A GAME THEORETIC FRAMEWORK TO SECURE CYBER PHYSICAL SYSTEMS (CPS) AGAINST CYBER ATTACKS

by

Khan Md Shafi Ahad Siddique B.S.

A thesis submitted to the Graduate Council of
Texas State University in partial fulfillment
of the requirements for the degree of
Master of Science
with a Major in Engineering
December 2018

Committee Members:

Clara Novoa, Chair

Mina Guirguis

Eduardo Pérez

FAIR USE AND AUTHOR'S PERMISSION STATEMENT

Fair Use

Duplication Permission

DEDICATION

Dedicated to my parents,

Md Abdul Hye Khan and Asma Khatun,

whose love and sacrifice have brought me here.

ACKNOWLEDGMENTS

I would like to express my gratitude towards Dr. Clara Novoa and Dr. Mina Guirguis for their continuous support, guidance and encouragement throughout the course of this thesis. I would also like to thank Dr. Eduardo Perez for agreeing to be on my thesis committee and for his valuable comments and support. This research is funded in-part by NSF CNS awards #1149397 and #1814064.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF ABBREVIATIONS

Abbreviation	Description
B&C	Branch and Cut
CPS	Cyber Physical System
IoT	Internet of Things
LP	Linear Programming
MIP	Mixed Integer Programming
MITM	Man In The Middle
MVA	Minimum Value Attack
OA	Offset Attack
PID	Proportional Integral Derivative
PLC	Programmable Logic Controller
PWM	Pulse Width Modulation
RNA	Random Noise Attack
SCADA	Supervisory Control And Data Acquisition

ABSTRACT

Cyber-Physical Systems (CPS) is a term describing a broad range of complex, multi-disciplinary, physically-aware next generation engineered systems that integrate embedded computing technologies (cyber part) into the physical world. CPS are engineered systems that are built from, and depend upon, the seamless integration of computational algorithms and physical components [2]. Generally speaking, they are sensor-based communication-enabled autonomous systems. Wireless sensor network for environmental control, smart grid system and industrial robotics systems can be a good example of CPS. With the exponential growth of CPS, new security challenges have emerged. Various vulnerabilities, threats, and attacks have been detected for the new generation of CPS. Additionally, the heterogeneity of CPS components and the diversity of CPS systems have made it very difficult to study the security problem with one generalized model.

This thesis focuses on the development of effective deterministic and stochastic mathematical programming approaches to protect the CPS against a wide range of cyber attacks. The primary goal of this work is to orchestrate an optimization methodology based on a game theoretic framework to protect the CPS and evaluate its results using a simulation model and a real world testbed. To assert that the game theoretic framework yields to an optimized performance, three other heuristic approaches (i.e. Greedy, Greedy-LP, Random) are formulated and their results are compared to the outcome from the game theory approach. The game theoretic model was further extended to include stochastic number of signals and stochastic effectiveness. A two-stage stochastic model was formulated and the results were compared. Further investigations included simulation of a real world system. The simulation model was coded in MatLab Simulink to emulate a real

world CPS. As a final step in this thesis, a real life CPS testbed was constructed with functioning cyber and physical components and the results from the different approaches studied are tested and compared. It has been found that the two-satge stochastic programming (two-SSP) model gives most optimized result to protect CPS.

1. INTRODUCTION

Advances in CPS enable capability, adaptability, scalability, resiliency, safety, security and usability that far exceed the simple embedded systems of today. CPS technology has transformed the way people interact with engineered systems – just as the Internet has transformed the way people interact with information. New smart CPS have driven innovation and competition in sectors such as agriculture, energy, transportation, building design and automation, health care, and manufacturing [3]. According to a report published in Forbes magazine in 2015, the total value of global cyber security market was $75 billion in 2015 and expected to reach $170 billion by 2020 [4]. Acording to another report published in 2017 in famous statistical portal *Statista* the global cyber security market is 137 billion USD in 2017 and will reach to 231 billion USD by 2022 [5].

The next generation technologies in the field of communication and IoT (Internet of Things) are expected to play an important role on CPS research. All of the CPS applications need to be designed considering the cutting-edge technologies, necessary system-level requirements, and overall impact on the real world. Fig: 1.1 shows an illustration of a CPS. Sensors collect data from physical system and send the signal to the control module which resides in a cyber domain via communication network (WAN, LAN or standard Internet protocol). The control module takes decision based on the signal it receives and sends the control signal via communication network again to the physical system actuators.The whole system operates under closed feedback loop.

