to individual ICS devices. These decisions affect whether the attack regresses from a penetrated state to the normal state, or whether the attack progresses to a hazard state.

Penetration recovery In all penetrated states, the defender may attempt to initiate recovery action. If the recovery is successful, the attacker is expunged from the system

and the game returns to the normal state. For simplicity, we assume a flat cost of $\$1 \times 10^6$ for recovery from all penetrated states, regardless of the number of devices that have been penetrated or the manner by which they have been penetrated. This is based on the assumption that during a known cyber attack, a thorough investigation would be conducted to examine all potentially affected NPP devices.

Hazard initiation In some penetrated states, the attacker may choose to cause a hazard to occur. The hazard options are dependent on the devices that are penetrated in that particular state, as shown in Table 12. We assume that there is no additional cost to the attacker for allowing a hazard to occur. For simplicity, we assume that the attacker can only select one hazard at a time.

The defender and attacker both have the option to abstain from any action. We assume that there is no cost to either player to abstain from action.

4.4.2 Hazard States

In the hazard states, the attacker and defender have choices to make that are not directly related to individual ICS devices. These decisions affect whether the attack regresses from a hazard state to the normal state, or whether the attack progresses to a loss state.

Hazard recovery In all hazard states, the defender may attempt to initiate recovery action. If the recovery is successful, the attacker is expunged from the system and the game returns to the normal state. For simplicity, we assume a flat cost of $\$6 \times 10^6$ for recovery from all hazard states.

Loss initiation In all hazard states, the attacker may choose to allow a loss to occur. The potential losses corresponding to each hazard are discussed in greater detail in Section 4.5. We assume that there is no additional cost to the attacker for allowing a loss to occur. Unlike the hazard initiation action, we assume that the attacker is only able to choose whether or not to allow a loss to occur — the attacker is not able to choose the specific loss. This is because the occurrence of a loss is dependent on other environmental conditions and safety systems beyond the attacker's control.

4.4.3 Loss States

Losses that are modelled as states in the SBG have the most severe consequences. These loss states are absorbing states. The game ends when an absorbing state is reached, therefore there are no actions available to either of the players in any of the loss states.

4.5 State Transitions

Now that the states and actions have been defined, we can define the state transition function. The state transition function is a discrete probability distribution over all of the states in the SBG. The transition function is dependent on the originating state, the action chosen by the defender, and the action chosen by the attacker.

We consider four types of transitions in this research: penetration transitions, hazard transitions, loss transitions, and recovery transitions. Penetration transitions describe the probability of the attacker breaching NPP defenses to gain access to the system. Hazard transitions describe the probability of the attacker causing a hazard given his level of access to the plant. Loss transitions describe the probability that a hazard causes a loss, and accounts for environmental factors beyond the players' control. Finally, recovery transitions describe the probability that the defender returns the game to the normal state from a penetrated or hazard state.

4.5.1 Penetration Transitions

Penetration transition describe the probability of the attacker penetrating the system, given the attacker and defender's actions. Penetration transitions are estimated using the TAL and the exploitability metrics defined by the Common Vulnerability Scoring System [28]. There are four CVSS metrics:

1. Attack Vector: The attack vector metric measures the context required for an attacker to exploit a vulnerability. The metric can be assigned values of "Physical" if the attacker needs to physically interact with the component, "Local" if the the component is not

Table 16: The success rates of the cybersecurity choices available to the defender in the normal and penetrated states.

PLCs		**Switch**	
Action	Rate	Action	Rate
Enable authentication	0.95	Enable authentication	0.95
Disable wireless	0.99	Enable firewall	0.90
		Access control	0.96

bound to the network, "Adjacent" if the component is bound to the network but the attack is limited at the protocol level to an adjacent topology, or "Network" if the component is bound to the network and the component is remotely exploitable.

2. Attack Complexity: The attack complexity metric measures the conditions outside of the attacker's control that must be met for a vulnerability to be exploited. The metric can be assigned values of "High" if the attack's success is dependent on conditions outside of the attacker's control, and "Low" if specialized access conditions do not exist.
3. Privileges Required: The privileges required metric measures the privileges the attacker must have to exploit the vulnerability. The metric can be assigned values of "High" if the attacker requires administrative privileges over the component, "Low" if the attacker requires basic user privileges, and "None" if the attacker does not require authorization.
4. User Interaction: The user interaction metric measures whether another user besides the attacker to participate in the exploitation of the component. The metric can be assigned values of "Required" or "None".

The estimated success rates of the defender's actions are given in Table 16 and the success rate of the attacker's actions are given in Table 17. These estimates are dependent on expert opinion and are an area of ongoing research. These estimates may be validated using representative capture-the-flag games with cybersecurity professionals.

These success rates are used to calculate the probability of transitioning from one state to each of the other states given the actions that were chosen. We do this is in three steps. The first step is to calculate the probability that a given attack vector is successful. The second

Table 17: The success rates of the cybersecurity choices available to the attacker in the normal and penetrated states.

PLCs				
Action	θ_A^1 Rate	θ_A^2 Rate	θ_A^3 Rate	θ_A^4 Rate
Connect	0.85	0.99	0.98	0.92
Wireless exploit	0.70	0.85	0.95	0.75

Switch				
Action	θ_A^1 Rate	θ_A^2 Rate	θ_A^3 Rate	θ_A^4 Rate
Connect	0.85	0.99	0.98	0.92
Cyber attack	0.75	0.65	0.97	0.85
Physical attack	0.80	0.98	0.90	0.78

step is to calculate the probability that a component is penetrated given several attempted attack vectors. The third step is to calculate the probability of transitioning from one state to another given the devices that are penetrated in those states.

Step 1: Attack vector success Consider an attack action α with success rate p_α and a corresponding defense action, δ, with success rate p_δ. In the trivial case that an attack is not initiated, the probability that the component is penetrated is zero. If an attack is initiated, there are two possibilities:

1. The first case is that an attack is implemented and the corresponding defense is not implemented. In this case, the probability that the device is penetrated is equal to the attack's success rate, p_α.
2. The second case is that an attack is implemented and the corresponding defense is also implemented. In this case, we assume that the defense and attack actions are independent and the probability that the device is penetrated is $p_\alpha(1 - p_\delta)$.

Step 2: Component penetration There may be multiple attack vectors by which a component can be penetrated. Consider a component that can be penetrated via any one of n attack vectors. Let $\mathbb{A}_i$ be the event that attack vector i is successful, then

$p(\mathbb{A}_i)$ is the result obtained from the first step. The probability that a component is penetrated is given by $p(\bigcup_{i=1}^{n} \mathbb{A}_i)$. Using the identity $p\left(\bigcup_{i=1}^{n} \mathbb{A}_i\right) = 1 - p\left(\bigcap_{i=1}^{n} \mathbb{A}_i^c\right)$ and assuming independence of all $\mathbb{A}_i$, the probability that the component will be penetrated is $1 - \prod_{i=1}^{n} p(\mathbb{A}_i^c)$, where superscript c indicates the complement. The probability that the component will not be penetrated is trivially $\prod_{i=1}^{n} p(\mathbb{A}_i^c)$.

Step 3: State transition We are now able to consider transitions between states. Here we consider a transition where some components become penetrated. Transitions where components go from a penetrated status to normal status are addressed in Section 4.5.4. Consider a transition from state s_0 to state s_1. Let N be a set of components that go from a normal status in s_0 to a penetrated status in s_1, and let M be a set of components that have normal status in both s_0 and s_1. Let $p(i)$ be the probability that component i is penetrated. Assuming all components are independent, the probability of transitioning from s_0 to s_1 is $\prod_i^N p(i) \prod_j^M (1 - p(j))$. The results from the first and second steps are substituted into $p(i)$ and $p(j)$ as appropriate.

4.5.2 Hazard Transitions

Hazard transitions describe the probability of a hazard occurring as a result of the attacker penetrating the system. We assume the hazard initiation success rate is the same for a given hazard and attacker type, regardless of the penetrated state, as long as the criteria in Table 12 have been met. The assumed hazard initiation success rates are given in Table 18.

The disgruntled employee has the greatest success rates because of his insider knowledge of the plant. The government cyberwarrior also has high success rates because of his access to government resources. The radical activist and terrorist have the lowest success rates of the types, because they do not have the knowledge or resources of the other types. All of the types have relatively high success rates because of their high skill levels.

4.5.3 Loss Transitions

Loss transitions describe the probability of a loss occurring as a result of existing hazards and environmental conditions. Whether the attacker is able to damage the plant is dependent

Table 18: The success rates of the attacker's hazard initiation.

Hazard	θ_A^1 Rate	θ_A^2 Rate	θ_A^3 Rate	θ_A^4 Rate
H_1	0.65	0.80	0.75	0.68
H_2	0.80	0.90	0.87	0.80
H_3	0.67	0.80	0.75	0.65
H_4	0.57	0.75	0.70	0.60
H_5	0.74	0.85	0.83	0.75
H_6	0.61	0.80	0.75	0.64
H_7	0.58	0.75	0.71	0.61

not only on the actions selected by the players, but also on the state of the plant. To account for the state of the plant, event tree analysis can be used [91]. Consider the example shown in Figure 13. In this example, the initiating event is a hazard. If the defender's corrective action fails and a plant safety system fails, a loss will occur. Probabilities of success are assigned to each action to determine the probability of the loss if the hazard occurs.

In this game, we assume the transition probabilities shown in Table 19. Table 19 gives the probability that a given hazard causes a given absorbing loss. For simplicity, we assume that each hazard can only cause one absorbing loss.

4.5.4 Recovery Transitions

Recovery transitions describe the probability of the defender expunging the attacker from the plant to return the plant to a normal operating state. These transitions can be estimated using expert opinion and the NIST Guide for Cybersecurity Event Recovery [12]. There are two types of recovery in the SBG: penetration recovery and hazard recovery.

Penetration recovery can be initiated by the defender in any of the penetrated states. If successful, penetration recovery results in a transition from a penetrated state to the normal state. For simplicity, we assume a penetration recovery success rate of 0.80 for all penetrated states. The penetration recovery action trumps any of the attacker's offensive

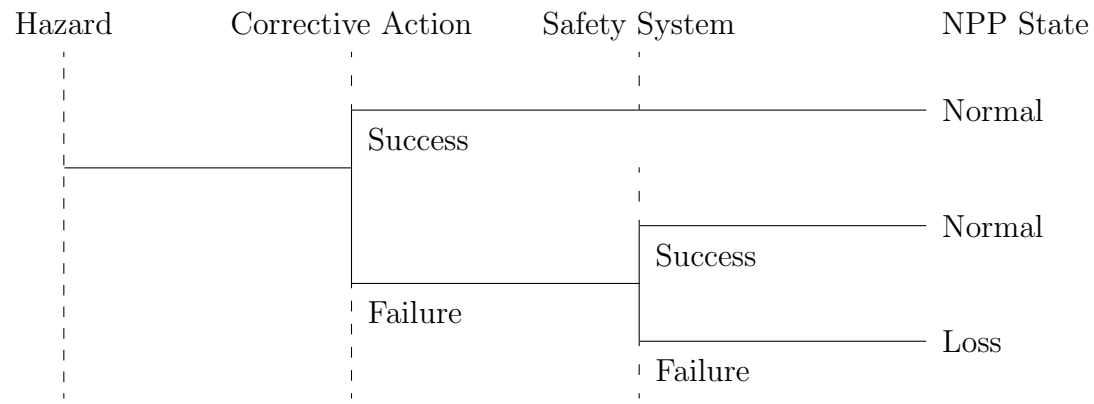


Figure 13: Event tree mapping a hazard to a loss.

actions directed towards ICS components, but the attacker's hazard initiation trumps the defender's penetration recovery. Consider the following examples for clarification.

- In this example, the defender chooses penetration recovery with a success rate of 0.80 and the radical activist chooses a wireless exploit on PLC-1A with success rate of 0.70. The probability of transitioning to the normal state is the penetration recovery success rate, 0.80. The probability of transitioning to another penetrated state where the attacker has penetrated PLC-1A is $0.70(1-0.80) = 0.14$. Finally, the probability of remaining in the originating penetrated state is $(1-0.70)(1-0.80) = 0.06$.
- In this example, the defender chooses penetration recovery with a success rate of 0.80 and the radical activist chooses H_1 hazard initiation with a success rate of 0.65. The probability of transitioning to the hazard state is the hazard initiation success rate, 0.65. The probability of transitioning to the normal state is $0.80(1-0.65) = 0.28$. Finally, the probability of remaining in the originating penetrated state is $(1-0.80)(1-0.65) = 0.07$.

Hazard recovery can be initiated by the defender in any of the hazard states. If successful, hazard recovery results in a transition from a hazard state to the normal state. For simplicity, we assume a hazard recovery success rate of 0.70 for all hazard states. The defender's hazard recovery action trumps the attacker's loss initiation action. Consider the example where the state is the H_5 hazard state the defender chooses hazard recovery with a success rate of 0.70,

Table 19: The probability of hazard states transitioning to loss states, given that the attacker has chosen loss initiation and the defender's hazard recovery is unsuccessful.

	L_1	L_2	L_3	L_4	L_5	L_6
H_1	—	5×10^{-6}	5×10^{-5}	—	2×10^{-2}	5×10^{-4}
H_2	5×10^{-2}	5×10^{-6}	5×10^{-5}	4×10^{-2}	9×10^{-1}	5×10^{-4}
H_3	—	—	—	4×10^{-2}	—	8×10^{-3}
H_4	9×10^{-1}	—	—	1×10^{-1}	—	—
H_5	—	1×10^{-5}	1×10^{-4}	—	—	1×10^{-3}
H_6	—	1×10^{-6}	1×10^{-5}	—	—	1×10^{-4}
H_7	2×10^{-2}	—	—	—	—	—

and the attacker chooses loss initiation. If the attacker successfully initiates a loss in the H_5 state, the probability of transitioning to L_2 is 1×10^{-5}, the probability of transitioning to L_3 is 1×10^{-4}, and the probability of transitioning to L_6 is 1×10^{-3}. The probability of transitioning to the normal state is the hazard recovery rate, 0.70. The probability of transitioning to L_2 is $(1 - 0.70) \times 10^{-5}$, the probability of transitioning to L_3 is $(1 - 0.70) \times 10^{-4}$, and the probability of transitioning to L_6 is $(1 - 0.70) \times 10^{-3}$. The probability of remaining in the hazard state is 0.2997.

4.6 Utility Functions

We now consider the utility functions of the players. The utility functions quantify the outcomes of the game for each player, and serves as the metric of each player's performance. There are two types of utility functions: immediate utility functions and cumulative utility functions. The immediate utility functions quantify the reward or penalty incurred by each player after one time step in the SBG. The cumulative utility function aggregates the immediate utilities to quantify performance over the entire game.

4.6.1 Immediate Utility Functions

The immediate utility function describes the payoff to the player after decisions have been made in a particular state. In this work, each player's immediate utility function has the units of dollars. The general form of player A/D's immediate utility function resulting from a transition from stochastic state s^i to state s^j after action profile $a_{D,A} = (a_D, a_A)$ and state history $s^H = \{s^0, s^1, \ldots, s^i, s^j\}$ is

$$r_{A/D}(s^H, a_{D,A}) = -\Psi_{A/D}(s^i, a_{D/A}) + \Omega_{A/D}(s^H) \tag{4.2}$$

The first term on the right-hand side, $\Psi_{A/D}$, represents the cost incurred by player A/D for selecting his action. This term is dependent only on the originating state and the action selected by that player. For the attacker, Ψ_A is the implementation cost of launching a cyber attack against the NPP. For the defender, Ψ_D is the cost of cybersecurity actions for the NPP.

The expenses for the defender's defensive cybersecurity choices are given in Table 14 and the expenses for the attacker's penetrating choices are given in Table 15. The expenses for the defender's recovery actions are given in Sections 4.4.1 and 4.4.2. The attacker incurs no costs to initiate hazards or losses after penetrating the system. We have assumed these values based on the access, resources, and skill of the threat agents as defined by the TAL. These values are assumed for demonstration purposes. In practice, additional threat intelligence and financial data should also be used. The government resources of the government cyberwarrior and organization resources of the radical activist and terrorist provide them with some advantages over the disgruntled employee, but the disgruntled employee's internal access to the NPP can also result in some reduced expenses. We assume that there is no expense to the attacker to abstain from a particular action. We also assume that there is no expense to the defender to leave a device in its default configuration.

The second term on the right-hand side, $\Omega_{A/D}$, represents the loss or gain incurred by player A/D as a result of the state transition. To be general, $\Omega_{A/D}$ is dependent on s^H, but often it is only dependent on s^i and/or s^j. Outcomes that could generate a gain for the attacker and a loss for the defender include the penetration of a device, initiation of a hazard,

Table 20: Utility ($) given to each player when a device is penetrated.

Device	Defender	θ_A^1	θ_A^2	θ_A^3	θ_A^4
PLC-1A	-2.50×10^5	7.50×10^6	5.00×10^7	1.25×10^7	2.50×10^5
PLC-1B	-2.50×10^5	7.50×10^6	5.00×10^7	1.25×10^7	2.50×10^5
PLC-2A	-2.50×10^5	7.50×10^6	5.00×10^7	1.25×10^7	2.50×10^5
PLC-2B	-2.50×10^5	7.50×10^6	5.00×10^7	1.25×10^7	2.50×10^5
Switch	-1.00×10^6	3.00×10^7	2.00×10^8	5.00×10^7	1.00×10^6
Network	-1.00×10^6	3.00×10^7	2.00×10^8	5.00×10^7	1.00×10^6

or the initiation of a loss. Outcomes that could generate a loss for the attacker and a gain for the defender include the prosecution of an attacker and publicity about a thwarted attack. Note that all of these outcomes must be assigned a monetary value for unit consistency.

The rewards given to each player when each device is penetrated are given in Table 20. These values are consistent with the L_7 magnitudes given in Table 6 for the defender and in Table 10 for the attacker. The rewards for device penetration are dependent on s^H. Each reward is only eligible to be earned once during the game. For example, if PLC-1A is penetrated by θ_A^1, θ_A^1 earns a reward of $\$7.50 \times 10^6$ and the defender incurs a penalty of $–\$2.50 \times 10^5$. If the defender returns the game from a penetrated state to the normal state, the attacker loses control of PLC-1A. If the attacker re-penetrates PLC-1A later in the game, the attacker does not gain an additional reward from penetrating the device. This is because the data that was on PLC has already been stolen by the attacker. The incentive for the attacker to penetrate the PLC again is the potential to cause a hazard or loss. The defender does incur the penalty each time a device is penetrated.

For every transition to the normal state, the defender earns a reward of $\$1 \times 10^8$ and the attackers do not incur a reward or penalty. For every transition to a penetrated state, the defender earns a reward of $\$1 \times 10^7$ and incurs the penalties specified in Table 20. This is because it is assumed that the plant can still operate in some capacity in these states.

The rewards earned by the attacker when the game transitions to a hazard state and the penalties incurred by the defender are given in Table 21. The hazard rewards and penalties

Table 21: Utility ($) given to each player when a hazard occurs.

Hazard	Defender	θ_A^1	θ_A^2	θ_A^3	θ_A^4
H_1	-1×10^{10}	5×10^7	3×10^9	2×10^{10}	5×10^{10}
H_2	-1×10^{10}	5×10^9	3×10^9	3×10^{11}	5×10^{10}
H_3	-1×10^7	5×10^9	3×10^9	2×10^{10}	3×10^7
H_4	-2×10^5	3×10^8	3×10^9	3×10^{11}	2×10^8
H_5	-1×10^{10}	5×10^7	3×10^9	2×10^{10}	5×10^{10}
H_6	-1×10^{10}	5×10^7	3×10^9	2×10^{10}	5×10^{10}
H_7	-2×10^5	3×10^8	3×10^9	3×10^{11}	2×10^8

were estimated to be 10% of the maximum loss reward or penalty that may result from that hazard. These values are only dependent on the hazard, not the penetrated state that causes the hazard, therefore these values are only a function of s^j.

The rewards earned by the attacker when the game transitions to a loss state are given in Table 10, and the penalties incurred by the defender are given in Table 6. These values are only dependent on the loss, not the hazard that causes the loss, therefore these values are only a function of s^j.

4.6.2 Cumulative Utility Functions

The cumulative utility functions aggregate the immediate utilities earned by the players throughout the game, and measures the performance of each player. Each player seeks to maximize his cumulative utility. The cumulative utility function of player A/D is

$$u_{A/D}(s^0, \sigma_A, \sigma_D) = \sum_{t=0}^{\infty} \beta_{A/D}^t \mathbb{E}[r_{A/D}(s^t, a_{D,A}^t, s^{t+1})] \tag{4.3}$$

The parameter $\sigma_{A/D}$ denotes player A/D's strategy — the discrete probability distribution that is assigned to the action set of that player at each state in the stochastic game. The cumulative utility function is also dependent on the initial state, s^0. Here the initial state is the normal state. The discount factor $\beta \in (0, 1)$ describes players' preferences for utility

earned earlier in the game relative to utility earned later in the game. The superscript t on the discount factor is an exponent, and superscript on the state and action profile variables is a time index. The function $\mathbb{E}[\cdot]$ denotes the expectation over the states and strategies.

The primary practical purpose of the discount factor is to affect the players' action selections. A player with a large discount factor is more willing to wait for a large payoff than a player with a small discount factor. We do not define discount factors for the attacker because the attacker's decision method will be governed by a separate algorithm. For the defender, we define a discount factor of 0.9999. This discount factor reflects the importance of utility earned at all time steps in the game.

4.7 Decision Algorithms

The final consideration in constructing the SBG is defining the players' decision-making processes. Consider the flowchart of SBG simulation shown in Figure 14. At each time step, the players make their decisions in parallel, and their decisions stochastically affect the progression of the game. There are three processes for the defender: Bayesian learning of the attacker's parameters, estimation of the attacker's type, and action selection via HBA. For the attacker, the decision-making process consists of a single algorithm that is dependent on the game history. This section describes the processes used by the players to select their actions.

4.7.1 The Defender

The defender uses Bayesian learning and HBA as described in Section 3.4.1 and Section 3.4.2, respectively. This section describes their implementation in the context of our game.

4.7.1.1 Bayesian Learning of Attacker's Parameters

We assume that the defender is unsure about the utility of environmental damage (L_2) for each attacker type. We selected

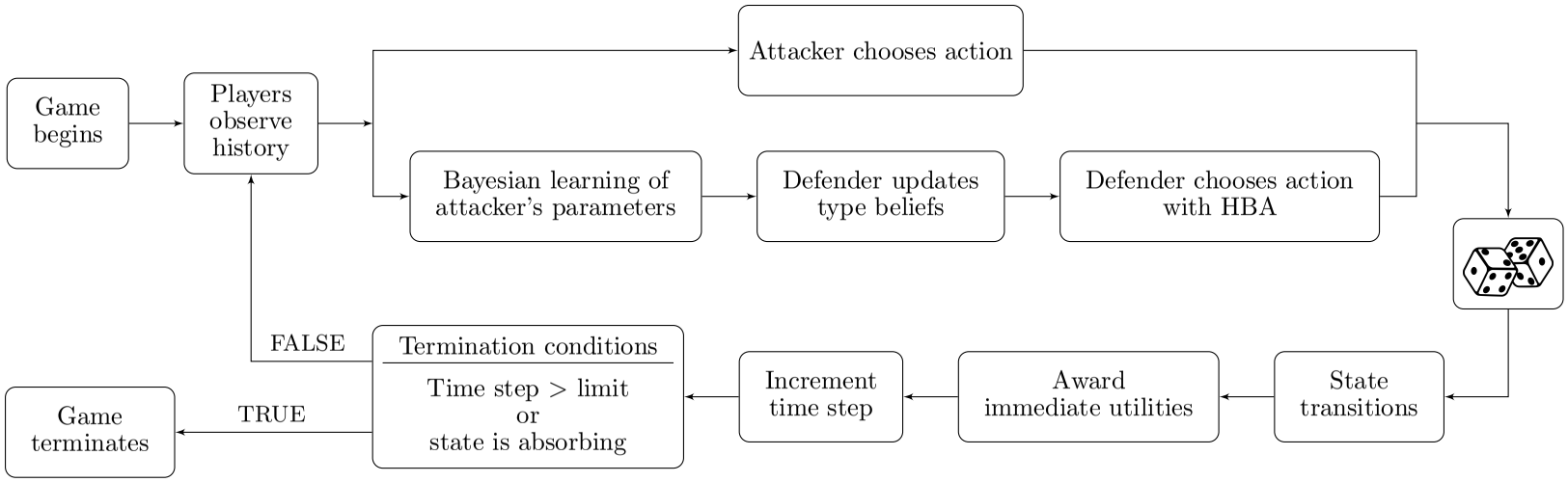


Figure 14: Flow chart of SBG simulation.

Table 22: Initial loss estimates for each attacker type.

Type	Lower Bound ($)	Upper Bound ($)	Initial Estimate ($)	True Value ($)
θ_A^1	1.00×10^5	1.00×10^7	5.05×10^6	2.00×10^6
θ_A^2	1.00×10^4	1.00×10^6	5.05×10^5	2.00×10^4
θ_A^3	1.00×10^5	1.00×10^7	5.05×10^6	1.00×10^6
θ_A^4	1.00×10^{10}	1.00×10^{12}	5.05×10^{11}	7.00×10^{11}

L_2 for several reasons. The first reason is that the utility of L_2 is likely to depend on factors that are not explicitly financial. For example, the attacker's code of ethics may affect the desirability of L_2, regardless of the financial damage to the NPP or the world. The second reason is that the value of L_2 is the most significant loss to the defender. The defender's penalty for L_2 is greater than the other loss penalties by at least three orders of magnitude.

The lower bound, upper bound, and initial estimate for each type are given in Table 22. The initial estimate is assumed to be the average of the lower and upper bounds. We define the initial belief density over L_2 for each type to be a truncated normal distribution centered about the initial estimate with a standard deviation equal to one half of the range of L_2.

The Bayesian learning algorithm uses approximate Bayesian updating to update the belief density over L_2 for each type. In approximate Bayesian updating, belief densities are approximated as polynomials and convolved to update beliefs. For all polynomial approximations, we sample 50 points uniformly distributed over the belief density and fit a fifth-degree polynomial to those points.

4.7.1.2 Estimating the Attacker's Type The probability of each attacker type is calculated using Equation 3.16 where the product posterior is given by Equation 3.17. The product posteriors are calculated using the attacker's strategy algorithm with the defender's estimates of the attacker's utility of L_2. To prevent the premature elimination of any types, we ensure that the minimum probability assigned to each type at each time step is 0.01. The product posteriors also require rescaling if values become close to machine epsilon.

4.7.1.3 HBA Implementation Implementing the full version of HBA given by Equations 3.14 and 3.15 for this game would be highly computationally expensive. Instead, we implement the version of HBA with path sampling given by Equations 3.18 and 3.19. The path-sampling version of HBA has three basic steps:

1. For each defense action available in the current state, compute the cumulative utility for n paths resulting from that action.
2. Calculate the average cumulative utility for each defense action.
3. Calculate the defense strategy as a uniform distribution over the defense actions that have the greatest average cumulative utility.

If the paths are stochastically selected, care must be taken to select an appropriate number of paths. If too few paths are sampled, there is a risk that the average cumulative utility of the sampled paths may not be close to the true expected utility of the defense action. If too many paths are sampled, the process may be too computationally expensive.