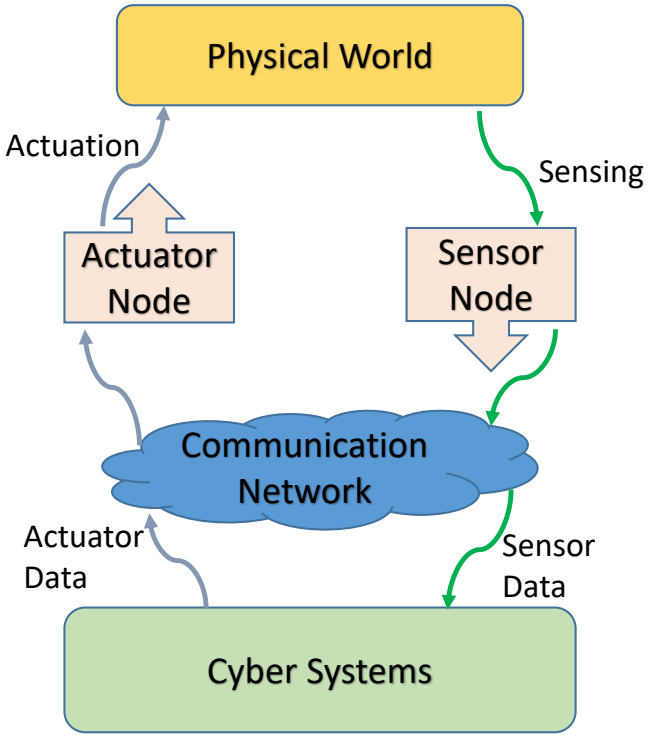

Figure 1.1: CPS holistic view.

Figure 1.2 shows a block diagram of a CPS that continuously operates under a feedback closed loop. Plant sensors send measurement signals through network to decision and control entities that process such signals and emit controls (i.e. control signals) that travel also over the network to the actuators to modify the plant operation. Adversaries can attack measurement signals or control signals to inflict damage on the plant.

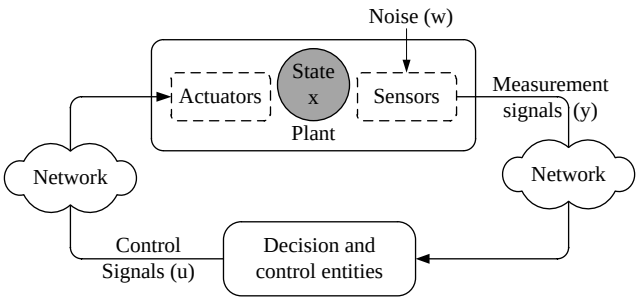

Figure 1.2: CPS block diagram

Protecting a CPS such as the one in Figure 1.2 from cyber attacks is a challenging problem as it requires orchestrating the control loop with various *blocks* such as threshold checks, model predictors, assertions and action blocks to ensure the correctness, timing and integrity of the control and measurement signals. The presence of these blocks must be within the time constraints dictated by the process to be controlled. Given a wide range of blocks available, each one with different processing time and effectiveness characteristics, it becomes challenging for the owner of the CPS (i.e. the defender) to choose the right ones, specially against a rational adversary (i.e. attacker) who is aware of the blocks that are present and seeks to inflict the maximum damage.

2. SECURITY PROBLEMS IN CPS

This is a critical time in the design and deployment of CPS and IoT. Advances in networking, computing, sensing and control systems have enabled a broad range of new devices. These systems are being designed and deployed now, however, security often is left for later. Industry is driven by functional requirements and fast-moving markets. Designs are evolving rapidly and standards are only now emerging. Many devices being deployed now have lifespans measured in decades, so current design choices will impact the next several decades in transportation, health care, building controls, emergency response, energy and other sectors.