We tested HBA using various path sampling sizes to determine what sample size was most cost-effective. We set the number of paths per action, and the paths were stochastically generated using the state transition function, the attacker's decision algorithm with the most current estimates of the attacker's parameters, and a uniform distribution over the defender's subsequent actions. It became clear that a large sample size was not computationally feasible for this game. When using small sample sizes of stochastically generated paths, the decisions made by the defender were inconsistent and often clearly suboptimal. To accommodate our computational limitations, we instead implemented a sampling approach that was partly deterministic and partly stochastic.

Figure 15 shows two approaches to sampling paths. Each line represents one path stemming from a particular defense action. The direction of the line represents the utility earned by the defender over that path, with paths angled upwards having greater utility and paths angled downwards having lesser utility. The dashed gray line represents an "average" path that results in the expected cumulative utility for the action in question. Now consider a small sample of paths given by the red lines. This sample is analogous to an entirely stochastic sample in the game. The dashed red line represents the "average" path for the

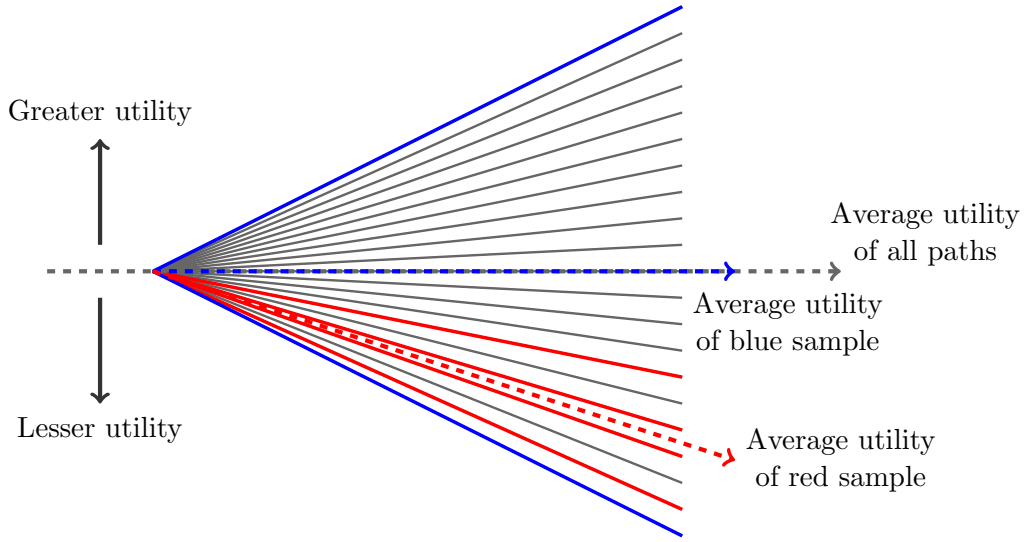


Figure 15: Sampling the upper and lower bounds of the paths provides a more consistent approximation of the average path utility than performing a purely stochastic sample.

red sample. Because the sample size is small, it is likely that the average utility of the sample will not match the true expected utility. Now consider the sample given by the blue lines. This sample is analogous to a partially stochastic sample in the game. The dashed blue line represents the "average" path for the blue sample. If the expected utility of an action is close to the midpoint between the lower and upper bounds of the utility, then intentionally sampling the lower and upper bounds may provide a better approximation of the true expected utility than sampling a random set of paths.

Because we are constrained by computational cost, we know that we cannot generate a sufficiently large sample with true stochastic sampling to consistently approximate the action's expected utility. For each action in the current state, we generate two paths. The first path is obtained by the defender always choosing to abstain from cybersecurity actions after taking the current action. This is expected to approximate a lower bound on the utility resulting from the current action. This assumption is not valid for scenarios where the attacker cannot significantly influence the defender's utility. The second path is obtained by the defender always choosing the most secure cybersecurity actions after taking

the current action. This is expected to approximate an upper bound on the utility resulting from the current action. This approximation is not valid when security costs significantly outweigh security benefits. Within these paths, the attacker's actions and state trajectory are still generated stochastically. The attacker's actions are generated stochastically using Algorithm 2 with the defender's beliefs about the attacker's utility of L_2.

The length of the path can also affect computation time. One option is to allow the path to continue indefinitely until an absorbing state is reached. The less expensive option is to limit the paths to a finite number of time steps. The path is terminated if it has not reached an absorbing state by the time the limit has been reached. In this work, we use a maximum path length of five time steps. This path length was selected to allow the simulated attacker approximately two attempts to reach a loss state.

The disadvantage of using a small path length is that the defender's long-term view of the game is limited. As an example, consider the situation where the game is in a hazard state, and the defender must choose between two actions: (1) attempt hazard recovery to return the game to the normal state, or (2) abstain from action and risk the game transitioning to a loss state. Hazard recovery has a significant cost, but the defender is able to accrue rewards when the game is in the normal state and some penetrated states. Transitioning from a hazard to a loss causes the defender to incur a significant penalty and terminates the game. Because the path size is limited, the defender may not recognize the full potential of future rewards in the normal and penetrated states. This defender does not have an accurate incentive to recover the plant and may choose to abstain from action.

We introduce a utility adjustment factor to account for potential future rewards that may be omitted by a small path length. The utility factor is added to the expected utility of the path if the path is not terminated by an absorbing state and the final time step of the path does not exceed a threshold. The utility factor is given by

$$UF_\theta = E[\Delta u_\theta] \sum_{\gamma=t_p}^{E[t_\theta]} \beta^\gamma \tag{4.4}$$

The utility factor is UF_θ and it is specific to the attacker type, θ, that is being faced in the current path. The time step at the end of the path is given by t_p and the expected duration

Table 23: Parameters of the utility adjustment factor used to evaluate paths in HBA.

Type	$E[t_\theta]$ (time steps)	$E[\Delta u]$ (\$)
θ_1	24	3.03×10^7
θ_2	12	2.53×10^7
θ_3	14	2.61×10^7
θ_4	17	2.14×10^7

of a game played against θ is given by $E[t_\theta]$. The utility factor is only implemented if $t_p < E[t_\theta]$. The defender's discount factor is β and $E[\Delta u_\theta]$ is the expected utility earned by the defender in per time step when facing type θ.

The values of $E[t_\theta]$ and $E[\Delta u_\theta]$ were obtained from preliminary simulation data. We simulated the attacker playing against each type 500 times. In these simulations, the attacker followed the decision algorithm given in Section 4.7.2 and the defender always chose the most secure action. The parameters $E[t_\theta]$ were selected from the mean of the data. The parameters $E[\Delta u_\theta]$ were selected from the medians of the data because the results were significantly skewed. The simulation results are discussed in greater detail in Chapter 6. The values of $E[t_\theta]$ and $E[\Delta u_\theta]$ are given in Table 23.

In summary, for HBA we use two paths per action in a given state, where the first path is given by the least secure subsequent defense actions, and the second path is given by the most secure subsequent defense actions. We limit the depth of the path to be five time steps and add a utility adjustment factor for paths that do not reach an absorbing state.

4.7.2 The Attacker

The mechanism of the attacker's decision-making in the normal and penetrated states is Algorithm 2. The implementation of this decision algorithm is somewhat arbitrary. In general, attacker types are not required to follow the same decision algorithm. It is assumed

Table 24: Type parameters used in the attacker's decision algorithm.

Parameter	θ_A^1	θ_A^2	θ_A^3	θ_A^4
η	2.0	1.0	4.0	2.5
ν	1.2	2.0	1.5	0.14

that all attacker types follow this algorithm, but with different parameters. In practice, additional types may be constructed if there are multiple credible decision algorithms for a particular threat.

The algorithm evaluates the hazard options accessible to the attacker from the current state, and compares them to the attacker's most desirable hazard. If the most desirable hazard is inaccessible, we calculate the probability that the attacker settles for the best accessible hazard. The remaining probability is distributed over the actions that could bring the attacker to a state from which a more desirable hazard is accessible. The probability distribution over the attacker's actions is then sampled to determine the attacker's action. The following section describes the algorithm in greater detail.

In line 1 of the algorithm, the attacker ranks the hazards by their expected utilities. The expected utility of hazard H is given by

$$u(H) = \sum_{L \in \mathcal{L}} p(L|H)\Omega_\theta(L) \tag{4.5}$$

The set of all losses is given by $\mathcal{L}$. The parameter $p(L|H)$ gives the probability of loss L occurring as a consequence of H, assuming no defender intervention. If defender intervention were included, all expected hazard utilities would be scaled by the same hazard recovery failure rate, because it is assumed that hazard recovery has the same cost and success rate for all hazards. The parameter $\Omega_\theta(L)$ is the utility that the attacker of type θ assigns to L. The hazard with the greatest expected utility is denoted by H^*.

In lines 2–7 of the algorithm, the attacker identifies the best hazard that is accessible from the current state, s_0, and compares that hazard to H^*. For further calculations, the

Algorithm 2 The attacker's decision algorithm for normal and penetrated states

Rank the hazards by their expected utilities, and identify hazard, H^*, with the greatest expected utility (Equation 4.5)
if at least one hazard is accessible from the current state, s_0 **then**
    Identify accessible hazard, H_a, that has the greatest scaled utility, μ_a (Equation 4.6)
    Calculate $p(H_a)$ (Equation 4.10)
    Assign $p(H_a)$ to initiation of H_a and assign zero to initiation of all $H \neq H_a$
else
    Set $p(H_a) = 0$, $\mu_a = 0$
end if
for each hazard, H_i, that is currently unavailable and has utility $\mu_i > \mu_a$ **do**
    Calculate $p(H_i)$ (Equation 4.11)
    for each set of penetrated devices, D_j, that can cause H_i **do**
        Identify state, s_1, where D_j has been achieved (relative to s_0)
        for each action, α, in the action set **do**
            if α targets a transition from s_0 to s_1 **then**
                Calculate estimated utility $\tilde{u}(\alpha|H_i, s_0, s_1)$ (Equation 4.12)
            else
                Set $\tilde{u}(\alpha|H_i, s_0, s_1) = 0$
            end if
        end for
    end for
end for
for each penetrating action, α in set of all penetrating actions, $\mathcal{A}$ **do**
    Calculate $p(\alpha)$ (Equation 4.15)
end for
Sample the discrete probability distribution over all actions to select the action

utility is scaled to introduce greater sensitivity with respect to L_2, the loss with uncertain utility. The scaled expected utility of a hazard, H, is

$$\mu(H) = \sum_{L_i \neq L_2} p(L_i|H)\Omega_\theta(L_i) + p(L_2|H)\Omega_\theta(L_2)\hat{\Omega}_\theta(L_2)^{\nu_\theta} \tag{4.6}$$

The circumflex on $\hat{\Omega}_\theta$ signifies the belief that the agent using the decision algorithm has regarding θ's utility parameter. For θ, $\Omega_\theta = \hat{\Omega}_\theta$, but if the defender is using this algorithm to make predictions about θ's behavior, the two parameters may differ. Finally, ν_θ is a scaling parameter specific to θ. Care should be taken when defining ν_θ to ensure that the argument of Equation 4.9 is within the function's domain. The values of ν for each type are given in Table 24. The term $\hat{\Omega}_\theta(L_2)^{\nu_\theta}$ acts a scaling factor used to increase the sensitivity of the attacker's action selection to variations in the utility assigned to L_2. The scaled utility of H^* is μ^*.

The best accessible hazard is denoted by H_a and its scaled utility is μ_a. The probability of initiating H_a is obtained by comparing μ_a to μ^*, and by considering the number of times the game has returned to the normal state from a penetrated or hazard state. The more times the game has returned to the normal state, the more likely the attacker is to settle for a hazard that is less desirable than H^*. To define $p(H_a)$, we began with the general form

$$p(H_a) = \frac{\mu_a}{\mu^*}[X \tanh(C\eta) + 1] \tag{4.7}$$

The parameter η is the number of times the game has returned to the normal state from a penetrated or hazard state. The hyperbolic tangent function was selected for its asymptotic properties as the argument approaches infinity. The parameters X and C were selected such that $p(H_a)$ approaches one as η increases and as μ_a approaches μ^*. We define X and C in Equations 4.8 and 4.9, respectively.

$$X = \frac{\mu^* - \mu_a}{\mu_a} \tag{4.8}$$

$$C_\theta = \frac{1}{\eta_\theta} \operatorname{arctanh}\left(\frac{0.9\mu^* - \mu_a}{\mu^* - \mu_a}\right) \tag{4.9}$$

The parameter X was defined such that $p(H_a)$ approaches one as μ_a approaches μ^*. The parameter C_θ was defined to control the rate at which $p(H_a)$ approaches one, and is dependent on the characteristics of the attacker type, θ. Types that are more patient have smaller values of C_θ and types that are less patient have larger values of C_θ. The parameter η_θ is the number of returns to the normal state for the attacker to settle to the best available hazard with a 90% chance. The values of η_θ are given in Table 24.

Through algebraic manipulation of Equations 4.7, 4.8, and 4.9, we obtain the probability of initiating H_a as

$$p(H_a) = \frac{\mu^* - \mu_a}{\mu^*} \tanh(C_\theta \eta) + \frac{\mu_a}{\mu^*} \tag{4.10}$$

If no hazards are accessible from s_0, the parameters $p(H_a)$ and μ_a are set equal to zero for later use in the algorithm.

In lines 9–21 of the algorithm, we calculate the value of attack actions with respect to achieving access to hazards with scaled utility greater than μ_a. First, we calculate the probability of pursuing each hazard H_i with $\mu_i > \mu_a$. This probability is calculated as the ratio of μ_i to the total scaled utility of all hazards that meet the scaled utility criteria. The probability is

$$p(H_i) = \frac{\mu_i}{\sum_{H_j \in \mathcal{H}} \mu_j} \tag{4.11}$$

The set of all hazards with $\mu > \mu_a$ is given by $\mathcal{H}$, and the scaled utility of H_j is given by μ_j.

Next, the algorithm considers each method by which the hazard in question can become accessible. For example, H_1 is accessible if PLC-1B is penetrated, if PLC-2B is penetrated, or if the switch is penetrated (Table 12). The state, s_1, is identified where the set of penetrated devices, D_j, has been achieved. The state where D_j has been achieved is identified relative to s_0. For example, suppose the game is in state 16 where PLC-1A has been penetrated and the attacker is considering penetrating PLC-1B to cause H_1. PLC-1B is the only device penetrated in state 8, but the algorithm defines s_1 as state 24, where both PLC-1A and PLC-1B are penetrated. This is because PLC-1A has already been penetrated and the underlying structure of the game does not offer the attacker any utility or advantage for relinquishing control of a penetrated device.

The attacker then evaluates the estimated utility of each attack action with respect to causing H_i via s_1. The estimated utility of action α given that the attacker is targeting H_i via s_1 is

$$\tilde{u}(\alpha|H_i, s_0, s_1) = -\Psi_\theta(s_0, \alpha) + \frac{1}{2}(SR_{\text{Abstain}} + SR_{\text{Secure}}) \sum_{L \in \mathcal{L}} p(L|H_i)\Omega_\theta(L) \tag{4.12}$$

The cost of the action is denoted by $-\Psi_\theta(s_0, \alpha)$ and is given in Section 4.6. The term $\frac{1}{2}(SR_{\text{Abstain}} + SR_{\text{Secure}})$ is the average success rate of the action when considering the most secure and least secure defense actions. These success rates are discussed in greater detail below. The weighting of the items in the average is arbitrary and for demonstration purposes. An attacker who believes the defender is more likely to play the most secure option could assign a greater weighting to the corresponding success rate. The attacker could also assign weighting to additional defense actions. The summation term is the expected utility of H_i, where $\Omega_\theta(L)$ is the utility earned by the attacker if loss L occurs (Section 4.6). The set of all losses is denoted by $\mathcal{L}$.

The attack action has the greatest success rate when the defender abstains from implementing a defense action. The attack action has the lowest success rate when the defender implements the most secure defense action. These success rates were already been calculated when analyzing state transitions, and are described in greater detail in Section 4.5.1. The success rates are denoted by

$$SR_{\text{Abstain}} = p(s_1|s_0, \text{Abstain}, \alpha, \theta) \tag{4.13}$$

$$SR_{\text{Secure}} = p(s_1|s_0, \text{Secure}, \alpha, \theta) \tag{4.14}$$

Lines 22–25 of the algorithm calculate the probability of each action and select the final action. This calculation only applies to penetrating actions. The probabilities of hazard initiation actions were previously calculated in the algorithm. The probability of selecting penetrating action α is

$$p(\alpha) = (1 - p(H_a)) \sum_{H \in \mathcal{H}} p(H) \frac{\sum_{s \in \mathcal{S}} \tilde{u}(\alpha|H, s_0, s)}{\sum_{A \in \mathcal{A}} \sum_{s \in \mathcal{S}} \tilde{u}(A|H, s_0, s)} \tag{4.15}$$

This equation comes from the probability chain rule. The first term is the probability that a hazard is not initiated. The term $p(H)$ is the probability that the attacker pursues H given that a hazard has not been initiated. The fraction is the probability that the attacker chooses α given that the attacker is pursuing H. It is the ratio of the total estimated utility of α from all pursuit options for H, to the total estimated utility of all actions for all pursuit options for H. This approach is similar to Equation 4.11 used to calculate $p(H_i)$.

We have now calculated the probability of all penetration actions and all hazard initiation actions. The last step of the algorithm is to randomly select an action using the probability distribution over the actions.

In hazard states, we define the attacker to always select loss initiation. All loss states in this game are absorbing, therefore there are no decisions to be made in these states for either player.

5.0 Simulation Examples

This chapter contains examples of individual games played against each attacker type. For each example, we provide the state history of the game, the utilities of the players, the defender's estimate of the attacker's utility of L_2, and the defender's belief regarding the type of the attacker. The aggregate results of the SBG are presented and discussed in Chapter 6.

5.1 Radical Activist Simulation

The state trajectory for the example game played against the radical activist (θ_A^1) is shown in Figure 16. The game has a relatively long duration and cycles between normal, penetrated, and hazard states five times. These results are consistent with the average state occurrences shown in Figure 17. The most frequently occurring penetrated states were states 4 and 16. In state 4, PLC-2A is penetrated, and in state 16, PLC-1A is penetrated. Excessive removal of suppression pool inventory (state 34) was the hazard that occurred the most frequently, and damaged public opinion (state 42) was the loss that occurred most frequently.

The cumulative utilities of the defender and θ_A^1 are plotted on a logarithmic scale in Figure 18. The defender loses utility when the game enters hazard states and gains utility during the returns to the normal state. The loss that occurs in this example costs on the order of $\$10^5$ to the defender, so the loss is not very visible on the logarithmic scale. The attacker steadily gains utility as the game enters the hazard states and makes a significant gain in utility when damaged public opinion occurs.

The defender's belief about the utility assigned to L_2 by θ_A^1 is shown in Figure 19. The dashed blue distribution is the original truncated normal distribution, the dashed orange distribution is the belief at the final time step of the game, the gray distributions are beliefs at the intermediate time steps, and the vertical dashed line is the true utility. The estimates

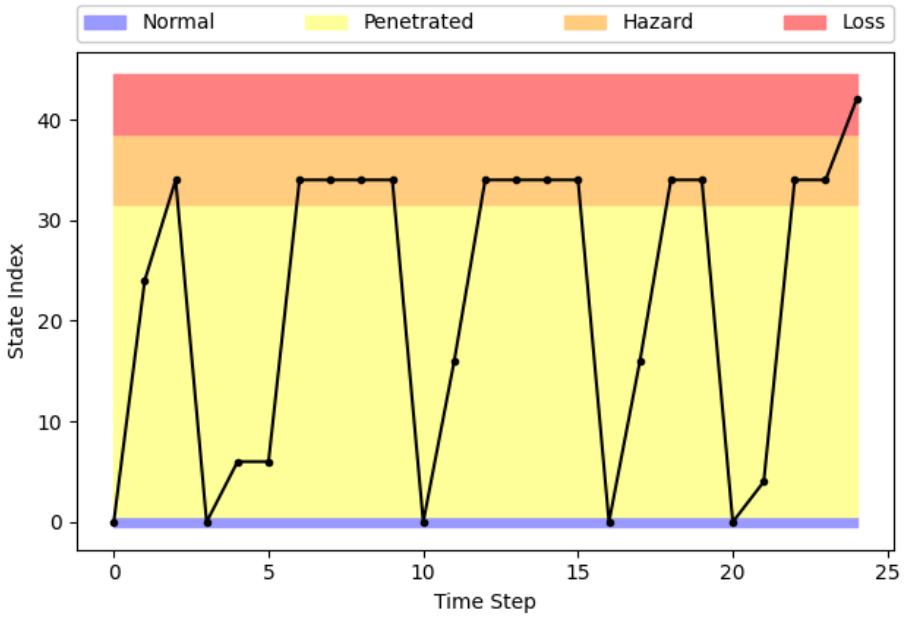


Figure 16: State trajectory of the example game played against θ_A^1.

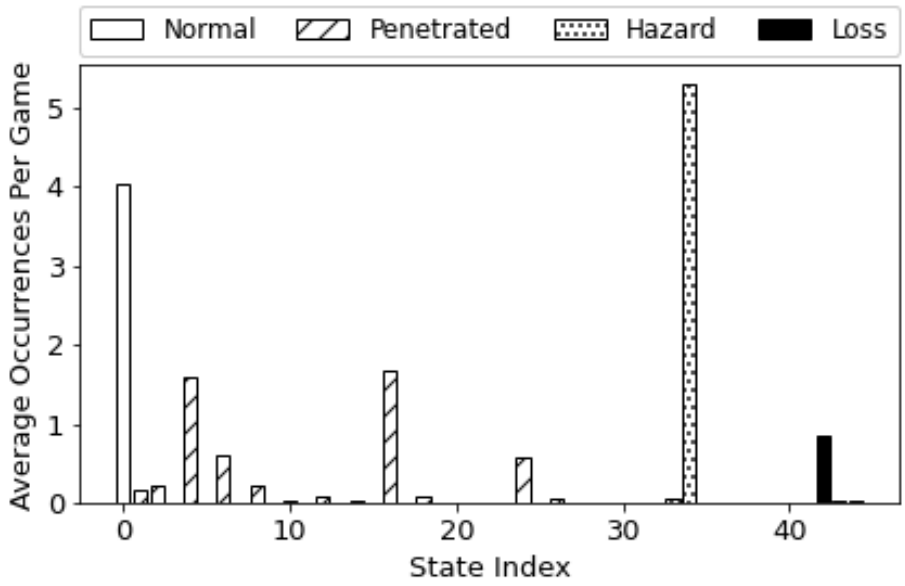


Figure 17: Average occurrences of each state when HBA is used against each θ_A^1.

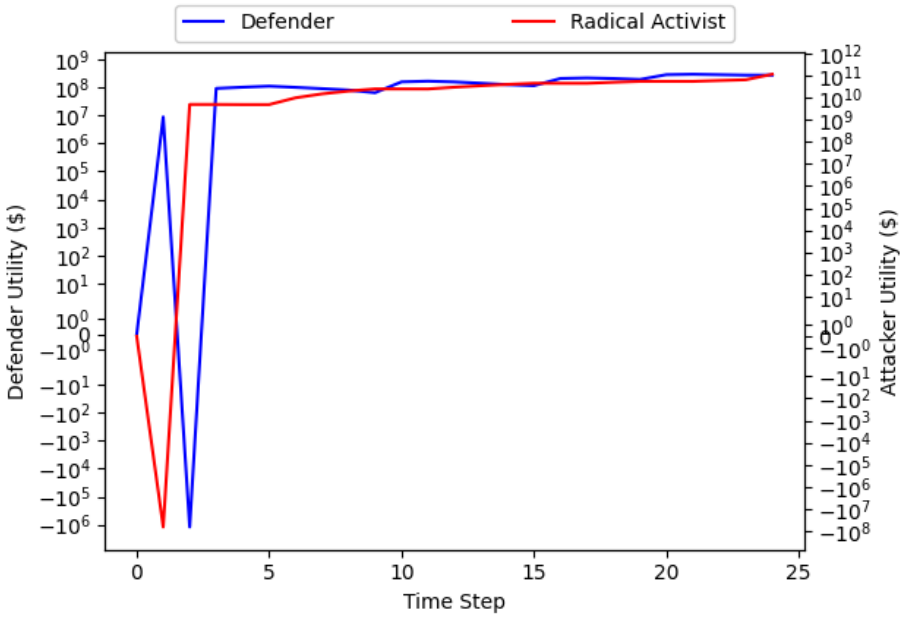


Figure 18: Players' utilities from the example game played against θ_A^1.

extracted from these distributions are plotted in Figure 20. The estimate made significant progress in the first five time steps of the game, but progress slowed because inferences could not be made in the hazard states and the attacker repeated actions in other states.

The defender's beliefs regarding the attacker's type while facing θ_A^1 are shown in Figure 21. The beliefs initially favored θ_A^3, but quickly converged to the correct type.

5.2 Disgruntled Employee Simulation

The state trajectory for the example game played against the disgruntled employee (θ_A^2) is shown in Figure 22. The game has a relatively short duration and cycles between normal, penetrated, and hazard states twice. These results are consistent with the average state occurrences in Figure 23. The most frequently occurring penetrated state was state 2.

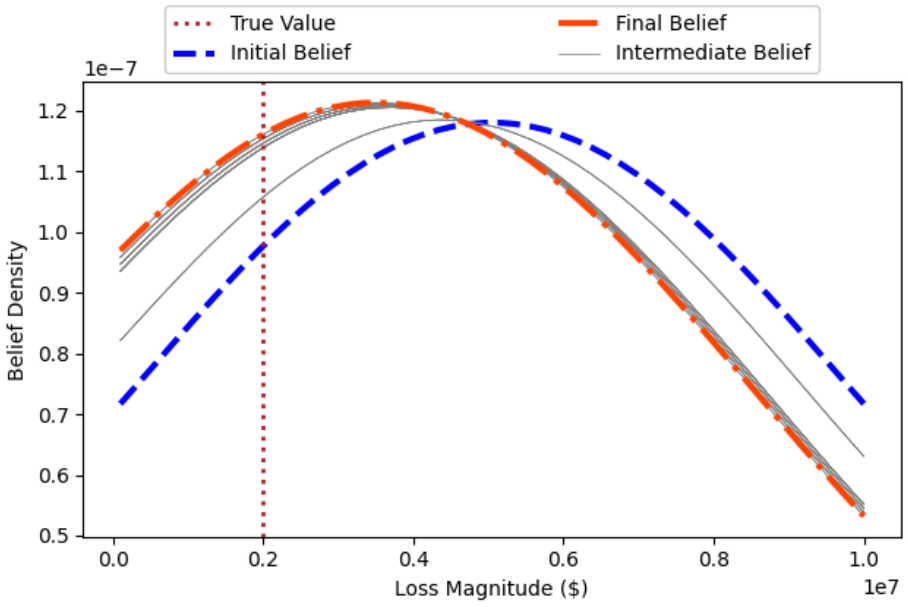


Figure 19: The defender's belief distributions over the range of the attacker's possible utilities assigned to L_2 from the example game played against θ_A^1.

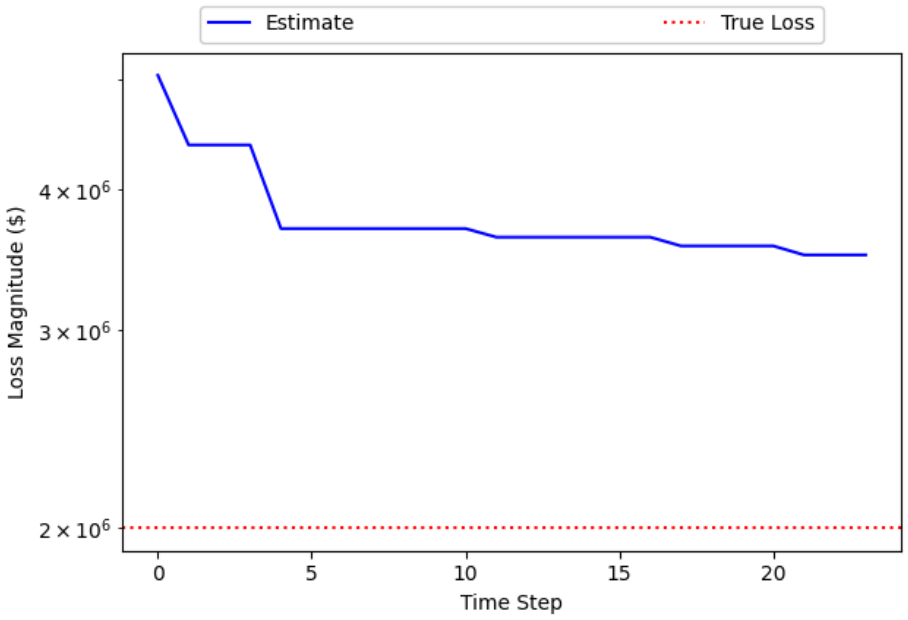


Figure 20: The defender's estimate of the attacker's utility of L_2 from the example game played against θ_A^1.

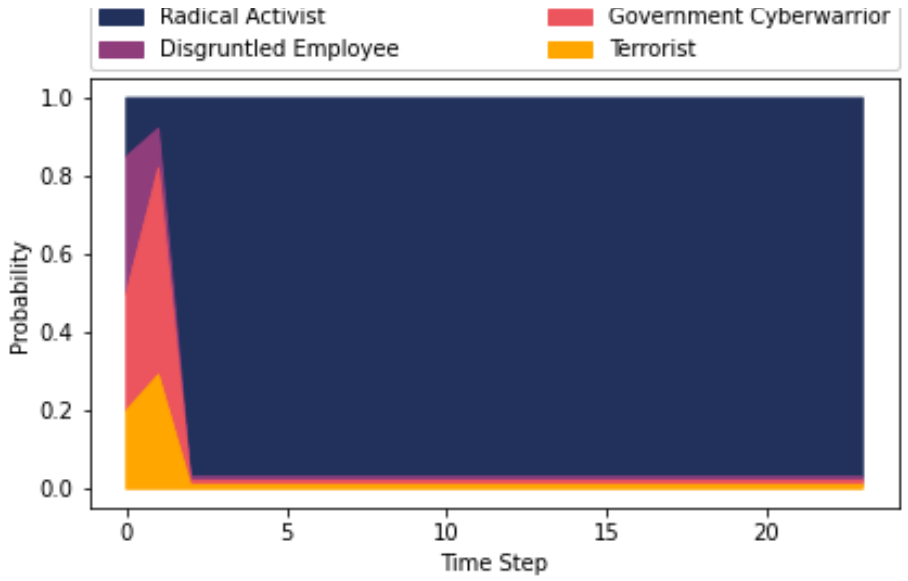


Figure 21: The defender's beliefs regarding the attacker's type from example game played against θ_A^1.

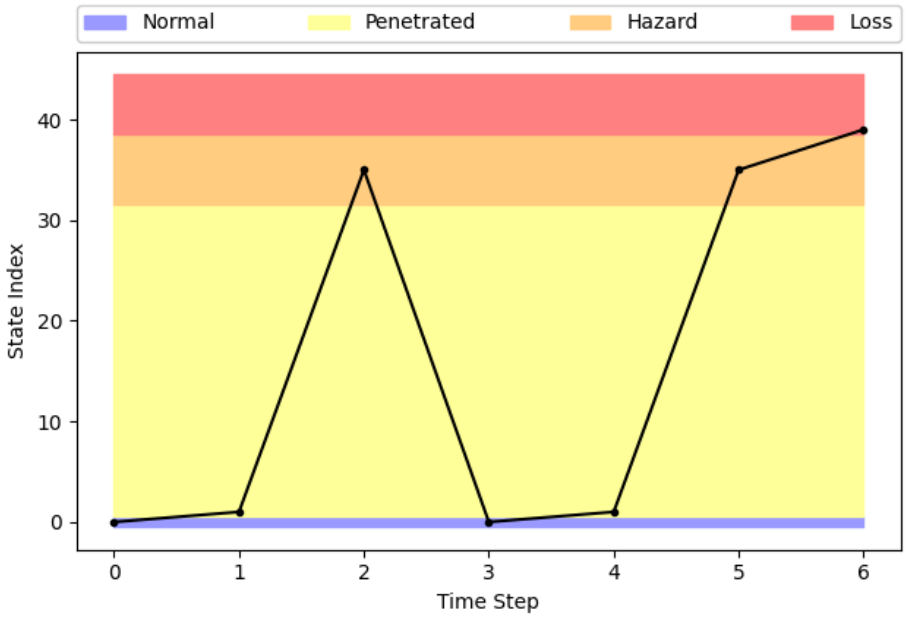


Figure 22: State trajectory of the example game played against θ_A^2.

In state 2, the switch is penetrated. Reactor trip (state 35) was the hazard that occurred the most frequently and loss of power generation (state 39) was the loss that occurred most frequently.

The cumulative utilities of the defender and θ_A^2 are plotted on a logarithmic scale in Figure 24. The defender loses utility when the game enters hazard states and gains utility during the returns to the normal state. The loss that occurs in this example costs on the order of $\$10^6$ to the defender, so the loss is not very visible on the logarithmic scale. The attacker gains utility when the game enters the hazard states and makes a significant gain in utility when loss of power generation occurs.

The defender's belief about the utility assigned to L_2 by θ_A^2 is shown in Figure 25. The dashed blue distribution is the original truncated normal distribution, the dashed orange distribution is the belief at the final time step of the game, the gray distributions are beliefs at the intermediate time steps, and the vertical dashed line is the true utility. The estimates

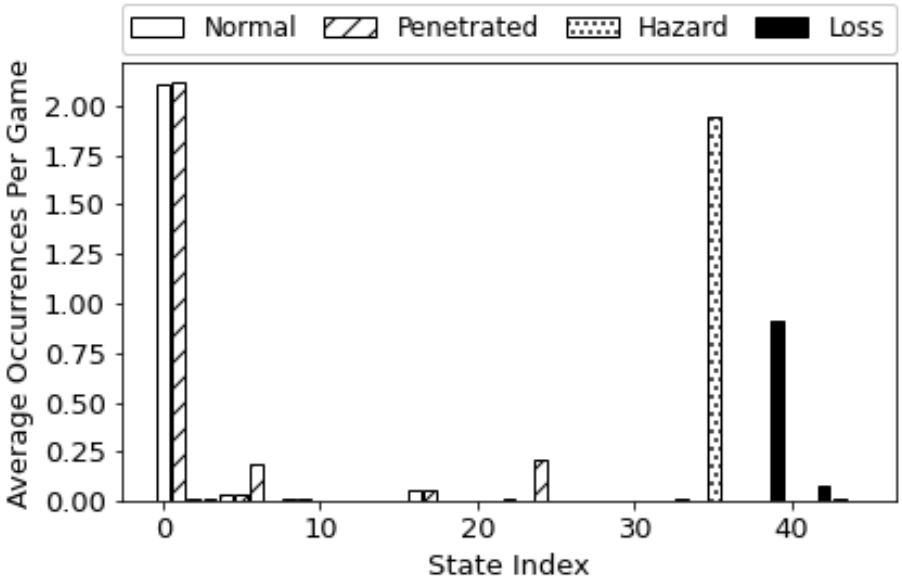


Figure 23: Average occurrences of each state when HBA is used against each θ_A^2.

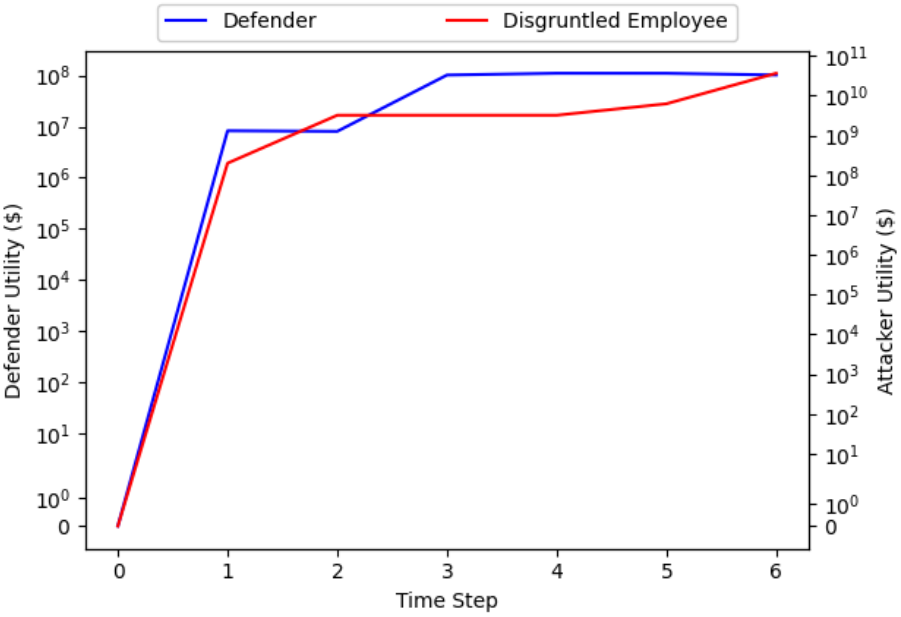


Figure 24: Players' utilities from the example game played against θ_A^2.

extracted from these distributions are plotted in Figure 26. The estimate did not make significant progress because the decisions made by θ_A^2 in the normal and penetrated states were not significantly sensitive to changes in the utility of L_2.

The defender's beliefs regarding the attacker's type while facing θ_A^2 are shown in Figure 27. The beliefs immediately identified to the correct type.

5.3 Government Cyberwarrior Simulation

The state trajectory for the example game played against the government cyberwarrior (θ_A^3) is shown in Figure 28. The game has a relatively short duration and cycles between normal, penetrated, and hazard states twice. This game also involves escalation from state 16 where PLC-1A is penetrated to state 22 where PLC-2A and PLC-2B are also penetrated. These results are consistent with the average state occurrences shown in Figure 29. The most frequently occurring penetrated states were state 6 and state 24. In state 6, PLC-2A and PLC-2B are penetrated, and in state 24, PLC-1A and PLC-2A are penetrated. Reactor trip (state 35) was the hazard that occurred the most frequently and loss of power generation (state 39) was the loss that occurred most frequently.

The cumulative utilities of the defender and θ_A^3 are plotted on a logarithmic scale in Figure 30. The interpretation of this figure is the same as that for the game played against θ_A^2 shown in Figure 24.

The defender's belief about the utility assigned to L_2 by θ_A^3 is shown in Figure 31. The dashed blue distribution is the original truncated normal distribution, the dashed orange distribution is the belief at the final time step of the game, the gray distributions are beliefs at the intermediate time steps, and the vertical dashed line is the true utility. The estimates extracted from these distributions are plotted in Figure 32. The estimate did not make significant progress because the decisions made by θ_A^2 in the normal and penetrated states were not significantly sensitive to changes in the utility of L_2.

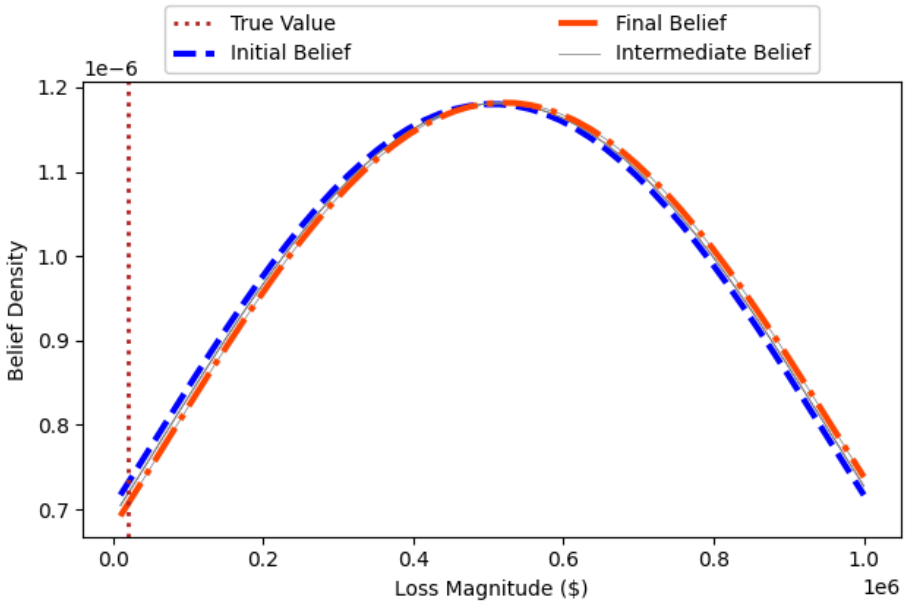


Figure 25: The defender's belief distributions over the range of the attacker's possible utilities assigned to L_2 from the example game played against θ_A^2.

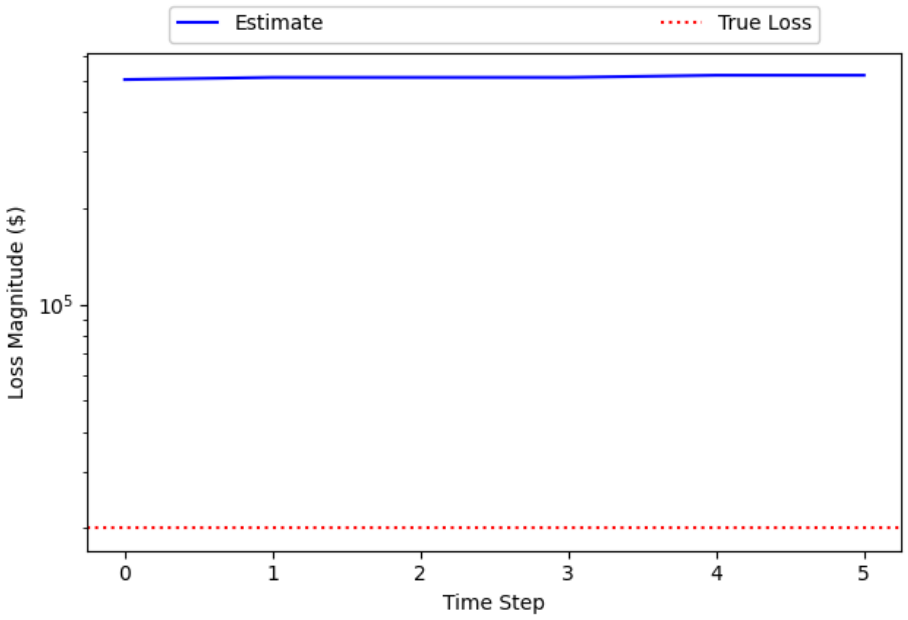


Figure 26: The defender's estimate of the attacker's utility of L_2 from the example game played against θ_A^2.

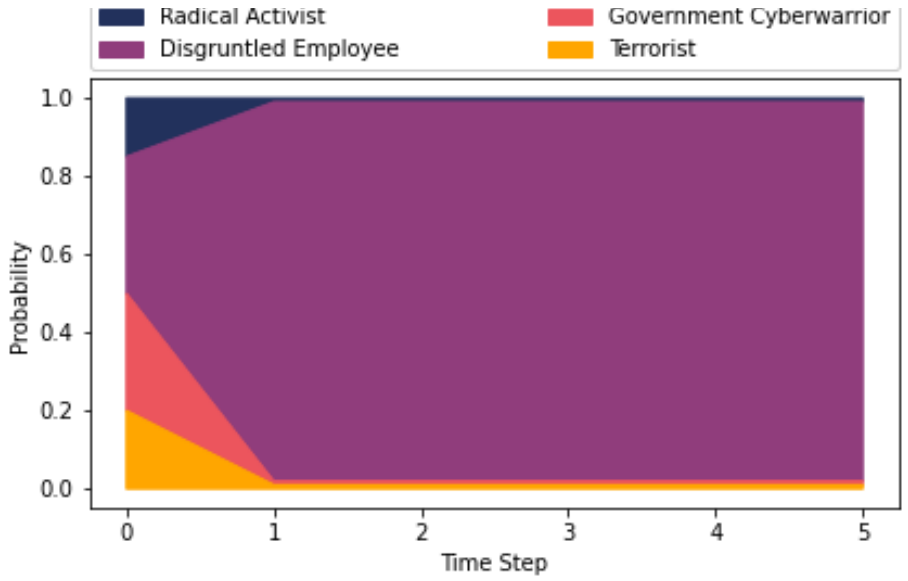


Figure 27: The defender's beliefs regarding the attacker's type from example game played against θ_A^2.

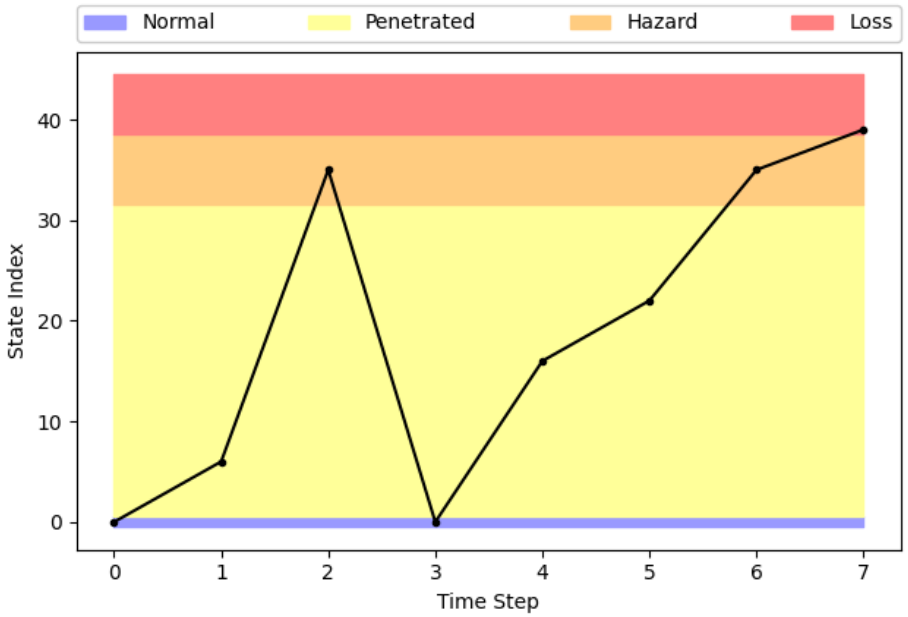


Figure 28: State trajectory of the example game played against θ_A^3.

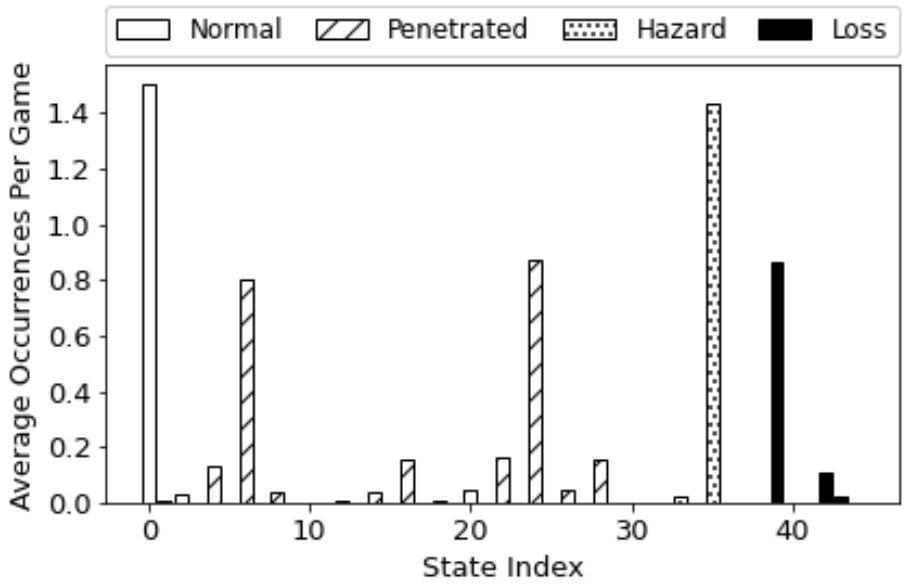


Figure 29: Average occurrences of each state when HBA is used against each θ_A^3.

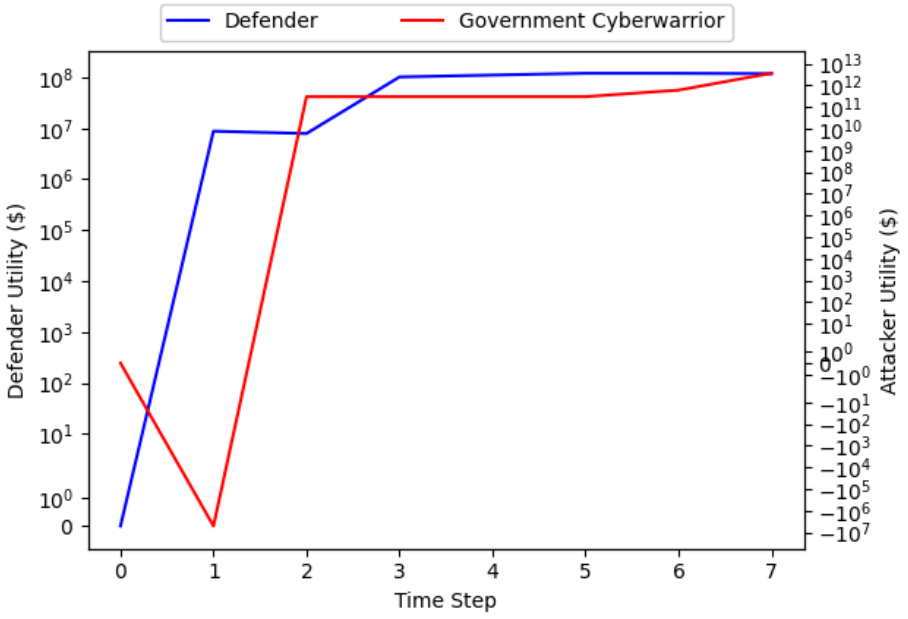


Figure 30: Players' utilities from the example game played against θ_A^3.

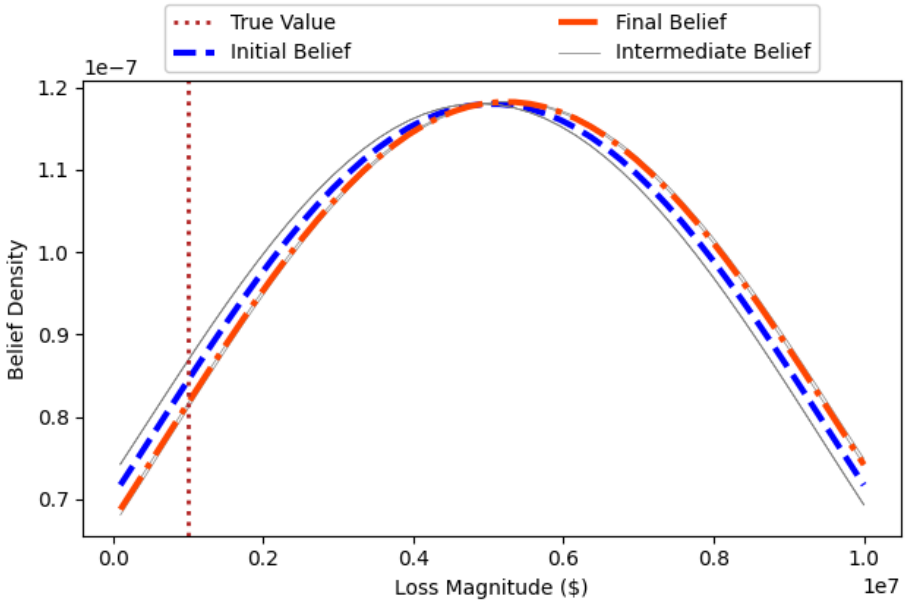


Figure 31: The defender's belief distributions over the range of the attacker's possible utilities assigned to L_2 from the example game played against θ_A^3.

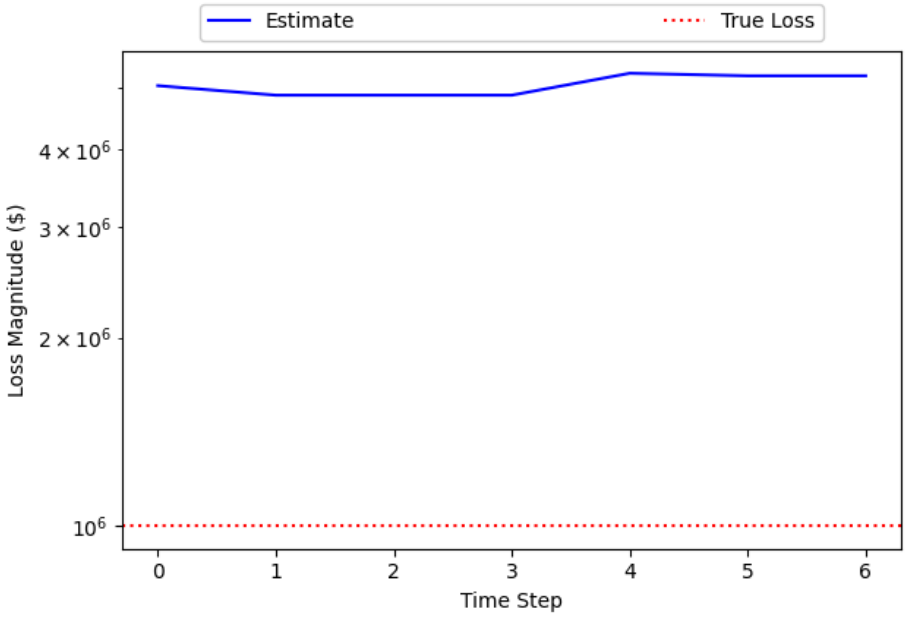


Figure 32: The defender's estimate of the attacker's utility of L_2 from the example game played against θ_A^3.