To understand the scope of the security challenge on CPS and IoT, one has to consider the recent advances in the modern autonomous/semi-autonomous cars, the advanced medical devices, the systems that operate our buildings, the power grid and a vast number of new IoT devices. Modern cars can automatically brake to avoid a collision, modern medical devices can monitor conditions in real time and adapt to changes, buildings and the energy grid are being enhanced with a number of new smart services, and it is anticipated billions of new IoT devices will be connected to the Internet. If security is overlooked, there is a risk of unintentional faults or even malicious attacks changing how cars brake, how medical devices adapt, and how buildings and the smart grid respond to events. Cyber security only becomes more challenging if billions of devices with security vulnerabilities are added. Addressing security issues by bolting solutions onto widely deployed systems is not viable. Security issues must be analyzed, understood and addressed in the early stages of design and deployment.

One of the major attacks on CPS in recent times is the Stuxnet incident in 2010 which reportedly compromised Iranian PLCs (Programmable Logic Controller), collecting information on industrial systems and causing the fast-spinning cen-

trifuges to tear themselves apart. It was a targeted 'Man in the Middle' (MITM) attack. This type of attack fakes process control sensor signals, so an infected system does not shut down due to inability to detect abnormal behavior [6].

According to ICS-CERT [7], 198 security incidents related to industrial control systems were reported in 2011. It is a great increase if compared to the numbers presented for 2009 and 2010. Most of the security incidents took place in the areas of energy resources, water conservancy, chemical industry, governmental agencies, and nuclear facilities. Also, 52 security incidents have taken place in the energy industry during the three years and they account for 21 % of the total security incidents [7].

In 2013, American Black Hat displayed car attacks targeted at Ford Explorer, Toyota Prius, and Tesla. In 2014, Singapore Black Hat attacks aimed at cardiac pacemaker and smart television [7] and in the security protection field, seven international renowned security institutions, Fireeye, Fortinet, Lancope, Neohapsis, Symantec, Websense, and Zscaler, mentioned other kinds of attacks targeted at CPS.

3. THESIS GOAL AND OBJECTIVES

The goal of this thesis is to build valid stochastic programming models that extend the deterministic model in [8] to model CPS security problems as more realistic Stackelberg games. The goodness of the stochastic models will be assessed though a computer simulation and the construction of a realistic testbed.

The objectives of this research are:

i. Construct a two stage stochastic programming model when the number of signals arriving per target is known only through a probability distribution instead of being a single known value.

ii. Construct a two stage stochastic programming model that considers the time to run an assertion as a piecewise linear function of the assertion.

iii. Study if other parameters (i.e. effectiveness)in deterministic model can be modeled as probability distribution.

iv Assess the superiority of the game theoretic framework over other methods to assign the assertions such as a random method and a greedy assignment method.

v. Use discrete event simulation to validate the solutions obtained from the deterministic model in [8] and the proposed stochastic models in (i) and (iii).

vi. Implement the solutions obtained from the mathematical programming models in a real world CPS comprised of functioning physical and cyber components.

4. LITERATURE REVIEW

Research in the domain of CPS was not very extensive. However, over the past few years the domain of CPS has expanded dramatically with the integration of IoTs, cloud computing, smart and connected devices and SCADA systems, among others. Therefore, security and safety concerns of CPS have started to receive a lot of attention from researchers. The biggest challenge that researchers face regarding securing a CPS is the heterogeneity of such systems if considering that they are vulnerable to a wide range of cyber attacks on the network components such as jamming (A kind of Denial of Service attack, which prevents other nodes from using the channel to communicate by occupying the channel that they are communicating on [9]), spoofing (a situation in which a person or program successfully masquerades as another by falsifying data, to gain an illegitimate advantage [10]) and delaying critical information the receivers expect. The published research also differ in their assumptions about the attacker's knowledge of the system and whether the state of the system can be observed by the proposed defense mechanisms.

This Literature Review chapter focuses on previous work on CPS from three different perspectives: (1) General work on cyber security of CPS (2) Game theory and application of game theory to CPS and cyber security (3) Stochastic programming and its application to formulate and solve game theoretic problems in the field of CPS.

4.1 Previous research in the field of Cyber Security of CPS

The authors in [11] have discussed three key challenges for securing CPS: (1) understanding the threats and possible consequences of attacks, (2) identifying the unique properties of CPS and their differences from traditional IT security, and (3) discussing security mechanisms applicable to CPS. The authors have analyzed security mechanisms for prevention, detection, recovery, resilience and deterrence of attacks.