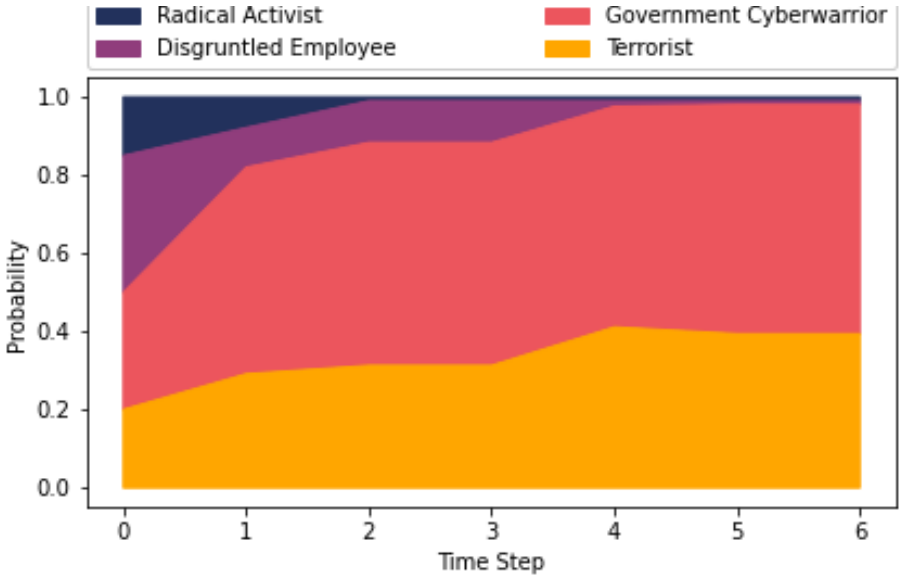


Figure 33: The defender's beliefs regarding the attacker's type from example game played against θ_A^3.

The defender's beliefs regarding the attacker's type while facing θ_A^3 are shown in Figure 33. At the conclusion of the game, the beliefs assigned the greatest probability to the correct type, but were not able to rule out θ_A^4.

5.4 Terrorist Simulation

The state trajectory for the example game played against the terrorist (θ_A^4) is shown in Figure 28. The game has a very short duration and does not return to the normal state once the NPP has been penetrated. As in the previous example for θ_A^3, this game also involves escalation from state 16 where PLC-1A is penetrated to state 22 where PLC-2A and PLC-2B are also penetrated. These results are consistent with the average state occurrences shown in Figure 35. Similar to the example for θ_A^3, the most frequently occurring penetrated states were state 6 and state 24. In state 6, PLC-2A and PLC-2B are penetrated, and in state 24,

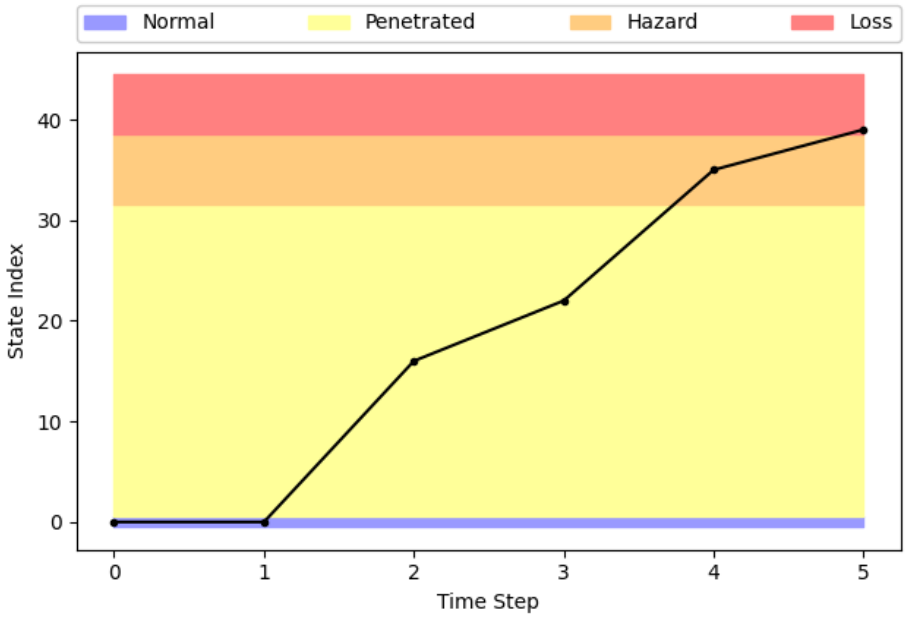


Figure 34: State trajectory of the example game played against θ_A^4.

PLC-1A and PLC-2A are penetrated. In comparison to the example for θ_A^3, other penetrated states occur more frequently. Reactor trip (state 35) was the hazard that occurred the most frequently and loss of power generation (state 39) was the loss that occurred most frequently.

The cumulative utilities of the defender and θ_A^4 are plotted on a logarithmic scale in Figure 36. The interpretation of this figure is similar to those for the games played against θ_A^2 and θ_A^3 shown in Figure 24 and Figure 30, respectively. The attacker's expenses and rewards are particularly clear in this graph because of the direct progression from the normal state to the loss state.

The defender's belief about the utility assigned to L_2 by θ_A^4 is shown in Figure 37. The dashed blue distribution is the original truncated normal distribution, the dashed orange distribution is the belief at the final time step of the game, the gray distributions are beliefs at the intermediate time steps, and the vertical dashed line is the true utility. The estimates extracted from these distributions are plotted in Figure 38. The estimation performed well with a final value of 97.6% of the true value.

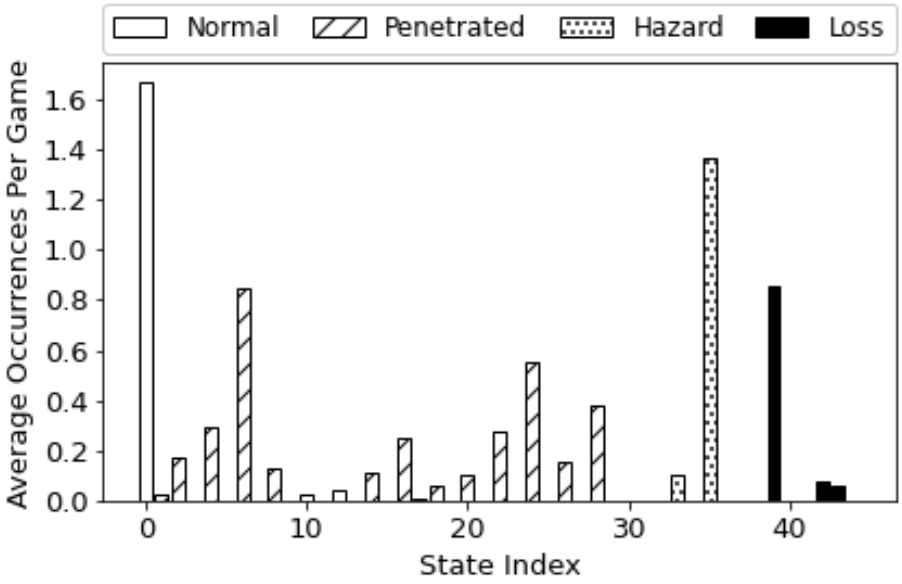


Figure 35: Average occurrences of each state when HBA is used against each θ_A^4.

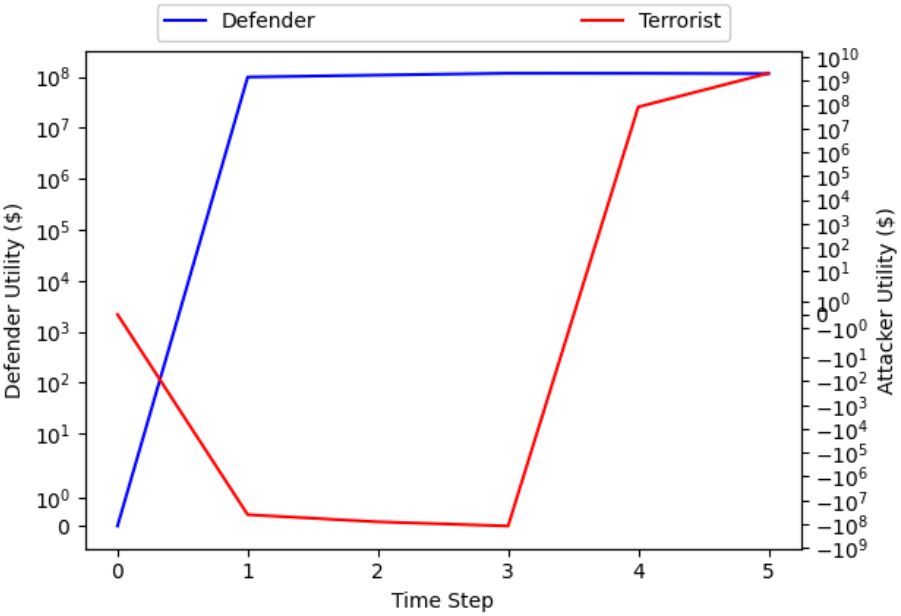


Figure 36: Players' utilities from the example game played against θ_A^4.

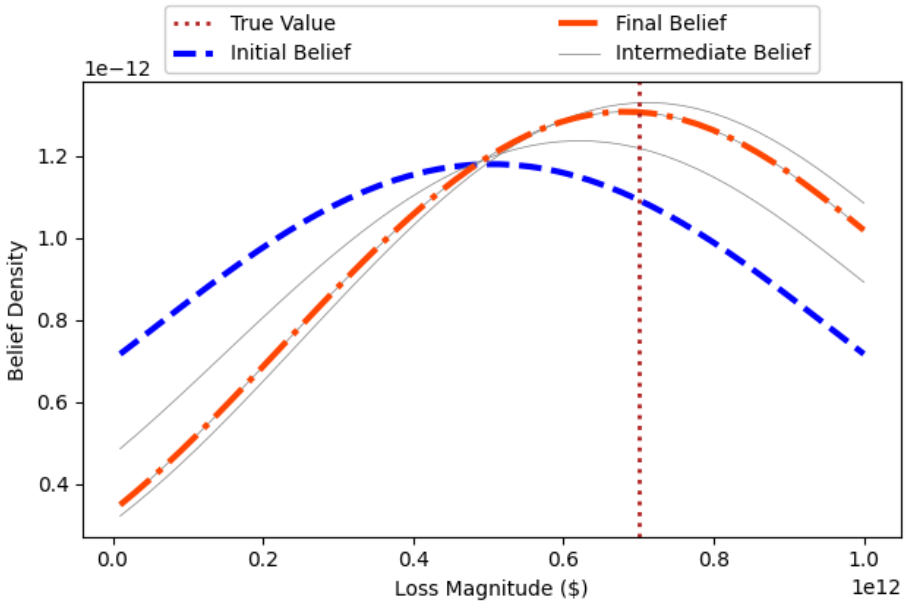


Figure 37: The defender's belief distributions over the range of the attacker's possible utilities assigned to L_2 from the example game played against θ_A^4.

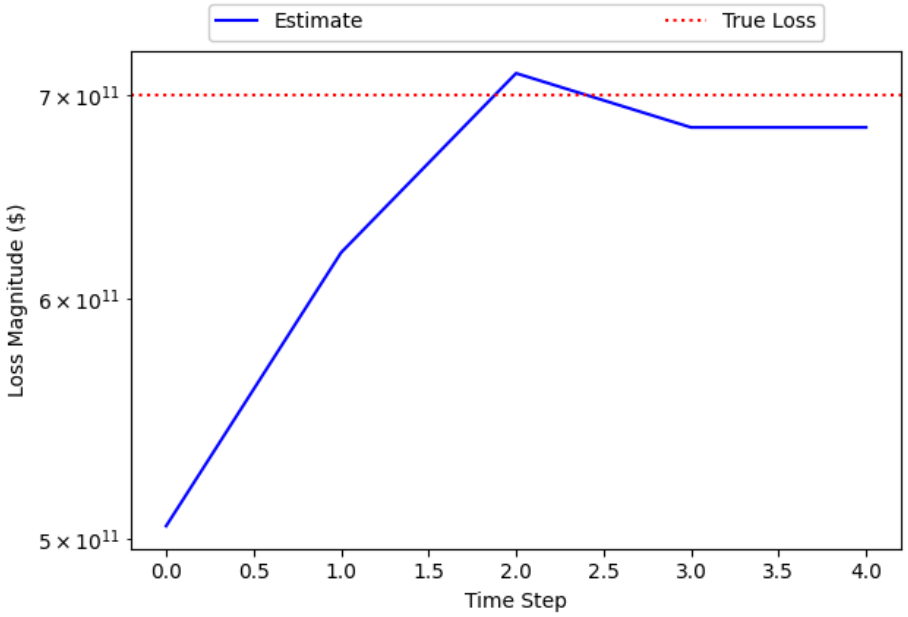


Figure 38: The defender's estimate of the attacker's utility of L_2 from the example game played against θ_A^4.

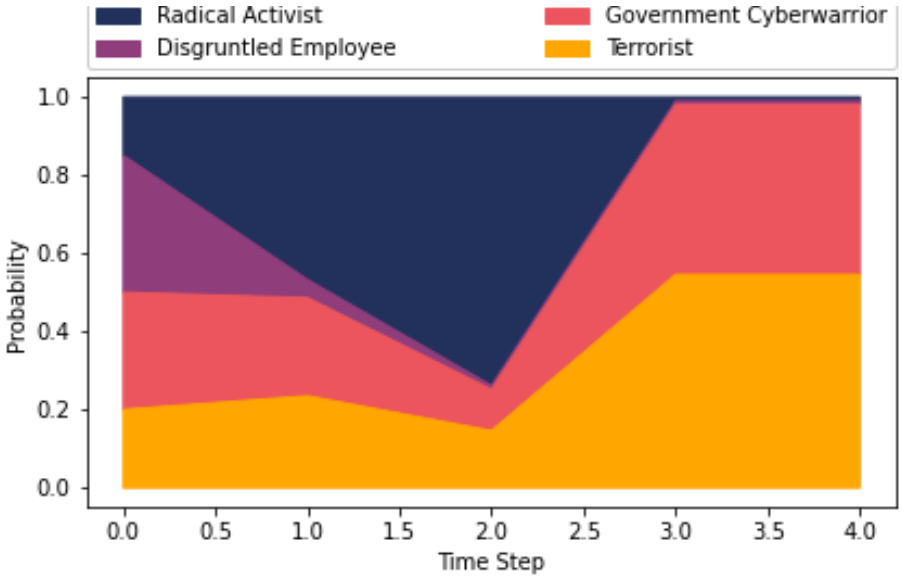


Figure 39: The defender's beliefs regarding the attacker's type from example game played against θ_A^4.

The defender's beliefs regarding the attacker's type while facing θ_A^4 are shown in Figure 39. The beliefs initially favored θ_A^1, but corrected as the game continued. At the conclusion of the game, the beliefs assigned the greatest probability to the correct type, but were not able to rule out θ_A^3.

6.0 Results and Discussion

This chapter describes the simulation results for the SBG constructed in Chapter 4. We simulated the game 500 times for each attacker type. From these simulations, we generate several security metrics and analyze the performance of the defender's Bayesian learning during the game.

6.1 Security Metrics

We can examine several security metrics using the SBG. The first metric is the time-to-loss. This metric is the time for the game to progress from the normal state to an absorbing loss state. The second metric is the availability. Availability is the percentage of time during which the NPP can operate as intended. The third metric is the defender's utility. This metric is the quantification of the defender's performance throughout the entire game. The final metric is the attacker's utility, which quantifies the attacker's performance throughout the game.

6.1.1 Mean Time-to-loss

The simulation times are plotted in Figure 40. On average, games played against θ_A^1 lasted at least twice as long as games played against the other types. For most types, long-lasting games are relatively rare, but they are common for θ_A^1. One reason for this is because θ_A^1 has a lower success rate for most actions than the other types. Type θ_A^3 has the highest success rates and the shortest average simulation. Types θ_A^2 and θ_A^4 also have high success rates for many actions, particularly θ_A^2 for actions that benefit from insider access to the NPP.

It is important to note that termination of the game by reaching a loss is not necessarily suboptimal. The goal of HBA is to maximize the defender's cumulative reward, given his

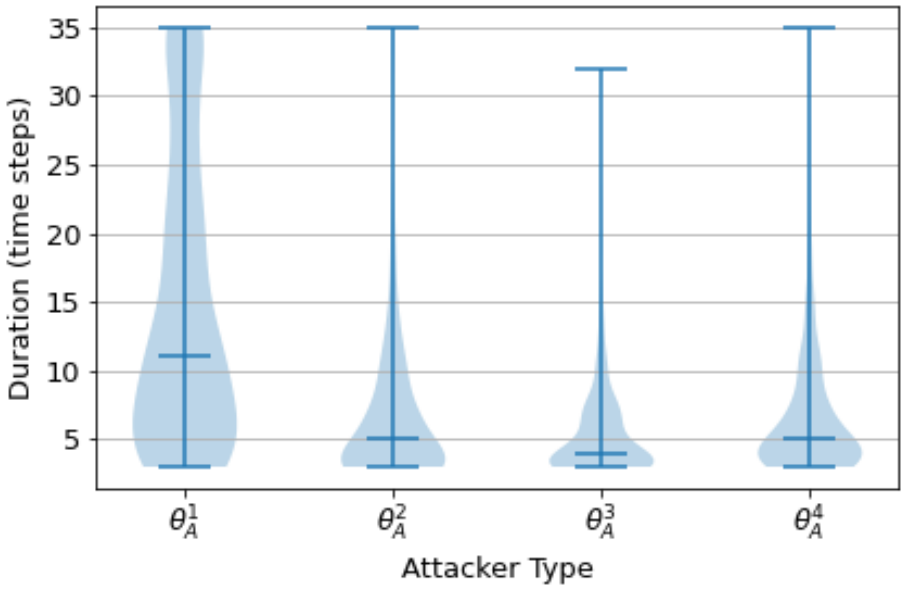


Figure 40: SBG duration from using HBA against each attacker type.

beliefs about the attacker. The utilities assigned to the losses are critical inputs to HBA's decision-making. If an NPP stakeholder viewed the frequency of these losses as unacceptable, that is a sign that there are errors in the quantification of the losses' utilities. The utility of a loss is often dependent on both objective factors, such as the costs of equipment and labor, and subjective factors, such as reputation within the industry and societal obligations. If the time-to-loss results are unacceptable, the subjective loss utility factors should be re-examined.

6.1.2 Mean Availability

The typical condition of the plant throughout a game is given in Figure 41. The most time is spent in the normal state when playing against θ_A^2, and the least amount of time is spent in the normal state when playing against θ_A^4. More time is spent in the penetrated states when playing against θ_A^4 than when playing against the other types. When playing against θ_A^1, a relatively large amount of time is spent in hazard states. The relatively small

amount of time spent in loss states when playing against θ_A^1 is because the average game against θ_A^1 is longer than against the other types, and the loss states are absorbing and can only occur once per game.

We define availability as the capacity to operate the NPP at the expected capacity. The NPP meets this criteria while the game is in the normal state or a penetrated state. The greatest percentage of availability occurs when facing θ_A^4. The percent availability when facing θ_A^4 is 67.5% while the percent availability for the other types is approximately 60–62%.

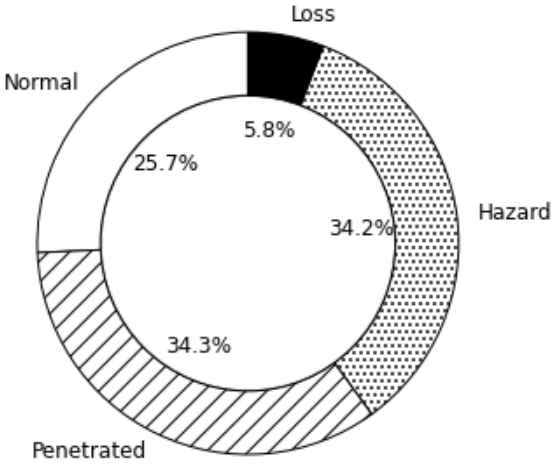
6.1.3 The Defender's Cumulative Utility

The defender's cumulative utilities are plotted in Figure 42. The defender's utilities are skewed because of rare highly costly losses. The defender had the greatest median utility when facing θ_A^1 ($\$1.68 \times 10^8$) and the median utility when facing all other types was on the order of $\$10^7$.

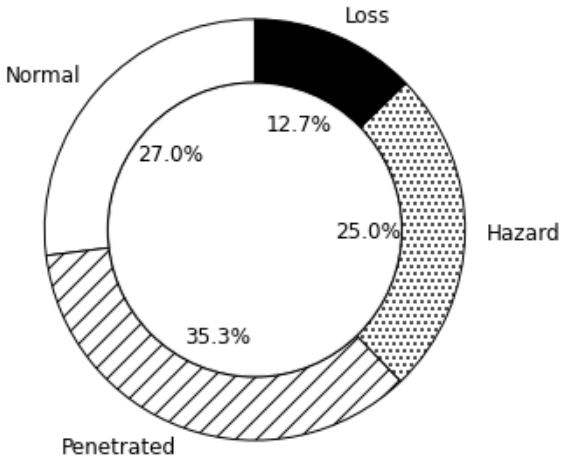
Figure 43 shows the percentage of simulations that resulted in positive utility for the defender. Notably, although the defender's median cumulative utility is greatest when facing θ_A^1, facing θ_A^1 also corresponds to the smallest percentage of simulations that end with positive utility for the defender. HBA does not assign any special consideration to the number zero. In other words, a change in utility from -$1 to $1 is equally as valuable as a change in utility from -$5 to -$3 or from $3 to $5. If obtaining a positive utility is of importance to the defender, the utility function can be transformed to model this preference.

6.1.4 The Attacker's Cumulative Utility

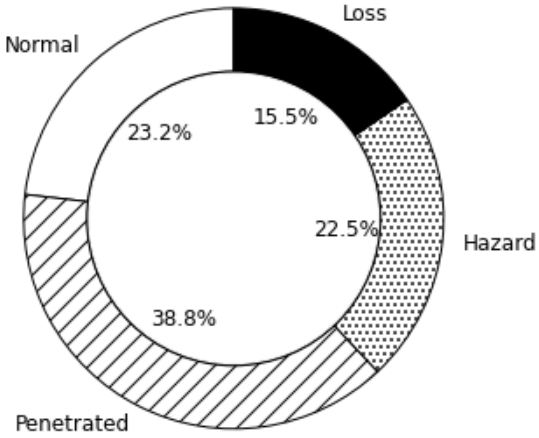
The attacker's cumulative utilities are plotted in Figure 44. The utilities are plotted in separate histograms because comparison of utility values between players and between types is generally improper. The utility of type θ_A^1 had the greatest relative standard deviation. The utilities of types θ_A^2 and θ_A^3 had smaller relative standard deviations, and the utility for θ_A^4 was the most consistent. It should be noted that minimizing the utility of each attacker type is not the defender's objective. The defender's objective is to maximize his own utility.



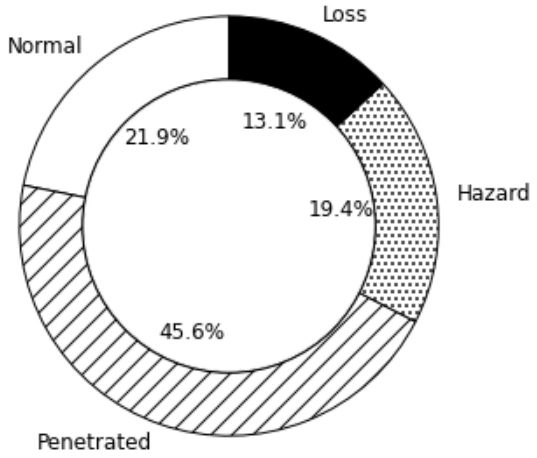
(a) Facing θ_A^1.



(b) Facing θ_A^2.



(c) Facing θ_A^3.



(d) Facing θ_A^4.

Figure 41: Relative time spent in each plant condition when HBA was used against each attacker type.

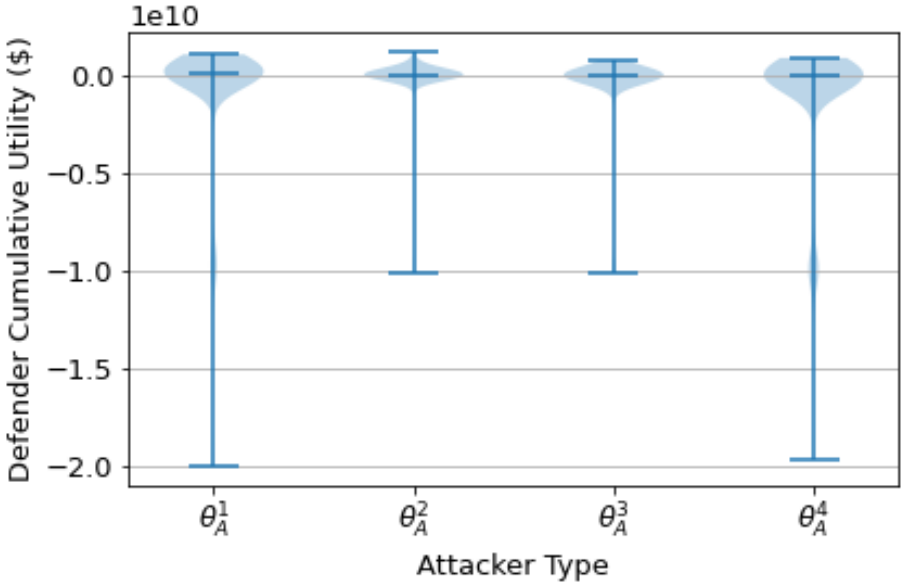


Figure 42: The defender's cumulative utility when HBA is used against each attacker type.

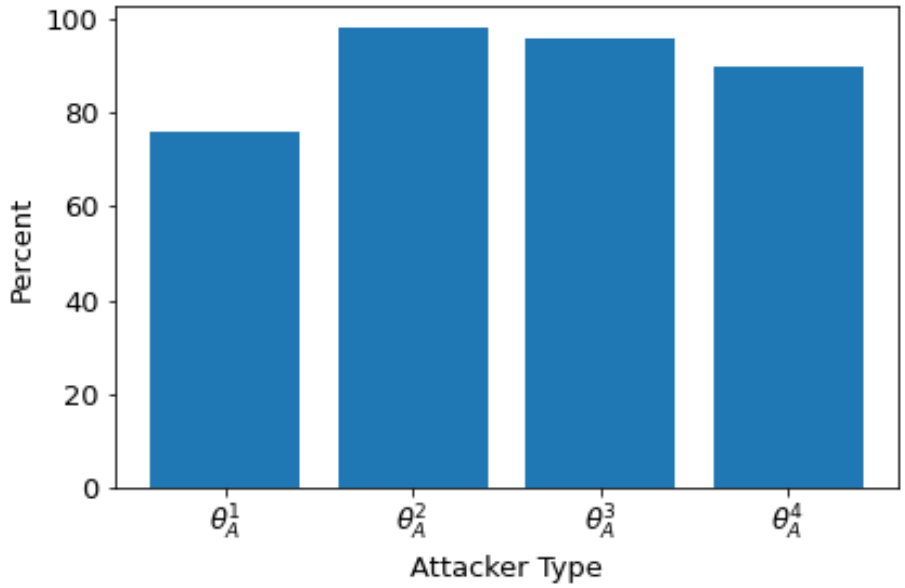
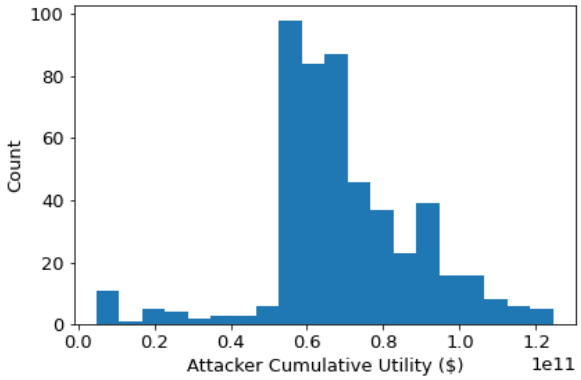
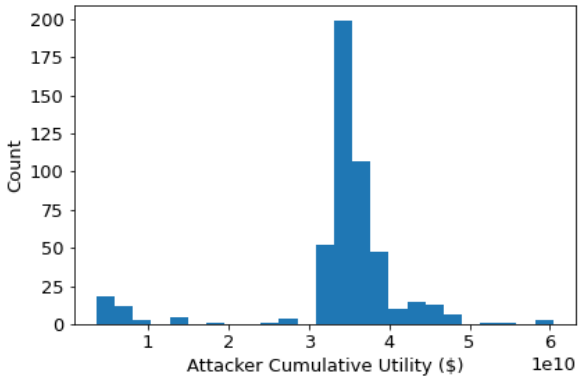


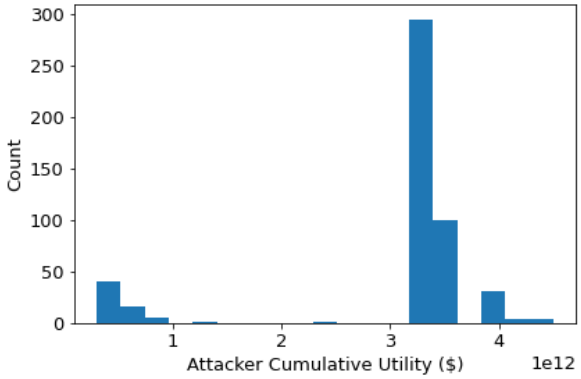
Figure 43: The percentage of simulations where the defender's cumulative utility is positive when HBA is used against each attacker type.



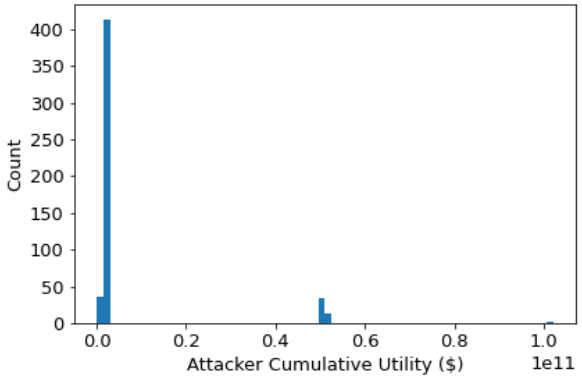
(a) Facing θ_A^1.



(b) Facing θ_A^2.



(c) Facing θ_A^3.



(d) Facing θ_A^4.

Figure 44: The attacker's cumulative utility when the defender uses HBA against each attacker type.

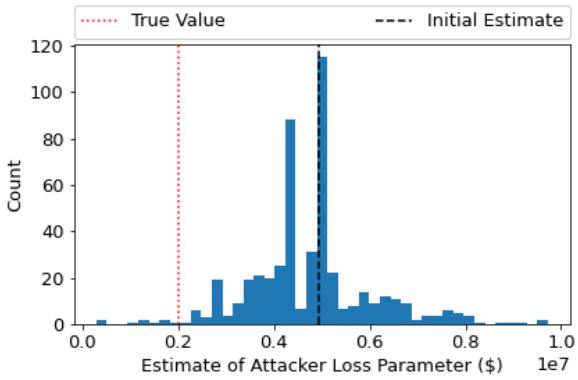
6.2 Bayesian Learning of the Attacker's Loss Utility

As the game is played, the defender draws inferences about the utility that the attacker assigns to loss L_2. Histograms of these estimates are given in Figure 45. Estimates of L_2 for θ_A^1 tended to progress towards the true value, but not converge. Estimates of L_2 for θ_A^2 and θ_A^3 made little progress from the initial estimate. Estimates for θ_A^4 were the most accurate.

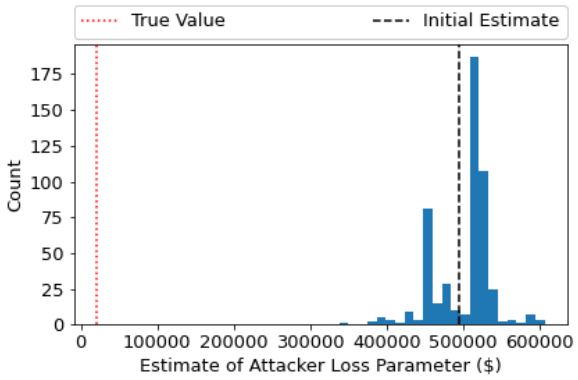
The first factor affecting the learning of L_2 is the short duration of most games. For most attacker types, the median duration of a game was less than five time steps. It is difficult to converge to the correct value of L_2 when interactions between the players are limited. This factor had the greatest affect when playing against θ_A^1 and θ_A^4. In these cases, the estimates often progressed towards the true value without converging.

The second factor affecting the learning of L_2 was a lack of new information as the game was played. For example, choices made in hazard states were defined to be independent of the loss utilities, therefore inferences cannot be made about L_2 in those states. Another example is when the game enters a state multiple times and the attacker chooses the same action each time. No new information is obtained in those interactions. These issues arise specifically from the attacker's decision-making process defined in Algorithm 2. Other decision-making processes may not encounter the same challenges. This factor affected the estimation of L_2 when playing against all types.

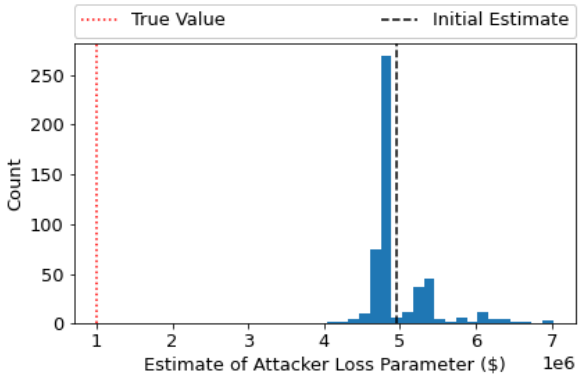
The third factor affecting the learning of L_2 is that the attacker's decisions are sometimes not sensitive to changes in L_2. The utility of L_2 can affect the attacker's decision-making at two points in Algorithm 2. The first point in the algorithm is when the attacker ranks the hazards by their expected utility. The expected values of each hazard are plotted as a function of the utility of L_2 in Figure 46. For types θ_A^1, θ_A^2, and θ_A^3, the ranking of hazards does not change over the range of possible utilities of L_2. For θ_A^4, there are four regions over the range of L_2 that correspond to different hazard rankings. This likely contributed to greater success in estimating L_2 for θ_A^4 in comparison to the other types. The second point in the algorithm is when the attacker assigns probability to each action, with the goal of achieving a particular hazard. If L_2 is of significant utility to the attacker, the attacker is more likely to select an action that leads to hazards that can cause L_2. For θ_A^4, L_2 has



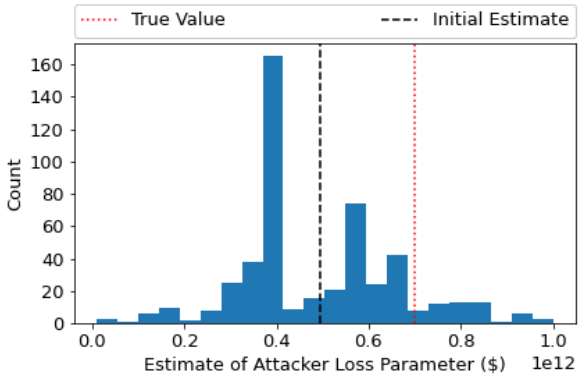
(a) Facing θ_A^1.



(b) Facing θ_A^2.



(c) Facing θ_A^3.



(d) Facing θ_A^4.

Figure 45: The defender's final estimate of the attacker's loss utility when the defender uses HBA against each attacker type.

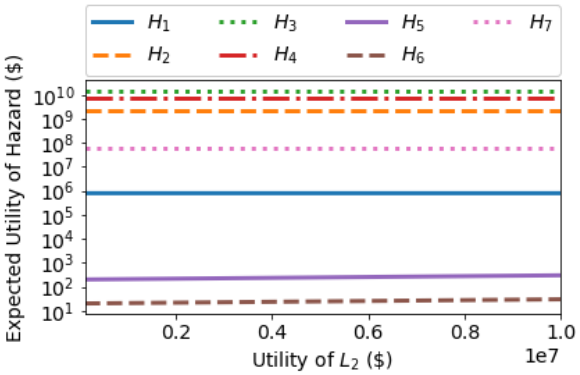
significant utility relative to the other losses, but for the other types, the utility of L_2 is less significant. For this reason, the utility of L_2 is less likely to influence the decision-making of those types.

Little can be done to improve the estimation of the utility parameter in cases where the parameter does not significantly affect the attacker's decision-making. But, if the parameter does not significantly affect the attacker's decision-making, it is of little use to the defender to attempt to estimate it. In this work, L_2 was selected for all attacker types because the occurrence of L_2 was of significant importance to the defender, because the utility of L_2 varied significantly between attacker types, and for the sake of consistency. For future work it is recommended to create plots similar to those in Figure 46 for each loss to identify the parameters that are most likely to affect each type's decision-making.

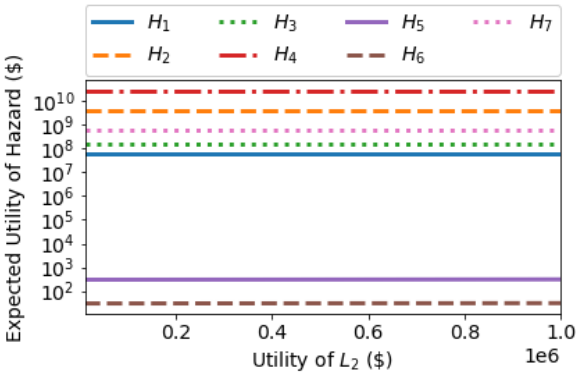
6.3 Estimating the Attacker's Type

As the game is played, the defender updates his beliefs about which attacker type he is facing. Histograms of these beliefs are given in Figure 47. The true type was consistently identified when facing θ_A^1 and θ_A^2. When playing against θ_A^3, the true type was assigned the greatest probability, but the probability never exceeded 0.71. The true type was most challenging to identify when facing θ_A^4.

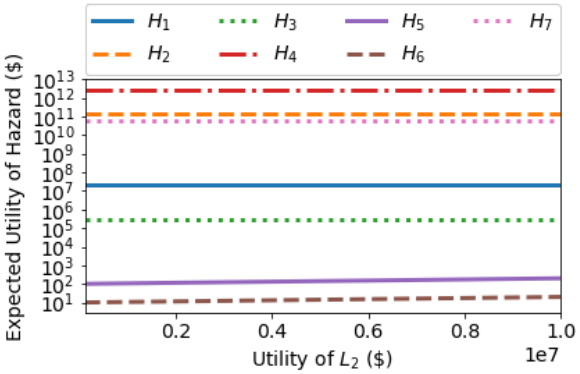
Type θ_A^1 was most straightforward to identify because θ_A^1 places significant value on L_4 relative to the other attacker types. Although θ_A^2 did not have a particularly unique valuing of the losses, θ_A^2 was also straightforward to identify because θ_A^2 had the propensity to leverage insider access to compromise the switch. Types θ_A^3 and θ_A^4 were not as easy to identify. This is because types θ_A^3 and θ_A^4 have similar desired losses and similar skill levels.



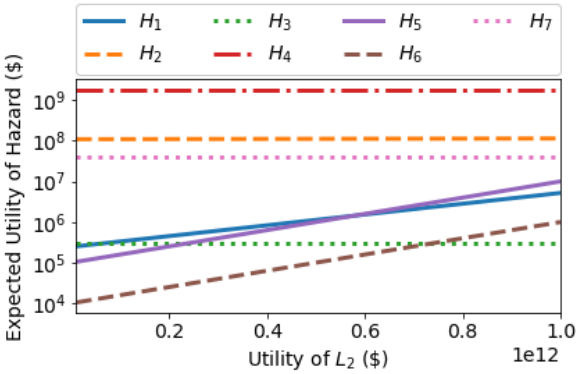
(a) Expected value of hazards for θ_A^1.



(b) Expected value of hazards for θ_A^2.

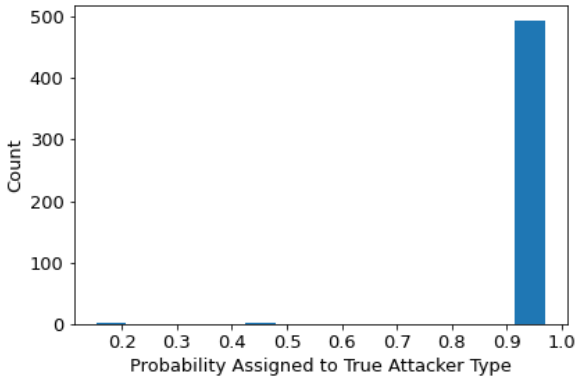


(c) Expected value of hazards for θ_A^3.

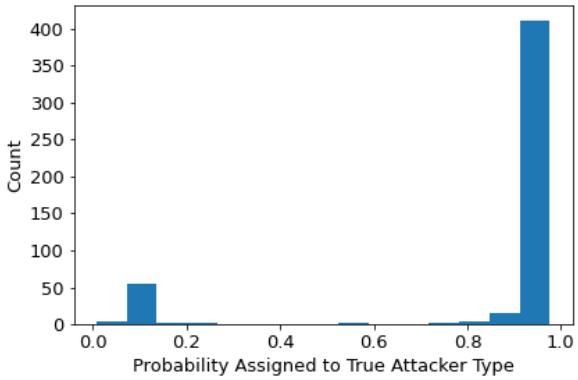


(d) Expected value of hazards for θ_A^4.

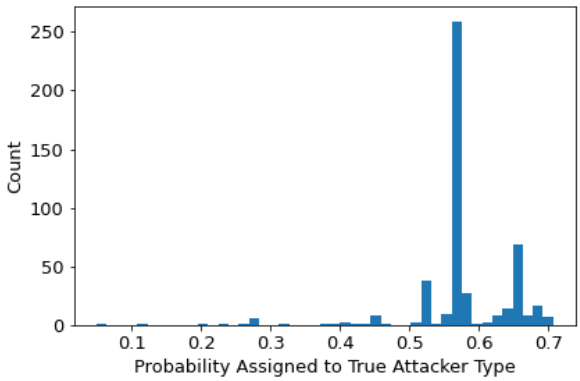
Figure 46: The expected value of each hazard as a function of the utility of L_2. The hazard preferences of θ_A^4 change as a function of L_2.



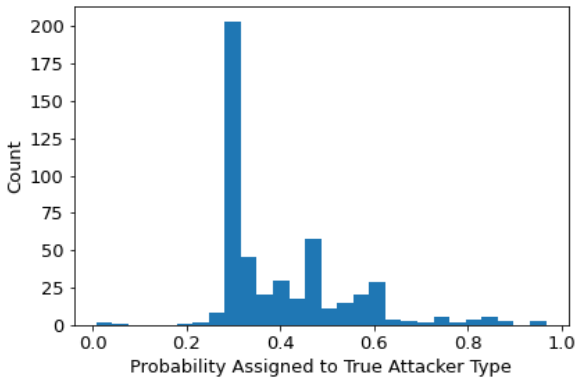
(a) Facing θ_A^1.



(b) Facing θ_A^2.



(c) Facing θ_A^3.



(d) Facing θ_A^4.

Figure 47: The defender's final estimate of the probability of the attacker's true type when the defender uses HBA against each attacker type.

7.0 Conclusions and Future Work

The goal of this research is to reduce the likelihood of successful attacks on NPPs. This goal is achieved through the following research objectives:

1. Predict how an adversary might target a nuclear power plant
2. Quantify nuclear power plant security
3. Optimally allocate security resources to defend a nuclear power plant

The first objective was met through the work in Chapter 4 and the real-time learning components of HBA. In Chapter 4, several approaches were presented to construct an SBG. The approaches included modeling attack progressions through the NPP as stochastic elements of the SBG and modeling threats to NPP as Bayesian elements of the SBG. These approaches can also be used to construct stochastic games and Bayesian games separately. The real-time learning components of HBA were used to estimate one of the attacker's utility parameters and to identify the attacker's true type. These efforts had mixed success for reasons discussed in Chapter 6. The limitations of these learning approaches are also discussed in Section 7.2 below.

The second objective was met through the simulation of the SBG discussed in Chapter 6. The first security metric is the mean time-to-loss. The time-to-loss is the time that elapses from the beginning of the game in the normal state to the termination of the game in a loss state. The second security metric is the mean availability. The availability is the percentage of time during which the NPP can operate as intended. The third security metric is the defender's cumulative utility. The defender's cumulative utility is a number that quantifies the defender's performance throughout the game. The utility is a combination of objective economic factors and subjective factors such as societal obligations. The attacker's cumulative utility can also be considered a security metric, but if the attacker's utility is of importance to the defender, that should be included in the construction of the defender's utility function.

The third objective was met through the implementation of HBA. The purpose of HBA is to maximize the defender's cumulative utility given the defender's beliefs about the attacker. Unlike the static predictive approach of many other game-theoretic methods, HBA leverages the defender's knowledge to make decisions in real-time. As the game is played, the defender can draw inferences about the attacker and incorporate the most current beliefs into HBA to optimize security decisions.

7.1 Summary of Contributions

The main contributions of this work to the fields of game theory and NPP cybersecurity are:

1. an approach to characterize threats to NPPs and model them as attacker types in a Bayesian game
2. an approach to construct the state space of a stochastic security game
3. an approach to define the transition function of a stochastic security game
4. a novel application of stochastic Bayesian games to cybersecurity challenges
5. methods to approximate Harsanyi-Bellman ad hoc coordination solution methods for stochastic Bayesian games with large action spaces

7.2 Limitations

The first limitation of this research is that this approach can be computationally expensive. The most significant cost arises from a large action space for the players, particularly for the defender. At each time step, each defender action must be analyzed by HBA to identify the optimal security action. This involves sampling paths resulting from each action to estimate the defender's utility resulting from that action. Care should be taken to limit the number of decisions to be made by the defender and to sample action paths in an efficient manner.

The second limitation of this research is that the Bayesian learning efforts are dependent on the defender's ability to define potential decision-making processes for the attacker types. If the defender cannot define the attacker's decision-making method, it is possible that HBA could learn incorrect parameters and types, and perform poorly.

The third limitation of this research is that learning of the attacker's parameters and type may not be feasible for some situations. These situations are:

1. When the interactions between the attacker and defender are of a short duration. Short games limit the amount of information that can be gathered by the defender, and therefore limit the inferences that can be drawn from that information.
2. When the attacker's decision-making is not sensitive to the game-theoretic parameters of interest. If the attacker's decisions are not significantly affected by the parameter, the defender will be unable to draw inferences about the parameter from the attacker's actions. If this is the case, the inference of this parameter is not important for the defender's decision-making process. But, if estimating the parameter is important for other reasons such as intelligence gathering, other methods would need to be pursued.
3. When a subset of the attacker types behave similarly. If multiple types select similar actions when exposed to the same situation, it will be challenging for the defender to distinguish them. If this is the case, the distinction between the types may not be of significance for that application. In fact, it may be beneficial to consolidate those types to reduce computational costs. But, if identifying the true attacker type is important for other reasons such as intelligence gathering, other methods would need to be pursued.

7.3 Future Work

The SBG model could be improved with future work regarding cybersecurity modeling. For example, the SBG in this work models a device as being either penetrated or normal. In reality, the condition of the device is not binary. An attacker could gain a variety of privileges when compromising an ICS device, and modeling those privileges within the state space of an SBG could provide greater insight.

Future work could also improve the SBG model by including more sophisticated economic modeling of the attacker's and defender's expenses. For example, in this work all actions were assigned a specific cost regardless of the state or game history. In reality, the costs for security actions may vary over time or be dependent on previous actions. For example, some actions may have larger initial costs and low maintenance costs, while other actions may have the opposite.

Future work related to the implementation of HBA could address computational efficiency. For example, more sophisticated path sampling algorithms could be applied to ensure an appropriate sample without significant additional costs. If this is accomplished, larger decision spaces could be analyzed that are closer to the complex cybersecurity challenges faced in industry.

Appendix A

Observability Attacks and Game Theory

A game-theoretic approach is presented to analyze observability attacks. The attacker's strategy set includes all possible combinations of masked measurements. The defender's strategy set includes all possible combinations of measurement reinforcements. The attacker's and defender's utilities are quantified using the responses of the observable and unobservable states. The observability attack game is analyzed for a nuclear balance of plant system. Multiple pure-strategy and mixed-strategy Nash equilibria are identified, and the conditions for their existence are presented. Using this procedure, a security and control engineer can select the optimal strategy to defend a cyber-physical system from observability attacks. The development of this problem and its solution are published in [59, 55, 57].

A.1 Attacker Controllability and Observability

The system under attack, G, is described by the linear system,

$$G : \begin{cases} \dot{x} &= Ax + Bu \\ y &= Cx \end{cases} \sim \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right] \tag{A.1}$$

where $x \in \mathbb{R}^n$ is the state variable, $u \in \mathbb{R}^q$ is the input to which the attacker has access, and $y \in \mathbb{R}^p$ is the measured output. The matrix $A \in \mathbb{R}^{n \times n}$ is the dynamics matrix. The matrix $B \in \mathbb{R}^{n \times q}$ is the input matrix, and $C \in \mathbb{R}^{p \times n}$ is the output matrix; these matrices describe how inputs enter the system and how measurements relate to the internal state variables. It should be noted that a system representation is not unique. While many systems are

nonlinear, most systems may be linearized about an operating point. This is particularly true for nuclear power systems that often operate continuously at steady-state conditions. For more information regarding linear systems, readers are encouraged to refer to [17].

When analyzing cyber-attack scenarios, we must consider two factors: the attacker's ability to affect the state of the system, and the attacker's ability to mask the state of the system. The system theory concepts of controllability and observability enable us to address these considerations.

We assume that the nominal system, the one unaffected by the attacker, is both controllable and observable. That is, if an attacker has access to the inputs of a controllable system, the system can be driven to any state. All states of the nominal, observable system can be reconstructed by a defender.

A system is controllable if it is possible to find some input, u, that can steer the state, x, to any desired value in finite time. Testing for controllability is straightforward [17]:

Test 1 (Controllability)**.** *A linear system with representation given in Eq.* (A.1) *is controllable if and only if the matrix*

$$\mathcal{C} = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix} \tag{A.2}$$

is full row rank.

Similarly, a system is observable if the state, x, can be determined from the observation of y in finite time. The test for observability is similar to that for controllability:

Test 2 (Observability)**.** *A linear system with representation given in Eq.* (A.1) *is observable if and only if the matrix*

$$\mathcal{O} = \begin{bmatrix} C' & (CA)' & (CA^2)' & \cdots & (CA^{n-1})' \end{bmatrix}' \tag{A.3}$$

is full column rank.

The tests for controllability and observability both depend on the rank of a matrix. Rank is defined as the number of linearly independent rows (columns) in a matrix. We require a more practical test for situations when numerical issues must be considered. A more practical test for rank compares the singular values of the matrix to some positive tolerance [32].

Test 3 (Matrix Rank)**.** *The rank, r, of a matrix, X, can be determined from the singular values, σ_i, of X according to the inequalities*

$$\sigma_1 \geq \cdots \geq \sigma_r > \Delta \geq \sigma_{r+1} \geq \cdots \geq \sigma_n \tag{A.4}$$

A matrix is full row rank if r is equal to the number of rows in the matrix.

The tolerance, Δ, is defined to be consistent with the precision of the problem, ε. The problem precision is dependent on the precision of the data in matrix X. If the data in X has infinite precision, then ε is equal to the machine precision — the smallest difference between two numbers that a computer can recognize. If the data in matrix X has finite precision (e.g. it is obtained experimentally), then ε is equal to the precision of the data. For a matrix X, the tolerance Δ is defined by

$$\Delta = \varepsilon \left\| X \right\|_\infty \tag{A.5}$$

where the matrix ∞-norm is the maximum absolute row sum of the matrix,

$$\left\| X \right\|_\infty = \max_i \sum_{j=1}^{n} \left| x_{ij} \right|. \tag{A.6}$$

The primary advantage of this rank test is that singular values are easy to compute and many singular value decomposition algorithms exist.

The masking of measurements by an attacker results in the elimination of rows from C, thereby affecting the rank of the observability matrix in Test 2. It is possible that without these measurements, a portion of the state space would be made unobservable by the attacker; that is, even with knowledge of the system, a portion of the system state cannot be reconstructed or estimated. An intelligent attacker could mask a particular subset of system measurements, and then directly target those states that are unobservable.

Examining the structure of the state space provides insight to how the states are related to the measurements and therefore how an attacker might mask an attack. A Kalman decomposition is a transformation of the system that partitions the states according to

their controllability and observability. There exists a transformation, T, that transforms the system representation to the following Kalman decomposition:

$$\left[\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & D \end{array}\right] = \left[\begin{array}{c|c} T^{-1}AT & T^{-1}B \\ \hline CT & D \end{array}\right] = \left[\begin{array}{cc|c} A_1 & A_{12} & B_1 \\ 0 & A_2 & B_2 \\ \hline 0 & C_2 & D \end{array}\right] \tag{A.7}$$

The details of how to construct this transformation can be found in [14]. In this representation, the state space has been partitioned into the unobservable states $x_1 \in \mathbb{R}^{n_1}$ and the observable states $x_2 \in \mathbb{R}^{n_2}$. It is clear that the unobservable states are not seen in the measurement y. Because of the structure of the dynamics matrix, the unobservable states do not affect the observable states, and remain hidden even when the system model is known.

A.2 Stealthy Observability Attacks

When a system is observable, internal state variables can be estimated by a state observer using a system model, control inputs, and system measurements. By masking system measurements, an attacker may cause part of the state space to become unobservable. Unobservable states cannot be reconstructed by state observers or operators. An intelligent attacker can mask specific measurements to cause certain states of interest to become unobservable — this is called an observability attack [59].

Given that an attacker has managed to render a portion of the state space unobservable, we must now consider how the attacker can affect those states. We will consider a specific type of input, an impulse excitation with the vector input,

$$u(t) = v\delta(t), \tag{A.8}$$

where the vector $v \in \mathbb{R}^q$ is a vector amplitude of the impulse; the impulse is modeled by the Dirac delta function. Such an input is attractive because it is transient and short-lived;

thus, it might be missed by operators. For a unit impulse excitation

$$\delta(t) = \lim_{\tau \to 0} \begin{cases} 1/\tau & 0 < t < \tau \\ 0 & \text{otherwise} \end{cases} \tag{A.9}$$

This is appropriate for excitations that pulse the system, but the pulse lasts for a duration, τ, much less than the system's characteristic time constant, T_c: $\tau \ll T_c$. For multi-input systems, both the magnitudes and signs of each element of the vector input, v, are important. In the case of a power plant, these impulses could be the large magnitude short-duration pulses of a valve, pump, or switch.

The attacker's objective is to choose v to maximize the response of the unobservable states, x_1, while minimizing the response of the observable states, x_2. By achieving this objective, the attacker can maximize the damage to the unobservable portion of the plant while remaining undetected.

The magnitudes of the unobservable and observable responses to the impulse are

$$\text{Unobservable} \qquad \|x_1^*\|^2 = v'(B'Q_1B)v \tag{A.10}$$

$$\text{Observable} \qquad \|x_2^*\|^2 = v'(B'Q_2B)v \tag{A.11}$$

where Q_1 and Q_2 are solutions to the Lyapunov equations

$$A'Q_i + Q_iA + C_i'C_i = 0, \qquad i = 1, 2 \tag{A.12}$$

for the unobservable and observable subsystems respectively.

Once the attacker has rendered a portion of the state space unobservable, the attacker can modify control inputs to drive those states to undesirable values.

$$\|x_1\| > \text{limit} \quad \Rightarrow \quad \text{damage.} \tag{A.13}$$

To avoid detection, the attacker must also limit the effect of the modified control inputs on the states that remain observable.

$$\|x_2\| < \text{threshold} \quad \Rightarrow \quad \text{undetectable.} \tag{A.14}$$

If the attack limits the response of the observable states while controlling the unobservable states, it is said to be stealthy.

The stealth of an attack can be quantified using a metric called the stealth ratio. This ratio compares the magnitude of the unobservable response to that of the observable response. The details of how to construct this metric can be found in [59].

$$\text{SR} = \frac{\|x_1^*\|}{\|x_2^*\|} \tag{A.15}$$

An attack is considered stealthy if $\text{SR} \gg 1$.

An attempted observability attack can be described using three stealth categories:

1. Observable: The entire state-space remains observable and the entirety of the attack can be seen.
2. Unstealthy: A portion of the state-space has been rendered unobservable, but the response of the unobservable states is small relative to the response of the observable states.
3. Stealthy: A portion of the state-space has been rendered unobservable, and the response of the unobservable states is large relative to the response of the observable states.

An observability attack can be analyzed using game theory, where the attacker and defender's utilities are functions of the attack stealth.

A.3 Game-Theoretic Approach

Using game theory, we will analyze the observability attack and select an effective defense. There are three components to a game:

1. *Players:* These are the individuals or entities participating in the game. The observability attack will be formulated as a two-player simultaneous game. The two players are an attacker targeting the plant and a defender protecting the plant. All parameters pertinent to the attacker will be denoted with a subscript, A, and all parameters pertinent to the defender will be denoted with a subscript, D.

2. *Strategies:* These are the actions taken by the players in the game.

   a. *Pure Strategy:* The ith strategy of the defender is denoted s_D^i. The entire set of strategies available to the defender is denoted $\mathcal{S}_D = \{s_D^i | i = 1 \ldots N\}$. The attacker's strategies, s_A^i, and strategy set, $\mathcal{S}_A$, are similarly defined.

   b. *Mixed Strategy:* A mixed strategy is a probability distribution over a player's pure strategy set. Let σ_D denote a mixed strategy of the defender. Let $\sigma_D(s_D)$ denote the probability of the defender playing pure strategy s_D. The support of σ_D is the set of pure strategies to which σ_D assigns a positive probability. Let σ_A and $\sigma_A(s_A)$ be similarly defined for the attacker. There are an infinite number of possible mixed strategy profiles.

3. *Utilities:* These are the payoffs to the players that result from the strategy profile of the game. In other words, each player's utility is dependent on both the strategy that they employ and the strategy of their opponent. The defender's utility for playing strategy s_D^i when the attacker plays strategy s_D^j is $\pi_D^{ij} = \pi_D(s_D^i, s_A^j)$. The attacker's utility, π_A^{ij}, is similarly defined.

A matrix of both players' strategies and their resulting utilities is constructed to analyze the game; see table 25. Each row in the utility matrix corresponds to one of the defender's strategies, s_D^i, and each column corresponds to one of the attacker's strategies, s_A^j. Each entry in the matrix shows the defender's utility, π_D^{ij}, and the attacker's utility, π_A^{ij}. Using these quantified utilities, we can evaluate the observability attack game to determine which defense strategy is likely to yield the greatest benefit.

It should be noted that these utility parameters could be a function of risk metrics determined by another formal risk analysis. The pairing of these game-theoretic techniques and traditional risk assessment methodologies such as probabilistic risk assessment may provide greater insight for complex systems. This is particularly appropriate for nuclear power systems, which have a long history of using probabilistic risk assessment tools to evaluate the safety of plants [80].

Using the process of iterated elimination of dominated strategies, we can eliminate strategies that no rational player would select. One of the defender's strategies, s_D^i, is

Table 25: An example of the observability attack game. The utilities for the attacker and defender are provided for each intersection of defender and attacker strategies.

		Attacker			
		s_A^1	s_A^2	$\cdots$	s_A^M
Defender	s_D^1	π_D^{11}, π_A^{11}	π_D^{12}, π_A^{12}	$\cdots$	π_D^{1M}, π_A^{1M}
	s_D^2	π_D^{21}, π_A^{21}	π_D^{22}, π_A^{22}	$\cdots$	π_D^{2M}, π_A^{2M}
	$\vdots$	$\vdots$	$\vdots$		$\vdots$
	s_D^N	π_D^{N1}, π_A^{N1}	π_D^{N2}, π_A^{N2}	$\cdots$	π_D^{NM}, π_A^{NM}

said to be dominated by another strategy, s_D^j, if s_D^j yields the defender a utility at least as great as that yielded by s_D^i for each of the attacker's strategies. That is, s_D^i is dominated by s_D^j if,

$$\pi_D(s_D^j, s_A^k) \geq \pi_D(s_D^i, s_A^k) \quad \forall s_A^k \in \mathcal{S}_A \tag{A.16}$$

A dominated strategy can be eliminated from the game because no rational player would select it. In the process of iterated elimination of dominated strategies, all dominated strategies are eliminated for one player, then for the next player. After the first round of elimination, there may be strategies that were not dominated in the first round, but are dominated in the reduced form of the game. These strategies can be removed from the game in a second round of elimination. Iterated elimination of dominated strategies continues until none of the players have any dominated strategies.

An effective defense strategy can be selected using the concepts of a best response and a Nash equilibrium. A player's best response to an opponent's strategy is the strategy that will yield the greatest utility for that player [22]. A defense strategy s_D^k is the defender's best response to the attacker's strategy s_A^j if

$$\pi_D(s_D^k, s_A^j) \geq \pi_D(s_D^i, s_A^j) \quad \forall s_D^i \in \mathcal{S}_D \tag{A.17}$$

The attacker's best response is similarly defined.

A Nash equilibrium is defined as a strategy profile such that each player's strategy is an optimal response to the other player's strategy. At a Nash equilibrium, neither player has an incentive to deviate from the equilibrium strategy if the other player's strategy remains unchanged. Nash equilibria may include pure and/or mixed strategies. In a mixed-strategy Nash equilibrium, a unilateral deviation to any of the pure strategies in the support of a given player's mixed strategy will yield that player an expected utility equal to the expected utility of playing the mixed strategy [29]. Let a player's mixed strategy at a Nash equilibrium be denoted by σ^*. The Nash equilibrium definition for this game is

$$\pi_D^*(\sigma_D^*, \sigma_A^*) \geq \pi_D(s_D, \sigma_A^*) \quad \forall s_D \in \mathcal{S}_D \tag{A.18}$$

$$\pi_A^*(\sigma_D^*, \sigma_A^*) \geq \pi_A(\sigma_D^*, s_A) \quad \forall s_A \in \mathcal{S}_A \tag{A.19}$$

Note that if the attacker's and/or the defender's strategy at the Nash equilibrium is pure, σ_A^* and/or σ_D^* in the previous inequalities can be replaced with s_A^* and s_D^* as necessary. The expected utilities of both players at a Nash equilibrium are

$$\mathbb{E}[\pi_D^*(\sigma_D^*, \sigma_A^*)] = \sum_{s_D \in \mathcal{S}_D} \sum_{s_A \in \mathcal{S}_A} \pi_D(s_D, s_A)\sigma_D(s_D)\sigma_A(s_A) \tag{A.20}$$

$$\mathbb{E}[\pi_A^*(\sigma_D^*, \sigma_A^*)] = \sum_{s_A \in \mathcal{S}_A} \sum_{s_D \in \mathcal{S}_D} \pi_A(s_D, s_A)\sigma_A(s_A)\sigma_D(s_D) \tag{A.21}$$

Having discussed the foundational elements of game theory, we will now introduce the application of game theory to observability attacks.

A.3.1 Game Overview

In this observability attack game, the defender will attempt to protect the system while the attacker attempts an observability attack. The attacker will choose strategies that mask certain measurements from the defender, thus altering the system's observability. The defender will choose strategies that reinforce certain measurements, thereby thwarting the attacker. The attacker's and defender's utilities both depend in part upon the resulting stealthiness of the plant for the selected pair of strategies. The structure of this game is similar to that used in [55, 56].

A.3.2 The Defender

The set of the defender's strategies includes reinforcing all possible combinations of measurements in the system. Examples of reinforcement include the installation of redundant sensors on a separate network or implementing trusted patches. It is assumed that if the defender has reinforced a sensor that the attacker has targeted, then the attack on that sensor is unsuccessful. It is assumed that the attacker has gained access to all control inputs, therefore the option of defending actuators is not included in the defender's strategy set.

The defender's utilities, π_D^{ij}, are calculated as a function of the attack stealth and the cost of reinforcing measurements.

$$\pi_D^{ij} = -L_D^{ij} - E_D^i \tag{A.22}$$

The attack stealth contributes to the defender's utility through the term L_D^{ij}. This term is dependent on both the strategy of the defender and that of the attacker, and there are three cases depending upon whether the attack is observable, unobservable but unstealthy, or unobservable but stealthy:

1. If the set of successfully masked measurements does not alter the results of the observability test, the attack is observable and the defender incurs a loss of L_D^O.
2. If the stealth ratio is small, the attack is unstealthy and the defender incurs a loss of L_D^U.
3. If the stealth ratio is sufficiently large, the attack is stealthy and the defender incurs a loss of L_D^S.

We assume that $0 < L_D^O < L_D^U < L_D^S$. For ease of demonstration, we have assumed that the defender's utility is dependent on attack stealth and not on which portion of the state space has been stealthily attacked.

In this paper, we assume that the cost of defense is proportional to the number of reinforced measurements. This assumption is applicable to a large class of problems. The fixed cost of defense increases for a greater number of sensors, and variable cost increases

due to greater operation and maintenance costs. In general, while a cost curve may not in fact be linear, this linear assumption is sufficient for this paper and it provides first-order insight to the problem.

The total cost of defending the system is represented by E_D^i. The cost of defending the system is computed as the product of the number of reinforced measurements, n_D^i, for defense strategy s_D^i, and the cost of reinforcing a measurement, C_D. We assume that $C_D > 0$. It is assumed that the cost of reinforcing each measurement is identical and that each measurement must be reinforced individually.

A.3.3 The Attacker

The set of the attacker's strategies includes masking all possible combinations of measurements in the system. Some combinations may be more feasible than others due to the number of masked measurements required to launch a stealthy attack. It is assumed that the attacker has gained access to all system control inputs and can access all of them for each strategy.

The attacker's utilities, π_A^{ij}, are calculated as a function of the attack stealth and the cost of masking measurements.

$$\pi_A^{ij} = G_A^{ij} - E_A^j \tag{A.23}$$

The attack stealth contributes to the attacker's utility through the term G_A^{ij}. This term is dependent on both the strategy of the attacker and that of the defender, and there are three cases depending upon whether the attack is observable, unobservable but unstealthy, or unobservable but stealthy:

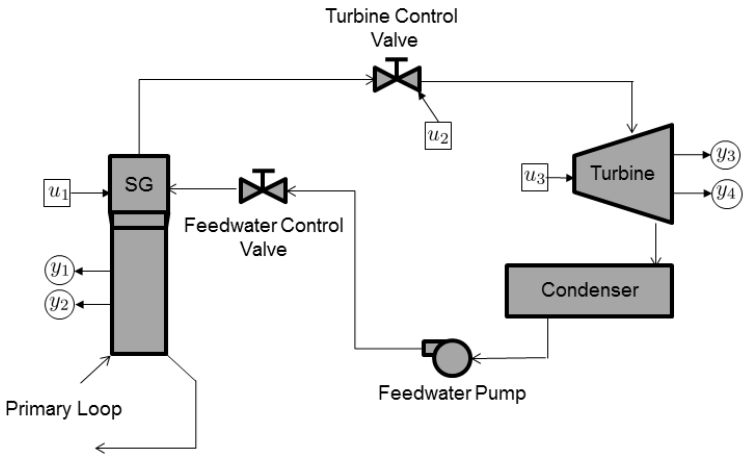
1. If the set of successfully masked measurements does not alter the results of the observability test, the attack is observable and the attacker receives a gain of G_A^O.
2. If the stealth ratio is small, the attack is unstealthy and the attacker receives a gain of G_A^U.
3. If the stealth ratio is sufficiently large, the attack is stealthy and the attacker receives a gain of G_A^S.

We assume that $0 < G_A^O < G_A^U < G_A^S$. For ease of demonstration, we have assumed that the attacker's utility is dependent on attack stealth and not on which portion of the state space has been stealthily attacked.

In this paper, we will assume that the cost of attack is proportional to the number of masked measurements. The reasoning for this assumption is similar to that for the defender's cost assumptions. The total cost of attacking the system is represented by E_A^j. The cost of attacking the system is computed as the product of the number of masked measurements, n_A^j, for attack strategy s_A^j, and the cost of masking a measurement, C_A. We assume that $C_A > 0$. It is assumed that the cost of masking each measurement is identical and that each measurement must be masked individually. It is assumed that the attacker has gained access to all control inputs for all strategies, therefore the cost of hijacking actuators is not included in the calculation of the attacker's utility.

A.4 Balance of Plant Model

We demonstrate an observability attack on the balance of plant (BOP) of a pressurized water reactor (PWR). The purpose of the BOP is to deliver the energy generated by the primary system in a usable form to a turbine-generator. In a pressurized water reactor plant, the BOP extracts thermal energy from the primary reactor loop, and converts that thermal energy to electricity. The BOP contains the following coupled components: U-tube steam generator, steam turbine, condenser, and pump. The steam generator is a heat exchanger used to extract thermal energy from the primary loop. The steam is then passed to a turbine that drives a generator to produce electricity. The output of the turbine is then condensed to a liquid state and pumped back to the steam generator. A schematic of the system is shown in figure 48. The model used in this work is identical to that used in [59, 55, 56].



System Inputs

#	Unit	Description
1	mm	Narrow-range level reference
2	kg/s	Steam flow rate
3	kg/s	Additive flow rate from turbine reheat cycle

System Outputs

#	Unit	Description
1	mm	Narrow-range level measurement
2	mm	Wide-range level measurement
3	MPa	High-pressure turbine first-stage pressure drop
4	N m	Turbine torque

Figure 48: The balance of plant system with global system inputs and measurements identified.

A.4.1 U-Tube Steam Generator

The U-tube steam generator model developed by [47] has been implemented in this work. The optimal algebraic controller designed by [1] has been used to stabilize and control the steam generator. The controller tracks a narrow-range water level reference input.

Two control inputs are associated with the steam generator system: the narrow-range reference level, u_1, and the steam flow rate exiting the steam generator, u_2. The actuator of the narrow-range water level is the feedwater control valve, and the actuator of the steam flow rate is the turbine control valve.

Two measurements are associated with the steam generator system: the narrow-range water level, y_1, and the wide-range water level, y_2. The narrow-range water level is based on the pressure differential between two points near the water level. The wide-range water level is based on the pressure differential between the top and bottom of the steam generator. While the narrow-range water level reflects the steam/water mixture level, the wide-range water level reflects the mass of water in the steam generator [49].

A.4.2 Steam Turbine

A generalized model of a steam turbine with high, medium, and low pressure sections was developed by [50]. That model has been modified in this work to include one high-pressure turbine and three low-pressure turbines to be consistent with common PWR turbine configurations. After passing through the high-pressure turbine, the steam goes through a reheat cycle and then passes through the three low-pressure turbines.

Two control inputs are associated with the turbine system: the steam flow rate entering the turbine, u_2, and the additive steam flow rate from the turbine reheat cycle, u_3. As previously mentioned, the actuator of the steam flow rate is the turbine control valve. The actuator of the additive steam flow rate from the reheat cycle is the reheat control valve.

Two measurements are associated with the turbine system: the pressure drop across the first stage of the high-pressure turbine, y_3, and the total torque produced by the turbine system, y_4. The pressure drop across the first stage of the high-pressure turbine is related to the total power produced.

A.4.3 Condenser

The condenser is implemented to change the phase of the turbine outlet from high quality steam to saturated liquid. This saturated liquid is then pumped to the steam generator as feedwater. The condenser dynamics are assumed to be sufficiently rapid to be omitted from the system model. The condenser is assumed to operate at nominal conditions. No inputs or measurements are included for the condenser system.

A.5 Results and Discussion

A game-theoretic approach has been applied to examine the observability attack scenario. Each pure strategy is defined in table 26. The observability outcome for each pure strategy profile is given in table 27. We first use iterated elimination of dominated strategies to eliminate strategies that would never be played by a rational individual. Next, we identify the pure-strategy and mixed-strategy Nash equilibria of the game, and present the conditions for their existence. Finally, we demonstrate the analysis for a numerical example.

A.5.1 Iterated Elimination of Dominated Strategies

Using iterative elimination of dominated strategies, we will reduce the dimension of the observability attack game. First, let us examine the attacker's strategies. We will begin by comparing the attack strategies that incur the same cost (i.e. have the same number of masked sensors, n_A), and then compare the remaining strategies. We take this approach because the cost of conducting the attack is the same for strategies that have the same n_A, and we can easily determine if a strategy is dominated by referring to table 27 for the attack stealth.

First let us examine strategies with $n_A = 1$. It is seen that s_A^2 is dominated by s_A^1 because s_A^1 does at least as well as s_A^2 against each of the defender's strategies. For eight of the defender's strategies, the outcome is observable when the attacker plays either s_A^1 or s_A^2. For the other eight defender strategies, the attacker is better off playing s_A^2 because s_A^2

Table 26: The strategies of the attacker and defender. The defender's strategy s_D^i includes a set of reinforced signals of quantity n_D. The attacker's strategy s_A^j includes a set of masked signals of quantity n_A.

Strategy	n	Signals					Strategy	n	Signals				
s^1	1	1					s^9	2	1			4	
s^2	1		2				s^{10}	2		2		4	
s^3	2	1	2				s^{11}	3	1	2		4	
s^4	1			3			s^{12}	2			3	4	
s^5	2	1		3			s^{13}	3	1		3	4	
s^6	2		2	3			s^{14}	3		2	3	4	
s^7	3	1	2	3			s^{15}	4	1	2	3	4	
s^8	1				4		s^{16}	0					

results in an unstealthy attack while s_A^1 results in an observable attack. By similar reasoning, s_A^4 and s_A^8 are also dominated by s_A^1. For $n_A = 2$, we can see that s_A^5, s_A^6, s_A^{10}, and s_A^{12} are dominated by s_A^3. We note that at this stage of iterative elimination, s_A^3 does not dominate s_A^9, and s_A^9 does not dominate s_A^3, therefore neither will be eliminated. For $n_A = 3$, we can see that s_A^7, s_A^{13}, and s_A^{14} are dominated by s_A^{11}. The only domination that occurs between strategies of varying n_A is that s_A^{15} is dominated by s_A^{11}. The dominated strategy has a greater cost of attack than the dominating strategy, and results in stealth gains less than or equal to those of the dominating strategy. Thus, the remaining undominated attacker strategies are s_A^1, s_A^3, s_A^9, s_A^{11}, and s_A^{16}. The columns corresponding to all dominated attacker strategies can be eliminated from the game matrix.

We will now examine the defender's strategies using the same methodology as was used for the attacker's strategies. It is important to note that the dominated attacker strategies have been removed from the game matrix, therefore we will only consider the game outcomes corresponding to the intersection of the remaining attacker strategies and

Table 27: Stealth outcomes of the observability attack game: (O) observable attack; (U) unstealthy attack; (S) stealthy attack.

	s_A^1	s_A^2	s_A^3	s_A^4	s_A^5	s_A^6	s_A^7	s_A^8	s_A^9	s_A^{10}	s_A^{11}	s_A^{12}	s_A^{13}	s_A^{14}	s_A^{15}	s_A^{16}
s_D^1	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
s_D^2	U	O	U	O	U	O	U	O	S	O	S	O	U	O	U	O
s_D^3	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
s_D^4	U	O	S	O	U	O	S	O	S	O	S	O	S	O	S	O
s_D^5	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
s_D^6	U	O	U	O	U	O	U	O	S	O	S	O	S	O	S	O
s_D^7	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
s_D^8	U	O	S	O	U	O	S	O	U	O	S	O	U	O	S	O
s_D^9	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
s_D^{10}	U	O	U	O	U	O	U	O	U	O	U	O	U	O	U	O
s_D^{11}	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
s_D^{12}	U	O	S	O	U	O	S	O	U	O	S	O	U	O	S	O
s_D^{13}	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
s_D^{14}	U	O	U	O	U	O	U	O	U	O	U	O	U	O	U	O
s_D^{15}	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O	O
s_D^{16}	U	O	S	O	U	O	S	O	S	O	S	O	U	O	S	O

Table 28: Normal form of the reduced observability attack game. The defender's utility is listed in the first row of each cell, followed by the attacker's utility.

	s_A^1	s_A^3	s_A^9	s_A^{16}
s_D^1	$-L_D^O - C_D$ $G_A^O - C_A$	$-L_D^O - C_D$ $G_A^O - 2C_A$	$-L_D^O - C_D$ $G_A^O - 2C_A$	$-L_D^O - C_D$ G_A^O
s_D^{16}	$-L_D^U$ $G_A^U - C_A$	$-L_D^S$ $G_A^S - 2C_A$	$-L_D^S$ $G_A^S - 2C_A$	$-L_D^O$ G_A^O

all defender strategies. For $n_D = 1$, we can see that s_D^2, s_D^4, and s_D^8 are dominated by s_D^1. For $n_D = 2$, we can see that s_D^6, s_D^{10}, and s_D^{12} are dominated by s_D^3. For $n_D = 3$, we can see that s_D^{14} is dominated by s_D^7. Comparing across strategies of varying n_D, we can see that s_D^1, s_D^3, s_D^5, s_D^7, s_D^9, s_D^{11}, s_D^{13}, and s_D^{15} all have the same stealth outcomes for each attack strategy. The strategy s_D^1 dominates the other strategies because it has the lowest value of n_D and therefore incurs the lowest cost to the defender. Thus, the remaining undominated defender strategies are s_D^1 and s_D^{16}. The rows corresponding to all dominated defender strategies can be eliminated from the game matrix.

Iterated elimination of dominated strategies continues until no strategy can be eliminated. The attacker's strategy s_A^{11} is now seen to be dominated by s_A^9 because s_A^{11} has stealth outcomes equal to s_A^9, but also a greater n_A. No additional defender strategies are dominated. The reduced normal form of the observability game is given in table 28. The existence of pure-strategy and mixed-strategy Nash equilibria will now be examined.

A.5.2 Pure-Strategy Nash Equilibria

There are four potential pure-strategy Nash equilibria in the observability attack game. The existence of each equilibrium is dependent on the relative magnitudes of the attacker's and defender's utility parameters. To verify that a pure-strategy profile is indeed a Nash

equilibrium, it is sufficient to check that unilateral deviations by each player to another pure strategy does not result in a greater utility for that player. Each pure-strategy Nash equilibrium is discussed below.

A.5.2.1 No reinforcement or masking The pure strategy profile (s_D^{16}, s_A^{16}) is a Nash equilibrium of the observability game if two conditions are satisfied.

1. $C_A > G_A^U - G_A^O$

   The first condition is that the cost of attacking must outweigh the benefit of achieving an unstealthy attack rather than an observable attack. This condition is obtained by examining the attacker's pure strategy deviation to s_A^1.

2. $C_A > \frac{1}{2}(G_A^S - G_A^O)$

   The second condition is that the cost of attacking must outweigh the benefit of achieving a stealthy attack rather than an observable attack. This condition is obtained by examining the attacker's pure strategy deviation to s_A^3 or s_A^9.

It is already noted that s_D^{16} is the defender's best response to s_A^{16}; therefore no additional conditions regarding the defender's utility parameters are necessary.

A.5.2.2 No reinforcement and one masking The pure strategy profile (s_D^{16}, s_A^1) is a Nash equilibrium of the observability game if three conditions are satisfied.

1. $C_D > L_D^U - L_D^O$

   The first condition is that the cost of defense must outweigh the loss of an unstealthy attack rather than an observable attack. This condition is obtained by examining the defender's pure strategy deviation to s_D^1.

2. $C_A < G_A^U - G_A^O$

   The second condition is that the benefit of achieving an unstealthy attack rather than an observable attack must outweigh the cost of attacking. This condition is obtained by examining the attacker's pure strategy deviation to s_A^{16}.

3. $C_A > G_A^S - G_A^U$

   The third condition is that the cost of attacking must outweigh the benefit of achieving a stealthy attack rather than an unstealthy attack. This condition is obtained by examining the attacker's pure strategy deviation to s_A^3 or s_A^9.

A.5.2.3 No reinforcement and two maskings The pure strategy profiles (s_D^{16}, s_A^3) and (s_D^{16}, s_A^9) are Nash equilibria of the observability game if three conditions are satisfied.

1. $C_D > L_D^S - L_D^O$

   The first condition is that the cost of defense must outweigh the loss of an stealthy attack rather than an observable attack. This condition is obtained by examining the defender's pure strategy deviation to s_D^1.

2. $C_A < G_A^S - G_A^U$

   The second condition is that the benefit of achieving an stealthy attack rather than an unstealthy attack must outweigh the cost of attacking. This condition is obtained by examining the attacker's pure strategy deviation to s_A^1.

3. $C_A < \frac{1}{2}(G_A^S - G_A^O)$

   The third condition is that the benefit of achieving a stealthy attack rather than an observable attack must outweigh the cost of attacking. This condition is obtained by examining the attacker's pure strategy deviation to s_A^{16}.

A.5.2.4 Impossible pure-strategy equilibria Some pure-strategy profiles cannot result in Nash equilibria. These results are evident from the best responses that were previously identified. The strategy profile (s_D^1, s_A^{16}) cannot be a Nash equilibrium because the

defender increases his utility from $-L_D^O - C_D$ to $-L_D^O$ by deviating to s_D^{16}. The strategy profiles (s_D^1, s_A^1), (s_D^1, s_A^3), and (s_D^1, s_A^9) cannot be Nash equilibria because the attacker increases his utility from $G_A^O - C_A$ or $G_A^O - 2C_A$ to G_A^O by deviating to s_A^{16}.

A.5.3 Mixed-Strategy Nash Equilibria

There are five support combinations that define the game's potential mixed-strategy Nash equilibria. The existence of each equilibrium is dependent on the the magnitudes of the attacker's and defender's utility parameters. All conditions are obtained by restricting the probability of playing a support strategy to be greater than or equal to zero and less than or equal to one. In all cases, the support of the defender's mixed strategy is the set of the two strategies s_D^1 and s_D^{16}. Each mixed-strategy Nash equilibrium is discussed below.

A.5.3.1 One masking and two maskings In this mixed-strategy equilibrium, the attacker's support is the set of two strategies: masking one measurement and masking two measurements. There are two potential supports for the attacker's strategy because there are identical results if the attacker plays s_A^3 or s_A^9. The mixed strategies for both players are defined below.

$$\sigma_D^1[s_D^1, s_D^{16}] = \sigma_D^2[s_D^1, s_D^{16}] = \left[1 - \frac{C_A}{G_A^S - G_A^U}, \quad \frac{C_A}{G_A^S - G_A^U}\right] \tag{A.24}$$

$$\sigma_A^1[s_A^1, s_A^3] = \sigma_A^2[s_A^1, s_A^9] = \left[1 - \frac{C_D}{L_D^S - L_D^U}, \quad \frac{C_D}{L_D^S - L_D^U}\right] \tag{A.25}$$

The existence of these two equilibria is dependent on two conditions:

1. $C_D < L_D^S - L_D^U$

   The first condition is that the loss of a stealthy attack rather than an unstealthy attack must outweigh the cost of defense.

2. $C_A < G_A^S - G_A^U$

   The second condition is that the benefit of achieving a stealthy attack rather than an unstealthy attack must outweigh the cost of attacking.

A.5.3.2 Zero maskings and one masking In this mixed-strategy equilibrium, the attacker's support is the set of two strategies: abstaining from masking and masking one measurement. The mixed strategies for both players are defined below.

$$\sigma_D^3[s_D^1, s_D^{16}] = \left[1 - \frac{C_A}{G_A^U - G_A^O}, \quad \frac{C_A}{G_A^U - G_A^O}\right] \tag{A.26}$$

$$\sigma_A^3[s_A^1, s_A^{16}] = \left[\frac{C_D}{L_D^U - L_D^O}, \quad 1 - \frac{C_D}{L_D^U - L_D^O}\right] \tag{A.27}$$

The existence of this equilibrium is dependent on two conditions:

1. $C_D < L_D^U - L_D^O$

   The first condition is that the loss of an unstealthy attack rather than an observable attack must outweigh the cost of defense.

2. $C_A < G_A^U - G_A^O$

   The second condition is that the benefit of achieving an unstealthy attack rather than an observable attack must outweigh the cost of attacking.

A.5.3.3 Zero masking and two maskings In this mixed-strategy equilibrium, the attacker's support is the set of two strategies: abstaining from masking and masking two measurements. There are two potential supports for the attacker's strategy because there are identical results if the attacker plays s_A^3 or s_A^9. The mixed strategies for both players are defined below.

$$\sigma_D^4[s_D^1, s_D^{16}] = \sigma_D^5[s_D^1, s_D^{16}] = \left[1 - \frac{2C_A}{G_A^S - G_A^O}, \quad \frac{2C_A}{G_A^S - G_A^O}\right] \tag{A.28}$$

$$\sigma_A^4[s_A^3, s_A^{16}] = \sigma_A^5[s_A^9, s_A^{16}] = \left[\frac{C_D}{L_D^S - L_D^O}, \quad 1 - \frac{C_D}{L_D^S - L_D^O}\right] \tag{A.29}$$

The existence of these equilibria is dependent on two conditions:

1. $C_D < L_D^S - L_D^O$

   The first condition is that the loss of a stealthy attack rather than an observable attack must outweigh the cost of defense.

2. $C_A < \frac{1}{2}(G_A^S - G_A^O)$

   The second condition is that the benefit of achieving a stealthy attack rather than an observable attack must outweigh the cost of attacking.

A.6 Summary and Conclusions

Cyber-physical systems are dependent on the integration of computational resources with physical processes. While modern instrumentation and control systems allow for advanced methods of monitoring and controlling systems, they also introduce new vulnerabilities. Because the cyber and physical worlds are connected, vulnerabilities in cyberspace can have consequences in the physical world.

Observers provide one technique to reconstruct signals masked by an attacker; however, if the system is unobservable, the attacker may be able to steer some states to undesirable levels while avoiding detection. To avoid detection, the attacker would design an attack input to cause damage to the unobservable states while minimizing damage to the observable states. This is called an observability attack.

A game-theoretic approach was presented to identify optimal strategies to defend against observability attacks. An attacker incurred a cost to mask a measurement in the system and received a benefit that was dependent on the stealth of the resulting attack. A defender incurred a cost to reinforce a measurement and suffered a loss that was dependent on the stealth of the resulting attack. For a nuclear balance of plant system, pure-strategy and mixed-strategy Nash equilibria were identified and the conditions for their existence were presented.

This technique can be used to analyze cyber-physical systems during the design process and to prioritize security upgrades for systems in operation. This technique is appropriate when the relative magnitudes of the attacker's and defender's utility parameters can be estimated. It is noteworthy that exact values of both player's utility parameters are not required to determine which Nash equilibria exist. By determining which Nash equilibria exist and estimating each player's utility parameters, security and control engineers can identify optimal strategies to defend against observability attacks.

Appendix B

Bayesian Game Examples

In a Bayesian game, some players have incomplete information about the other players. Within the context of critical infrastructure cybersecurity, plant defenders have incomplete information about threat agents, and threat agents have incomplete information about plant defenders. A Bayesian game provides a quantitative method for security teams to identify optimal defense strategies.

The Bayesian game-theoretic approach is demonstrated on the residual heat removal system of a boiling water reactor. Threat agents are modelled as types in the game using a threat agent library that defines each threat's characteristics. Similarly, different types of defenders are modelled by considering consequences of importance to plant stakeholders. Using these type definitions, utility functions are defined for each player. Nash equilibria of the Stackelberg game and two simultaneous games are identified and discussed. Using this procedure, a security team at a nuclear power plant can select the optimal strategy to defend the plant from cyber-threats.

B.1 Bayesian Game Theory

Bayesian games are discussed in Chapter 3. This section provides additional background about Bayesian game theory.

B.1.1 Stackelberg Games

Another factor to consider in constructing the Bayesian game is whether the game is simultaneous or Stackelberg. In a simultaneous game, both players choose their strategies

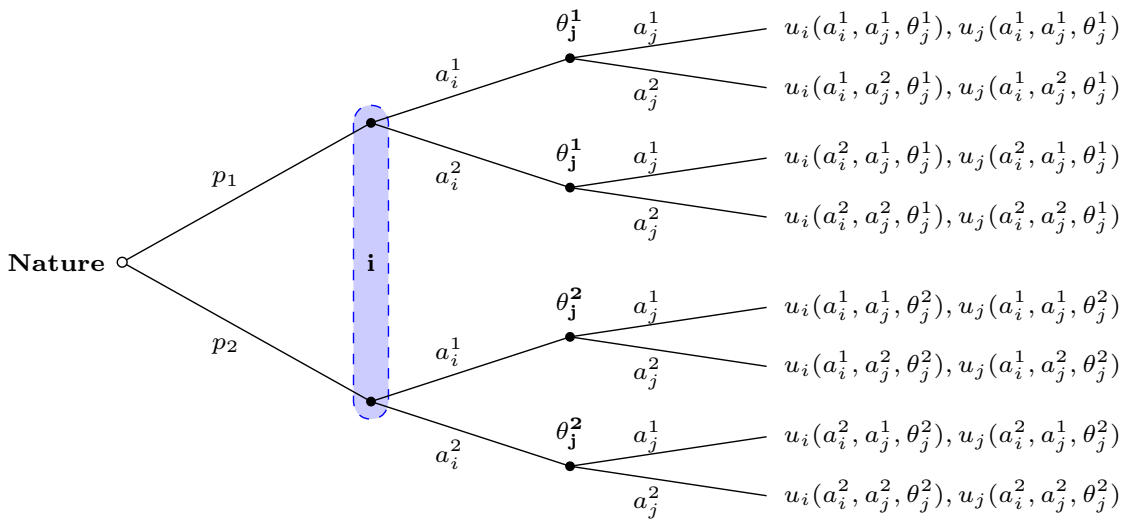


Figure 49: The extensive form of a Bayesian Stackelberg game.

at the same time. In a Stackelberg game, a leader chooses a strategy, then the follower chooses a strategy after observing the leader's strategy. A Stackelberg game is appropriate for many security applications where a defender first implements a security strategy and then the attacker implements an offensive strategy after observing the defenses.

Consider a Bayesian Stackelberg game played by i and j. The notation is the same as the example in Chapter 3, except here i is the leader and has a single type. The extensive form of this Bayesian game is shown in Figure 49. An information set is constructed for i because i does not know whether j is type θ_j^1 or θ_j^2. There are no additional information sets for j because i is a single type and j observes i's strategy before selecting a strategy.

B.1.2 Decomposed Optimal Bayesian Stackelberg Solver

Here we discuss the Decomposed Optimal Bayesian Stackelberg Solver (DOBSS) presented in [68]. DOBSS is a mixed-integer linear program that has been shown to quickly solve Bayesian Stackelberg games. This efficiency is partly because DOBSS does not require

the Harsanyi transformation to identify the Nash equilibrium. Within the context of a security game, the leader is the defender and the follower is the attacker. The defender is assumed to have a single type and the attacker is assumed to have multiple types. The assumption that the defender has a single type is dependent on the assumption that the attacker's utility is independent of the defender's type, and is often a reasonable assumption for security applications.

DOBSS is the following optimization problem,

$$
\begin{aligned}
\underset{q,z,m}{\text{maximize}} \quad & \sum_{i\in X}\sum_{\theta\in\Theta}\sum_{j\in Q} p^{\theta} R_{ij}^{\theta} z_{ij}^{\theta} \\
\text{subject to} \quad & \sum_{i\in X}\sum_{j\in Q} z_{ij}^{\theta} = 1 \\
& \sum_{j\in Q} z_{ij}^{\theta} \leq 1 \\
& q_j^{\theta} \leq \sum_{i\in X} z_{ij}^{\theta} \leq 1 \\
& \sum_{j\in Q} q_j^{\theta} = 1 \\
& 0 \leq \left(m^{\theta} - \sum_{i\in X} C_{ij}^{\theta} \left(\sum_{h\in Q} z_{ih}^{\theta} \right) \right) \leq (1 - q_j^{\theta}) M \\
& \sum_{j\in Q} z_{ij}^{l} = \sum_{j\in Q} z_{ij}^{1} \\
& z_{ij}^{\theta} \in [0,1] \\
& q_j^{\theta} \in \{0,1\} \\
& m^{\theta} \in \mathbb{R}
\end{aligned}
$$

Let x be a vector denoting the leader's strategy, and let i be a pure strategy contained in x. Then, x_i is the probability i is played in x. Let $\theta \in \Theta$ denote the follower's types, and let q^{θ} be a vector denoting θ's strategy, where j is a pure strategy contained in q. Let X and Q denote the index sets of the leader's and follower's pure strategies, respectively. Let R_{ij}^{θ} and C_{ij}^{θ} be the payoff matrices for the leader and follower, respectively, when the follower is

type θ. Let p^θ be the a-priori probability of type θ. Let M be a large positive number and let m^θ be an upper bound on θ's reward for any action. Finally, let $z_{ij}^\theta = x_i q_j^\theta$. This relationship can be used to calculate the leader's optimal pure strategy.

DOBSS returns the optimal mixed strategy for the defender and a pure-strategy best response for the attacker. A mixed-strategy best response is not required for the attacker, because any pure strategy that is part of the support of a mixed-strategy best response is also a best response. For greater detail regarding the derivation of DOBSS, readers are referred to [68].

B.2 Bayesian Game Construction

This section describes the construction of the Bayesian games. Several aspects are similar to the methods discussed in Chapter 4. The system under consideration is the RHR system described in Chapter 4. There are some differences in the construction of the game, the most notable of which is the definition of defender types.

B.2.1 The Defender's Types

The defender's types can be defined based on the defender's preferential relationships on avoiding negative losses. When given the choice between two losses, L_1 and L_2, a defender of one type may prefer to incur L_1, while a defender of another type may prefer to incur L_2. In general, if there are n consequences, this expression of preferences results in $n!$ types. The number of types is even greater when the defender's preferences are cardinal rather than ordinal. Simplifying assumptions or a brief list of consequences are required to define a manageable set of types for the defender.

We define the defender's types in terms of the losses described in Section 4.2.1. Here we also consider loss of sensitive data to be a loss (L_7). For each loss, we assume a range of values describing its magnitude. These values are assumed for demonstration purposes.

Table 29: The defender's losses and their possible values.

Consequence	Description	Lower Limit ($)	Upper Limit ($)
L_1	Loss of power generation	1×10^6	3×10^6
L_2	Environmental damage	1×10^9	5×10^{11}
L_3	Personnel injury or death	1×10^6	5×10^6
L_4	Damaged public opinion	5×10^5	1×10^6
L_5	Major equipment damage	1×10^7	5×10^7
L_6	Core damage	1×10^7	1×10^8
L_7	Loss of sensitive data	5×10^5	5×10^6

In practice, these values could be informed by business analysts, engineering teams, and regulators. The consequences and their possible magnitudes are summarized in Table 29.

The defender's types are defined by the magnitudes of these consequences. For this case study, we define five types for the defender. The defender's types and corresponding loss magnitudes are given in Table 30.

We define each type based on whether the loss magnitudes are large or small within their individual ranges. For the environmentalist type, the loss magnitudes for environmental damage and core damage are at the top of their respective ranges. Similarly, the humanitarian type assigns large values to personnel injury or death, the industrialist assigns large values to loss of power generation, damaged public opinion, major equipment damage, and core damage, and the data defender assigns large values to loss of sensitive data. The true type of the defender assigns loss magnitudes that are in the middle or upper end of their respective ranges. The defender is aware that his true type is θ_D^1, but the attacker is unaware of the defender's true type.

The definition of multiple defender types is generally only needed for Bayesian games with simultaneous decisions. With the assumption that the attacker's utility is independent of the defender's type, only one defender type is needed for Bayesian Stackelberg games.

Table 30: The defender's types and corresponding loss magnitudes.

Type	Name	$\lvert L_1 \rvert$ ($)	$\lvert L_2 \rvert$ ($)	$\lvert L_3 \rvert$ ($)	$\lvert L_4 \rvert$ ($)	$\lvert L_5 \rvert$ ($)	$\lvert L_6 \rvert$ ($)	$\lvert L_7 \rvert$ ($)
θ_D^1	True defender	2×10^6	1×10^{11}	3×10^6	7×10^5	4×10^7	1×10^8	1×10^6
θ_D^2	Environmentalist	1×10^6	5×10^{11}	1×10^6	5×10^5	1×10^7	1×10^8	5×10^5
θ_D^3	Humanitarian	1×10^6	5×10^{11}	5×10^6	5×10^5	1×10^7	1×10^7	5×10^5
θ_D^4	Industrialist	3×10^6	1×10^9	3×10^6	1×10^6	5×10^7	1×10^8	3×10^6
θ_D^5	Data defender	1×10^6	1×10^{10}	2×10^6	5×10^5	1×10^7	1×10^7	5×10^6

Table 31: The probability distributions of the types.

θ_D^i	θ_D^1	θ_D^2	θ_D^3	θ_D^4	θ_D^5
$p(\theta_D^i)$	0.30	0.30	0.25	0.10	0.05

θ_A^j	θ_A^1	θ_A^2	θ_A^3	θ_A^4
$p(\theta_A^j)$	0.35	0.25	0.30	0.10

This is because in a Stackelberg game, the attacker observes the defender's action, then acts to maximize his own utility. The assumption that the attacker's utility is independent of the defender's type is often reasonable.

B.2.2 Type Distributions

For this case study, we assume the type distributions in Table 31. We assume the defender assigns the greatest probability to the terrorist type, followed by the government cyberwarrior, disgruntled employee, and radical activist. We assume that the attacker assigns high probability to the true defender, environmentalist, and humanitarian, and low probability to the industrialist and data defender.

With the assumption that the distribution of types is common knowledge to the players, the Bayesian game can be transformed from a game of incomplete information to a game of imperfect information. This is a strong but necessary assumption to find the Bayesian Nash equilibrium. The assumption of common knowledge of the type distributions is most reasonable if a large amount of intelligence used to construct the game is open-source, or common knowledge among the players.

B.3 The Players' Actions

The actions available to the players are nearly identical to those discussed in Chapter 4. There are some minor differences, so we discuss all actions here for clarity.

The attacker and defender each have several choices to make regarding each component in the RHR system. For the defender, these choices address the configurations of the industrial control system devices. For the attacker, these choices address the attack vector for circumventing the defender's cybersecurity controls. The choices available to each player are summarized in Table 32.

We assume that the default configuration of each PLC is that authentication is off and wireless communication is enabled. The defender can choose to enable authentication on each PLC and can choose to disable wireless communication on each PLC. If the defender has enabled authentication, the attacker will not be able to connect to the PLC. If the defender has disabled wireless, the attacker will not be able to conduct the wireless exploit. The attacker requires local access for the wireless exploit, but does not require local access to connect with an unsecured PLC.

We assume that the default configuration of the switch is that the firewall is off. The defender can choose to enable the firewall. In practice, there are many possibilities for firewall configuration, but here we assume a binary decision to either enable or not enable the firewall. The attacker can choose whether to attempt an attack. If the defender has enabled the firewall, the attacker will not be able to conduct the attack.

We assume that the default configuration of the communication network is that all communication is unencrypted. The defender can choose to enable encryption. In practice, there are several encryption standards from which to choose and communication between different devices can have different encryption protocols. Here we assume a binary decision to either encrypt or not encrypt all communication. The attacker can choose whether or not to attempt to eavesdrop. If the attacker has compromised a PLC or the switch, and the network is unencrypted, then the attacker has also compromised the communication between the hacked device and the devices that are directly connected to it.

Table 32: The choices available to each player.

	PLCs	Switch	Communication Network
Defender's Choices	Authentication: on/off Wireless: on/off	Firewall: on/off	Encryption: on/off
Attacker's Choices	Connect: yes/no Wireless exploit: yes/no	Attack: yes/no	Eavesdrop: yes/no

For each of the attack and defense scenarios, we have assumed that the attack will be unsuccessful if the defender has implemented the corresponding defense. In practice, the outcome is not deterministic. The outcome is dependent on the ability of both players to execute their selected actions. For example, a complex and sophisticated custom attack may not be executed correctly, even if the attacker has the financial resources to conduct the attack. The probability that an attack is successful given that a specific defense has implemented can be estimated using expert judgment, capture-the-flag experiments, and metrics such as the Common Vulnerability Scoring System [58, 28]. The stochastic nature of the attack outcomes can be represented using chance nodes on the extensive form of the Bayesian game.

A complete action for a player consists of selecting an option for each available choice. For simplicity, we consolidate the defender's choices across the PLCs. Specifically, we give the defender the choice to enable authentication on all or none of the PLCs and the choice to disable wireless on all or none of the PLCs. We allow the attacker the option to connect or exploit individual PLCs. An example of a complete defender action is: enable authentication on all of the PLCs, disable wireless on all of the PLCs, enable the firewall on the switch, and enable encryption on the communication network. An example of a complete attacker action is: connect to PLC-1A and PLC-1B and do not connect PLC-2A and PLC-2B, conduct a wireless exploit on PLC-2A and PLC-2B and do not exploit PLC-1A and PLC-1B, do not attack the switch, and eavesdrop on the communication network.

B.3.1 Action Profiles and Consequences

In this game, the outcome of a cyber attack depends upon several factors. Two obvious factors are the actions chosen by the players. The set of actions chosen by the players during a particular play of the game is referred to as an action profile. In contrast, a strategy profile specifies the actions taken by all types of all players. In addition to the action profile, the outcome of an attack is also dependent on other NPP systems that interact with the RHR system. To cause severe consequences, not only must the cyber attack be successful, but also redundant plant systems and safety systems must fail.

First, we will consider the consequences that only depend upon the game's action profile. The loss of sensitive data consequence is assumed to be the only consequence that is dependent solely on the action profile. This is because most of the consequences are physical and are also dependent on other NPP systems. We assume the damaged public opinion consequence can only occur if another physical consequence occurs. Data loss can occur if any of the attacker's offensive actions are not defended.

Second, we will consider the consequences that depend upon the game's action profile and other NPP systems. To model the game's dependency on other NPP systems, we first consider the hazards of the RHR system (Section 4.1. Each hazard may be caused by certain action profiles. Table 12 shows which hazards can occur as a consequence of different combinations of hacked devices. We do not list each action profile because there are multiple ways some devices can be hacked.

Many of the defender's severe consequences are dependent on the failure mode of the RHR system and the failure of other NPP systems. For example, if both RHR systems operating in LPCI mode fail, the core spray system can also provide core cooling. The probability of failure of other plant systems can be calculated using a risk analysis method like fault tree analysis or Bayesian networks. The probabilities that we have assumed for this example are shown in Table 33. Note the probability that each hazard causes L_7 is zero. This is because L_7 is not a physical loss, and is not dependent on any of the hazards we have identified. Additionally, while L_4 may occur as a result of a physical failure, it may also occur as a direct result of the action profiles.

B.3.2 The Players' Utility Functions

The players' utility functions quantify the outcome of the game and enable us to identify effective security strategies. The utility functions capture the cost of the players' actions and the impact of a successful attack. The utility functions are given by Equations B.1 and B.2.

Table 33: The probability of hazards causing losses.

	L_1	L_2	L_3	L_4	L_5	L_6	L_7
H_1	—	5×10^{-6}	5×10^{-5}	—	2×10^{-2}	5×10^{-4}	—
H_2	5×10^{-2}	5×10^{-6}	5×10^{-5}	4×10^{-1}	1×10^{0}	5×10^{-4}	—
H_3	—	—	—	3×10^{-1}	—	8×10^{-3}	—
H_4	1×10^{0}	—	—	9×10^{-1}	—	—	—
H_5	—	1×10^{-5}	1×10^{-4}	—	—	1×10^{-3}	—
H_6	—	1×10^{-6}	1×10^{-5}	—	—	1×10^{-4}	—
H_7	2×10^{-2}	—	—	—	—	—	—

$$u_D(s_D, s_A, \theta_D) = \Psi_D(s_D) + \mathbb{E}[\Omega_D(s_A, s_D, \theta_D)] \tag{B.1}$$

$$u_A(s_D, s_A, \theta_A) = \Psi_A(s_A, \theta_A) + \mathbb{E}[\Omega_A(s_A, s_D, \theta_A)] \tag{B.2}$$

The utility functions of the players have two terms. The first term, $\Psi_{D/A}$, is the expense associated with the player's strategy. The second term, $\mathbb{E}[\Omega_{D/A}]$, is the expected value of the gain or loss from a successful attack.

The first terms in Equations B.1 and B.2 are the costs to the players for selecting their strategies. For the defender, this term is the expense of implementing cybersecurity measures. We assume that the expenses for the defender's strategy are the same for all of the defender's types. This is because the type definitions for the defender are based on the defender's loss magnitudes, not the defender's capabilities. This term may be positive in some applications. For example, if a cybersecurity action is expected to improve operational efficiency, the plant may expect a profit from implementing the action. For the attacker, this term is the expense of conducting a cyber attack. The attacker's expenses are dependent on the attacker's type. For both players, we assume that this term is not dependent on the opponent's strategy or type.

The expenses for the defender's and attacker's choices are summarized in Table 34. We have assumed these values based on the access, resources, and skill of the threat agents as

defined by the TAL. These values are assumed for demonstration purposes. In practice, additional threat intelligence and financial data should also be used. The government resources of the government cyberwarrior and organization resources of the radical activist and terrorist provide them with some advantages over the disgruntled employee, but the disgruntled employee's internal access to the NPP can also result in some reduced expenses. We assume that there is no expense to the attacker to abstain from a particular action. We also assume that there is no expense to the defender to leave a device in its default configuration.

The second terms in Equations B.1 and B.2 are the expected values of the reward or penalty from a successful attack. The terms are expected values because the success of an attack is often dependent on the failure of other NPP systems outside of the players' control (as described in Section B.3.1). For both players, this term is a function of the game's strategy profile, and the individual player's type. We assume that the defender only cares about whether or not a loss occurs, and that the perpetrator of the attack is irrelevant; that is, the second term is not a function of the attacker's type. We make a similar assumption for the attacker.

To find the expected values of the gain or loss from a successful attack, we need to know the magnitude of the attack outcome and the probability that the attack is successful. The magnitudes of the attack outcomes come from each player's type definition. The consequence magnitudes for the defender's types are given in Table 30 and the consequence magnitudes for the attacker's types are given in Table 10. The probability of attack success is often dependent on the failure of other redundant NPP systems. The probability of each loss given a particular hazard is shown in Table 33. We restrict the attacker to only cause one hazard. If the attacker has the capability to cause multiple hazards, we assume the attacker will choose the hazard resulting in the greatest value of $E[\Omega_A]$.

Table 34: The expense parameters for each player in the Bayesian game.

Parameter	PLCs	Switch	Communication Network
Ψ_D (\$)	Authentication on: 1×10^3 Wireless off: 2×10^2	Firewall on: 6×10^3	Encryption on: 2×10^4
Ψ_A^1 (\$)	Connect: 4×10^2 Wireless exploit: 5×10^4	Attack: 1×10^4	Eavesdrop: 5×10^2
Ψ_A^2 (\$)	Connect: 7×10^1 Wireless exploit: 6×10^3	Attack: 1×10^8	Eavesdrop: 9×10^1
Ψ_A^3 (\$)	Connect: 2×10^2 Wireless exploit: 8×10^3	Attack: 2×10^3	Eavesdrop: 3×10^2
Ψ_A^4 (\$)	Connect: 4×10^2 Wireless exploit: 5×10^4	Attack: 1×10^4	Eavesdrop: 8×10^1

Table 35: Nash equilibrium of the Bayesian Stackelberg game.

Player	Type	Strategy	Expected utility
Defender	True defender	Enable PLC authentication Disable PLC wireless Enable switch firewall Encrypt communication network	-$30,800
Attacker	Radical activist	Abstain from all attacks	$0
Attacker	Disgruntled employee	Abstain from all attacks	$0
Attacker	Gov. cyberwarrior	Abstain from all attacks	$0
Attacker	Terrorist	Abstain from all attacks	$0

B.4 Results and Discussion

In this section we discuss the results for three formulations of the Bayesian cybersecurity game: a Stackelberg game and two simultaneous games. The first simultaneous game has one defender type, and the second simultaneous game has four defender types.

B.4.1 Stackelberg Game

In this Stackelberg formulation of the Bayesian game, the defender has a single type and the attacker has four types. DOBSS was used to solve the Stackelberg game. In the Stackelberg game, first the defender chooses a strategy without knowledge of the true type of the attacker, then the attacker chooses a strategy after observing the defender's strategy.

The Nash equilibrium solution of the Stackelberg game is summarized in Table 35. In this game, the Nash equilibrium solution is for the defender to implement all security measures, and for every attacker type to abstain from attacking. At the Nash equilibrium, the defender incurs the cost of defense implementation, and the attackers do not lose or gain utility.

This Nash equilibrium is expected, given the structure of the game. In constructing the game, we have assumed that if a defender has implemented a defense for a given attack, the attack will be unsuccessful. If the defender were to choose not to implement a defense, the attacker would be able to observe that and exploit that specific vulnerability. But, if the

defender has infallibly implemented all defenses, and the attacker can observe the defender's strategy before selecting his strategy, it is expected that the attacker would abstain from attacking. It would not be rational for the attacker to incur the cost of attacking while knowing that no reward can be gained.

B.4.2 Simultaneous Game with One Defender Type

Solving Bayesian simultaneous games with large strategy spaces and large numbers of types is computationally expensive. To make the problem more tractable, we assume the attacker's strategy space consists of the set of hacked devices necessary to cause a physical failure (Table 12), and the method by which those devices are hacked. This reduces the attacker's pure strategy space from 1,024 strategies to 18 strategies. Each individual PLC can be hacked either by connecting or by wireless exploit, each pair of PLCs can be hacked by the four combinations of connecting and wireless exploits, and the switch and network can only be hacked together if the attacker attacks the switch and eavesdrops on the network. The attacker can also abstain from attacking.

This simultaneous game was solved using Gambit. A Nash equilibrium of the simultaneous game is summarized in Table 36. At this Nash equilibrium, the defender implements all defenses with near-certainty, and is expected to only incur the cost of strategy implementation. The radical activist, disgruntled employee, and terrorist types abstain from attacking and do not gain or lose utility. The government cyberwarrior type has a large probability of abstaining, but also a significant probability of attacking the switch and network, and near-zero probability of attacking PLC-2B. The government cyberwarrior's expected utility is nonzero because of the positive probability assigned to attacks on PLC-2B, the switch, and network. It is positive because there is a small chance the attacker could target a vulnerability the defender has not addressed. The expected utilities for both the defender and attacker are close to the cost of implementing the dominant pure strategy because the probability of a consequence given the Nash equilibrium strategy profile is close to zero.

Table 36: Nash equilibrium of the Bayesian simultaneous game with one defender type.

Defender

Type	Strategy	Probability	Expected utility
True defender	Disable PLC wireless, enable switch firewall and network encryption	1.323×10^{-9}	-$30,800.001
	Enable PLC authentication, switch firewall, and network encryption	5.293×10^{-8}	
	Enable PLC authentication, disable PLC wireless, enable switch firewall	4.600×10^{-5}	
	Enable PLC authentication, disable PLC wireless, enable switch firewall and network encryption	0.9999539	

Attacker

Type	Strategy	Probability	Expected utility
Radical activist	Abstain from all attacks	1	$0
Disgruntled employee	Abstain from all attacks	1	$0
Gov. cyberwarrior	Abstain from all attacks	0.9330	$0.0127
	Connect to PLC-2B	3.180×10^{-4}	
	Wireless exploit on PLC-2B	6.360×10^{-5}	
	Attack switch & eavesdrop on network	6.667×10^{-2}	
Terrorist	Abstain from all attacks	1	$0

Enumerating over all possible Nash equilibria in a Bayesian simultaneous game is computationally expensive. This is one Nash equilibrium of the game, but more equilibria may exist. For example, it is likely that a similar equilibrium exists where the government cyberwarrior connects to or wirelessly exploits PLC-1B instead of PLC-2B. This is for three reasons: (1) PLC-1B and PLC-2B serve similar roles in the RHR system, (2) compromising PLC-1B or PLC-2B can lead to the same failures, and (3) it costs the same amount to attack either PLC. It is also possible that other equilibria exist where the government cyberwarrior abstains from attacking, and one of the other attacker types assigns positive probability to an attack strategy. If multiple equilibria are found to exist, methods such as Pareto dominance, risk dominance, or focal points can be used to identify the most credible equilibrium [29, 73].

B.4.3 Simultaneous Game with Four Defender Types

In this game, we use the same action sets as in the previous simultaneous game. Instead of using one defender type, we consider four types: the true defender, environmentalist, humanitarian, and industrialist, as defined in Section B.2.1. To reduce computational expense, we eliminate the data defender type and we increase the probability of the industrialist type from 0.10 to 0.15.

This simultaneous game was solved using Gambit. A Nash equilibrium of the simultaneous game is summarized in Table 37. The results of this game are similar to that of the previous simultaneous game. At this equilibrium, the true defender, environmentalist, and industrialist types all implement every cybersecurity defense, and the humanitarian type implements every defense with near-certainty. The humanitarian type implements every defense slightly more frequently than the defender in the previous simultaneous game. The radical activist, disgruntled employee, and terrorist types abstain from attacking and do not gain or lose utility. The government cyberwarrior type has a large probability of abstaining, but also a significant probability of attacking the switch and network, and near-zero probability of attacking PLC-2B. The cyberwarrior has slightly greater probability of

abstaining from attack in this game than in the previous simultaneous game. The government cyberwarrior's expected utility is nonzero because of the positive probability assigned to attacks on PLC-2B, the switch, and network.

Similar to the simultaneous game with one defender type, it is possible that more Nash equilibria exist. For the same reasons as above, it is likely that a similar equilibrium exists where the government cyberwarrior connects or wirelessly exploits PLC-1B instead of PLC-2B. It is also possible that other equilibria exist where the government cyberwarrior abstains from attacking, and one of the other attacker types assigns positive probability to an attack strategy. Similarly, it is also possible that other equilibria exist where the humanitarian type implements all defenses and one of the other defender types assigns positive probability to more than one pure strategy. If multiple equilibria are found to exist, methods such as Pareto dominance, risk dominance, or focal points can be used to identify the most credible equilibrium [29, 73].

B.5 Summary and Conclusions

A strong cybersecurity program is essential to protect the critical assets of commercial nuclear power plants. As cyber-physical systems, nuclear power plants must have an effective cybersecurity program to ensure efficient and safe operations in the physical world. Nuclear power plants may be targeted by a variety of threat agents with varying motivations and capabilities. Nuclear power plant security teams must defend against this spectrum of threats while remaining cost-effective. To meet this need, a Bayesian game-theoretic approach was presented.

A Bayesian game-theoretic approach enables nuclear power plant security teams to identify an optimal cybersecurity strategy, given their knowledge of potential threat agents. Several tools were presented to assist security teams in the construction of a Bayesian game. Using these tools, a security team can identify the threats that pose the greatest risk to the plant, and model those threats as types in a Bayesian game. The construction and solution of several Bayesian games were demonstrated for a residual heat removal system.

Table 37: Nash equilibrium of the Bayesian simultaneous game with four defender types.

Defender

Type	Strategy	Probability	Expected utility
True defender	Enable PLC authentication, disable PLC wireless, enable switch firewall and network encryption	1	-$30,800
Environmentalist	Enable PLC authentication, disable PLC wireless, enable switch firewall and network encryption	1	-$30,800
Humanitarian	Disable PLC wireless, enable switch firewall and network encryption	5.293×10^{-9}	-$30,799
	Enable PLC authentication, switch firewall, and network encryption	2.117×10^{-7}	
	Enable PLC authentication, disable PLC wireless	3.067×10^{-9}	
	Enable PLC authentication, disable PLC wireless, enable switch firewall and network encryption	0.999999780	
Industrialist	Enable PLC authentication, disable PLC wireless, enable switch firewall and network encryption	1	-$30,800

Attacker

Type	Strategy	Probability	Expected utility
Radical activist	Abstain from all attacks	1	$0
Disgruntled employee	Abstain from all attacks	1	$0
Gov. cyberwarrior	Abstain from all attacks	0.9543	$0.0542
	Connect to PLC-2B	1.006×10^{-3}	
	Wireless exploit on PLC-2B	2.012×10^{-4}	
	Attack switch & eavesdrop on network	4.444×10^{-2}	
Terrorist	Abstain from all attacks	1	$0

The games constructed in this paper indicate that it is best for the defender to implement all cybersecurity actions to protect the residual heat removal system. The Stackelberg game indicated that the defender should implement all defenses, and the two simultaneous games indicated that the defender should implement all defenses with a probability close to one. Although these results may appear obvious in retrospect, they do not devalue the merit of the analysis. The results may not be as intuitive for games where the potential cybersecurity costs are larger relative to the magnitude of the defender's consequences if the plant system is hacked.

From the attacker's perspective, it was demonstrated that the attacker should not attack in the Stackelberg game, and the attacker should nearly always abstain from attacks in the simultaneous games. The defender benefited from the uncertainty introduced in the simultaneous game with multiple defender types. The attacker should abstain 93% of the time in the simultaneous game with one defender type, but the attacker should abstain 95% of the time in the simultaneous game with four defender types. If the attacker does choose to attack in either of the simultaneous games, it is most likely that the attacker would target the switch and communication network. Although this is considered a modification of the original game, the defender could choose to leverage those results by closely monitoring the switch and communication network.

Implementing a Bayesian cybersecurity game requires a multi-disciplinary team. Risk engineers and nuclear engineers provide insight into NPP processes and safety systems. Industrial control engineers and cybersecurity experts provide insight into the offensive and defensive actions that can be included in the game. Threat intelligence experts are valuable for the construction of the attacker's types. Using a Bayesian game-theoretic approach, cybersecurity teams can optimally allocate resources to protect the plant, given their knowledge of malicious threat actors.

Bibliography

[1] G. Ablay. A robust estimator-based optimal algebraic approach to steam generator feedwater control system. *Turkish Journal of Electrical Engineering & Computer Sciences*, 24:206–218, 2016.

[2] Blake Ryan Abrecht. Systems theoretic process analysis applied to an offshore supply vessel dynamic positioning system. Master's thesis, Massachusetts Institute of Technology, 2014.

[3] Rachid Ait Maalem Lahcen, Ram Mohapatra, and Manish Kumar. *Cybersecurity: A survey of vulnerability analysis and attack graphs*, volume 253. Springer Singapore, 2018.

[4] Stefano V. Albrecht, Jacob W. Crandall, and Subramanian Ramamoorthy. Belief and truth in hypothesised behaviours. *Artificial Intelligence*, 235:63–94, 2016.

[5] Stefano V. Albrecht and Subramanian Ramamoorthy. A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems. *12th International Conference on Autonomous Agents and Multiagent Systems 2013, AAMAS 2013*, 2(January):1155–1156, 2013.

[6] Stefano V. Albrecht and Peter Stone. Reasoning about Hypothetical Agent Behaviours and their Parameters. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, Sao Paulo, Brazil, 2019.

[7] T Alpcan and T Basar. A game theoretic analysis of intrusion detection in access control systems. *2004 43rd Ieee Conference on Decision and Control (Cdc), Vols 1-5*, pages 1568–1573, 2004.

[8] Tansu Alpcan and Tamer Başar. *Network Security: A Decision and Game-Theoretic Approach*. Cambridge University Press, 2011.

[9] Tansu Alpcan and Tamer Basar. A Game Theoretic Approach to Decision and Analysis in Network Intrusion Detection. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 2595–2600, 2003.

[10] [Authors Redacted]. Financial Challenges of Operating Nuclear Power Plants in the United States. Technical report, Congressional Research Service, Washington, D.C., 2016.

[11] B Averill and Eric A M Luiijf. Canvassing the Cyber Security Landscape: Why Energy Companies Need to Pay Attention. *Journal on Energy Security*, pages 1–7, 2010.

[12] Michael Bartock, Jeffrey Cichonski, Murugiah Souppaya, Matthew Smith, Greg Witte, and Karen Scarfone. Guide for cybersecurity event recovery. *NIST Special Publication*, pages 800–184, 2016.

[13] Richard Bellman. *Dynamic Programming*. Dover Publications, 1957.

[14] D. Boley. Technical notes and correspondence computing the Kalman decomposition: An optimal method. *IEEE Transactions on Automatic Control*, 29(1):51–53, 1984.

[15] Steven J. Brams and D. Marc Kilgour. *Game Theory and National Security*. Basil Blackwell, 1988.

[16] Eric Byres. The myths and facts behind cyber security risks for industrial control systems. *Proceedings of the VDE Kongress*, pages 1–6, 2004.

[17] Chi-Tsong Chen. *Linear System Theory and Design*. The Oxford University Press, Oxford, England, UK, 3 edition, 1998.

[18] F. Crazzolara and G. Winskel. Petri nets in cryptographic protocols. *Proceedings - 15th International Parallel and Distributed Processing Symposium, IPDPS 2001*, 00(C):1507–1515, 2001.

[19] Rene David and Hassane Alla. *Discrete, Continuous, and Hybrid Petri Nets*. Springer, 2005.

[20] Defense Science Board. Resilient Military Systems and the Advanced Cyber Threat. Technical Report January, U.S. Department of Defense, Washington, D.C., 2013.

[21] Airong Dong. Application of CAST and STPA to railroad safety in china. Master's thesis, Massachusetts Institute of Technology, 2012.

[22] Prajit K. Dutta. *Strategies and Games: Theory and Practice*. The MIT Press, 1999.

[23] G. R. Eidam. Core Damage. In L.M. Toth, editor, *The Three Mile Island Accident*, chapter 5, pages 87–106. American Chemical Society, Washington, D.C., 1986.

[24] EPRI. HAZCADS: Hazards and consequences analysis for digital systems. Technical Results 3002012755, Electric Power Research Institute, 2018.

[25] Nicolas Falliere, Liam O. Murchu, and Eric Chien. W32.Stuxnet Dossier. Technical report, Symantec Security Response, 2011.

[26] Norman Fenton and Martin Neil. *Risk Assessment and Decision Analysis with Bayesian Networks*. CRC Press, Boca Raton, FL, second edition, 2018.

[27] Jerzy Filar and Koos Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, New York, NY, 1996.

[28] FIRST. Common Vulnerability Scoring System v3.1. Technical report, FIRST, 2019.

[29] Drew Fudenberg and Jean Tirole. *Game Theory*. The MIT Press, 1991.

[30] General Electric. *Residual Heat Removal System*, chapter 10.4. General Electric, 1996.

[31] Alessandro Giua and Manuel Silva. Petri nets and Automatic Control: A historical perspective. *Annual Reviews in Control*, 45(2):223–239, 2018.

[32] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, 3 edition, 1996.

[33] John C. Harsanyi. Games with Incomplete Information Played by "Bayesian" Players: Part I, The Basic Model. *Management Science*, 14(3):159–182, 1967.

[34] John C. Harsanyi. Games with Incomplete Information Played by "Bayesian" Players: Part II, Bayesian Equilibrium Points. *Management Science*, 14(5):320–334, 1968.

[35] John C. Harsanyi. Games with Incomplete Information Played by "Bayesian" Players: Part III, The Basic Probability Distribution of the Game. *Management Science*, 14(7):486–502, 1968.

[36] Matthew H. Henry, David R. Zaret, J. Ryan Carr, J. Daniel Gordon, and Ryan M. Layer. Cyber risk in industrial control systems. In *Cyber-security of SCADA and Other Industrial Control Systems*, pages 133–166. Springer, 2016.

[37] M. Ilyas and H. Khalil. Modeling of Communication Protocols by using Petri Nets. *Computers and Industrial Engineering*, 11(1):547–551, 1986.

[38] Institute of Nuclear Power Operations. Traits of a Healthy Nuclear Safety Culture. Technical Report January, Institute of Nuclear Power Operations, 2013.

[39] Intel Information Technology. Threat Agent Library Helps Identify Information Security Risks. Technical report, Intel Corporation, 2007.

[40] Intel Information Technology. Prioritizing Information Security Risk with Threat Agent Risk Assessment. Technical report, Intel Corporation, 2009.

[41] Intel Security and Privacy Office. Understanding Cyberthreat Motivations to Improve Defense. Technical report, Intel Corporation, 2015.

[42] International Atomic Energy Agency. Safety Culture in Nuclear Installations. Technical Report IAEA-TECDOC-1329, International Atomic Energy Agency, Vienna, Austria, 2002.

[43] International Atomic Energy Agency. INES: The International Nuclear and Radiological Event Scale User's Manual. Technical report, International Atomic Energy Agency, Vienna, Austria, 2008.

[44] International Atomic Energy Agency. Computer Security at Nuclear Facilities. Technical Report 17, International Atomic Energy Agency, Vienna, Austria, 2011.

[45] International Atomic Energy Agency. Nuclear Power Plant Outage Optimization Strategy. Technical report, International Atomic Energy Agency, Vienna, Austria, 2016.

[46] International Nuclear Safety Advisory Group. Safety Culture. Technical Report 75-INSAG-4, International Atomic Energy Agency, Vienna, Austria, 1991.

[47] E. Irving, C. Miossec, and J. Tassart. Towards efficient full automatic operation of the PWR steam generator with water level adaptive control. In *Proceedings of the 2nd International Conference on Boiler Dynamics and Control in Nuclear Power Stations*, pages 309–329, 1979.

[48] Stamatis Karnouskos. Stuxnet Worm Impact on Industrial Cyber-Physical System Security. In *IECON Proceedings (Industrial Electronics Conference)*, pages 4490–4494, 2011.

[49] M. V. Kothare, B. Mettler, M. Morari, P. Bendotti, and C.M. Falinower. Level control in the steam generator of a nuclear power plant. *IEEE Transactions on Control Systems Technology*, 8(1):55–69, 2000.

[50] P. Kundur. *Power System Stability and Control*. McGraw-Hill, New York, NY, 1994.

[51] Edward A. Lee. Cyber Physical Systems: Design Challenges. In *11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369, 2008.

[52] Nancy G. Leveson and John P. Thomas. STPA Handbook, 2018.

[53] John Leyden. Feds quiz former worker over Texas power plant hack. *The Register*, 2009.

[54] Kong Wei Lye and Jeannette M. Wing. Game strategies in network security. *International Journal of Information Security*, 4(1-2):71–86, 2005.

[55] Lee T. Maccarone and Daniel G. Cole. A Game-Theoretic Approach to Defending Nuclear Instrumentation and Control Systems from Cyber-Threats. *Proceedings of the ASME 2018 International Mechanical Engineering Congress and Exposition*, 4A: Dynamics, Vibration, and Control, 2018.

[56] Lee T. Maccarone and Daniel G. Cole. A Sequential Game-Theoretic Approach to Defending Nuclear Systems from Cyber-Threats. *Proceedings of 11th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies, NPIC and HMIT 2019*, pages 387–398, 2019.

[57] Lee T. Maccarone and Daniel G. Cole. A Game-Theoretic Approach for Defending Cyber-Physical Systems from Observability Attacks. *ASCE-ASME Journal of Risk*

and Uncertainty in Engineering Systems, Part B: Mechanical Engineering, 6(2):1–9, 2020.

[58] Lee T. Maccarone and Daniel G. Cole. Advantages of a Game-Theoretic Approach for Nuclear Cybersecurity. In *ANS Annual Meeting*, 2020.

[59] L.T. Maccarone, C. J. D'Angelo, and D. G. Cole. Uncovering Cyber-Threats to Nuclear System Sensing and Observability. *Nuclear Engineering and Design*, 331:204–210, 2018.

[60] Mark Mateski, Cassandra M. Trevino, Cynthia K. Veitch, John Michalski, J. Mark Harris, Scott Maruoka, and Jason Frye. Cyber Threat Metrics. Technical report, Sandia National Laboratory, 2012.

[61] Richard D. McKelvey, Andrew M. McLennan, and Theodore L. Turocy. Gambit: Software tools for game theory, version 15.1.1. `http://www.gambit-project.org`, 2016.

[62] Jean-François Mertens, Sylvain Sorin, and Shmuel Zamir. *Repeated Games*. Cambridge University Press, 2015.

[63] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.

[64] Nuclear Energy Institute. U.S. nuclear industry safety accident rate one-year industry values. `https://www.nei.org/resources/statistics/us-nuclear-industrial-safety-accident-rate`, July 2019. Accessed: 2020-05-22.

[65] Hamed Orojloo and Mohammad Abdollahi Azgomi. A Stochastic Game Model for Evaluating the Impacts of Security Attacks Against Cyber-Physical Systems. *Journal of Network and Systems Management*, 2018.

[66] Xinming Ou and Anoop Singhal. Security risk analysis of enterprise networks using attack graphs. Technical report, NIST, Gaithersburg, MD, 2011.

[67] Praveen Paruchuri, Jonathan P. Pearce, Janusz Marecki, Milind Tambe, Fernando Ordonez, and Sarit Kraus. Efficient Algorithms to Solve Bayesian Stackelberg Games for Security Applications. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 1559–1562, 2008.

[68] Praveen Paruchuri, Jonathan P. Pearce, Janusz Marecki, Milind Tambe, Fernando Ordonez, and Sarit Kraus. Playing Games for Security: An Efficient Exact Algorithm for Solving Bayesian Stackelberg Games. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 895–902, 2008.

[69] T. E.S. Raghavan and J. A. Filar. Algorithms for stochastic games - A survey. *ZOR Zeitschrift für Operations Research Methods and Models of Operations Research*, 35(6):437–472, 1991.

[70] R.J. Reinhart. 40 years after Three Mile Island, Americans split on nuclear power. https://news.gallup.com/poll/248048/years-three-mile-island-americans-split-nuclear-power.aspx?version=print, March 2019. Accessed: 2020-05-15.

[71] Wolfgang Reisig. *Understanding Petri Nets*. Springer, 2013.

[72] Lydia Saad. Gallup Vault: Nuclear Power Plant Fears After Chernobyl. Technical report, Gallup, Washington, D.C., 2016.

[73] Thomas Schelling. *The Strategy of Conflict*. Harvard University Press, Cambridge, MA, 1960.

[74] D.J. Sieracki and M.C. Thompson. U.S. Nuclear Regulatory Commission Safety Culture Oversight. In *Proceedings of the International Conference on Human and Organizational Aspects of Assuring Nuclear Safety*, Vienna, Austria, 2016.

[75] Elion Solan. Stochastic Games. In *Encyclopedia of Complexity and Systems Science*, pages 8698–8708. Springer, 2009.

[76] Keith Stouffer, Victoria Pillitteri, Suzanne Lightman, Marshall Abrams, and Adam Hahn. Guide to Industrial Control Systems (ICS) Security. Technical report, NIST, 2015.

[77] Paul D. Stukus. Systems-theoretic accident model and processes applied to a U.S. Coast Guard buoy tender integrated control system. Master's thesis, Massachusetts Institute of Technology, 2017.

[78] Milind Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, New York, NY, 2011.

[79] André Teixeira, Kin Cheong Sou, Henrik Sandberg, and Karl H. Johansson. Secure Control Systems. *IEEE Control Systems Magazine*, 35(1):24–45, 2015.

[80] The Nuclear Regulatory Commission. NUREG/CR-2300: PRA Procedures Guide. Technical report, The U.S. Nuclear Regulatory Commission, Washington, DC, 1983.

[81] The White House, Office of the Press Secretary. Presidential policy directive – critical infrastructure security and resilience. https://obamawhitehouse.archives.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil, 2013.

[82] Garry R. Thomas. Description of the Accident. In L.M. Toth, editor, *The Three Mile Island Accident*, chapter 1, pages 2–25. American Chemical Society, Washington, D.C., 1986.

[83] United States Nuclear Regulatory Commission. BWR/4 technology manual (R-104B), 2018.

[84] U.S. Department of Energy. 21 Steps to Improve Cyber Security of SCADA Networks. Technical report, U.S. Department of Energy, 2002.

[85] U.S. Department of Homeland Security. Nuclear Reactors, Materials, and Waste Sector-Specific Plan. Technical report, U.S. Department of Homeland Security, 2015.

[86] U.S. Energy Information Administration. U.S. nuclear plant outages increased in September after remaining low during summer. https://www.eia.gov/todayinenergy/detail.php?id=37252, October 2018. Accessed: 2020-05-27.

[87] U.S. Energy Information Administration. Nuclear energy overview. https://www.eia.gov/totalenergy/data/browser/index.php?tbl=T08.01#/?f=M&start=200001, February 2020. Accessed: 2020-05-27. Query: nuclear energy data category, nuclear energy overview table.

[88] U.S. Energy Information Administration. Cyberattack halts fuel movement on colonial petroleum pipeline. https://www.eia.gov/todayinenergy/detail.php?id=47917, May 2021. Accessed: 2021-05-19.

[89] U.S. Executive Order No. 13636. Improving Infrastructure Cybersecurity, 2013.

[90] U.S. General Accounting Agency. Technology Assessment: Cybersecurity for Critical Infrastructure Protection. Technical Report May, U.S. General Accounting Agency, 2004.

[91] U.S. Nuclear Regulatory Commission. NUREG-0492: Fault Tree Handbook. Technical report, U.S. Nuclear Regulatory Commission, Washington, DC, 1981.

[92] U.S. Nuclear Regulatory Commission. NRC Information Notice 2003-14: Potential Vulnerability of Plant Computer Network to Worm Infection. Technical report, U.S. Nuclear Regulatory Commission, 2003.

[93] U.S. Nuclear Regulatory Commission. NUREG-0980, 2015.

[94] U.S. Nuclear Regulatory Commission. Information security. https://www.nrc.gov/security/info-security.html, August 2017. Accessed: 2020-05-14.

[95] U.S. Nuclear Regulatory Commission. Defense in depth. https://www.nrc.gov/reading-rm/basic-ref/glossary/defense-in-depth.html, March 2019. Accessed: 2020-05-19.

[96] U.S. Office of the Federal Register. 10 CFR §95 - Facility security clearance and safeguarding of national security information and restricted data. https://www.govinfo.gov/app/details/CFR-2001-title10-vol2/CFR-2001-title10-vol2-part95, January 2001. Accessed: 2020-05-14.

[97] U.S. Office of the Federal Register. 10 CFR §2.390 - Public inspections, exemptions, requests for withholding. https://www.govinfo.gov/app/details/CFR-2014

-title10-vol1/CFR-2014-title10-vol1-sec2-390, January 2014. Accessed: 2020-05-14.

[98] U.S. Office of the Federal Register. 10 CFR §25 - Access authorization. https://www.govinfo.gov/app/details/CFR-2014-title10-vol1/CFR-2014-title10-vol1-part25, January 2014. Accessed: 2020-05-14.

[99] U.S. Office of the Federal Register. 10 CFR §73.21 - Protection of safeguards information: Performance requirements. https://www.govinfo.gov/app/details/CFR-2019-title10-vol2/CFR-2019-title10-vol2-sec73-21/summary, January 2019. Accessed: 2020-05-14.

[100] U.S. Office of the Federal Register. 10 CFR §73.22 - Protection of safeguards information: Specific requirements. https://www.govinfo.gov/app/details/CFR-2019-title10-vol2/CFR-2019-title10-vol2-sec73-22, January 2019. Accessed: 2020-05-14.

[101] U.S. Office of the Federal Register. 10 CFR §73.23 - Protection of safeguards information-modified handling: Specific requirements. https://www.govinfo.gov/app/details/CFR-2019-title10-vol2/CFR-2019-title10-vol2-sec73-23, January 2019. Accessed: 2020-05-14.

[102] Bernhard Von Stengel. Chapter 45 computing equilibria for two-person games. In *Handbook of Game Theory with Economic Applications*, volume 3, pages 1723 – 1759. Elsevier, 2002.

[103] Jiacun Wang. Petri Nets for Dynamic Event-Driven System Modeling. *Handbook of Dynamic System Modeling*, 2007.

[104] Lingyu Wang, Tania Islam, Tao Long, Anoop Singhal, and Sushil Jajodia. An attack graph-based probabilistic security metric. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5094 LNCS:283–296, 2008.

[105] Edgar Wingender. *Biological Petri Nets*. IOS Press, 2011.

[106] World Nuclear Association. Three Mile Island accident. https://www.world-nuclear.org/information-library/safety-and-security/safety-of-plants/three-mile-island-accident.aspx, March 2020. Accessed: 2020-05-23.

[107] Alireza Zarreh, Yooneun Lee, Rafid Al Janahi, Hung Da Wan, and Can Saygin. Cyber-physical security evaluation in manufacturing systems with a Bayesian game model. *Procedia Manufacturing*, 51(2019):1158–1165, 2020.

[108] Y. Zhao, L. Huang, C. Smidts, and Q. Zhu. A game theoretic approach for responding to cyber-attacks on nuclear power plants. *11th Nuclear Plant Instrumentation,*

Control, and Human-Machine Interface Technologies, NPIC and HMIT 2019, pages 399–410, 2019.