In [12], the authors show the effect of false data injection attacks (FDIA) on state estimators in power grids. In FDIA, the idea is to craft the attack vector is such a way that when combined with the state estimators it would still pass detection. The authors assume the attacker knows the configuration of the power system but may not have access to all the meters. The work in [13] generalizes FDIA on control systems using specific controllers (e.g., Kalman filters) and shows that FDIA can cause the system to become unstable. In [14], the authors illustrate a scheme in which the control signals can be spoofed, while at the same time the measurement signals are replaced by other ones to hide the effect of the attack. This scheme, in effect, hijacks the operation of the CPS. The authors performed the analysis in Simulink, using Simulink design verifier as the verification engine. A small case study is presented to illustrate the results using simple temperature control system.

In [15], the authors construct a safety envelope from the measurements obtained under normal operation of the system. Safety envelopes are highly nondeterministic and abstract models of the system that is formed by collecting data during normal operation of the system. Attack detectors are constructed to compare the measurements received during the real operation of the system to the ones maintained by the safety envelope. The authors have presented a learning-based procedure for detecting sensor attacks in a cyber-physical system.

The authors in [16] have used a new relation-graph-based detection scheme using alternation vectors with state relation graph to defeat false data injection attacks at the SCADA system. They also have evaluated the system using a real world power plant simulator. In [17], the authors prevent an adversary from finding attack vectors through identifying two sets: a set of sensors to protect and a set of state variables that can be independently verified. The authors chose a set of sensors and a set of state variables using statistical estimation and topology matrix such that, when the measurements from the sensors in the chosen set are protected and when the values of state variables from the chosen set can be verified independently, then an adversary cannot find attack vectors that can inject false data without being detected

There has been a lot of work on the impact of loss and delay of control and measurement signals on the overall stability of CPS. Research studies vary in their assumptions about the process involved in delaying and/or dropping packets as illustrated in [18, 19]. The authors in [20] study the performance of a linear control system subject to various Denial of Service (DoS) attack models on the measurements and control signals (e.g., random Bernoulli, constrained and general). The effect of wireless jamming has been shown to cause severe effects that may cripple the whole system (e.g., [21, 22]). The work in [23] describes a framework that can identify stealthy attacks on CPS in which the decision to jam a signal is based on the observed state of the system and on the cost of the attack (e.g., risk of being detected).

In cyber security, a man-in-the-middle attack (MITM) is an attack where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other. One example of man-in-the-middle attacks is active eavesdropping, in which the attacker makes independent connections with the victims and relays messages between them to make them believe they are talking directly to each other over a private connec-

tion, when in fact the entire conversation is controlled by the attacker. In [24], the authors shows ways of thwarting MITM attacks by using authentication and tamper detection.

4.2 Application of game theory in the field of CPS and cyber security

Game theory (GT) is "the study of mathematical models of conflict and cooperation between intelligent rational decision-makers" [25]. Originally, GT addressed zero-sum games, in which one person's gains result in losses for the other participants. Today, GT applies to a wide range of behavioral relations from economics to political science as well as logic, computer science and biology [25].

Game-theoretic approaches have been previously employed in the research area of network security as well as in CPS [26, 27, 28, 29, 30]. A reason is that GT allows modeling multiple ways of interaction between an attacker and a defender, assuming different defender and attacker characteristics and thus building models with diverse dynamics. One of the games used is the Stackelberg Game in which a leader moves first and then the followers move sequentially. The authors in [26] propose a Stackelberg Security Game of three player to protect an oil pipeline. The players are the system administrator acting as the leader and cyber and physical attackers performing as followers. The game has two types of targets attacked by two distinct types of adversaries with different motives. The game can coordinate the players to maximize their rewards. The solution to this game assists the system administrator of the oil pipeline CPS to allocate the cyber security controls for the cyber targets.

The authors in [27] present a model to enhance the reliability of wireless communications systems. They develop a zero-sum stochastic game to capture the interactions between a transmitter and a jammer in a communication-based train control (CBTC) system [27]. They also present analytical results and apply dynamic programming to find the equilibrium of the stochastic game.

In [28, 29, 30], the authors follow similar game theoretic frameworks to model and solve security problems for different CPS. In [31], the authors show that because of the imperfect performance of automated detection techniques, responses to such attacks are highly dependent on human-driven decision-making processes. In this work, to secure the system, the authors propose Q-Learning to react automatically to the adversarial behavior of a suspicious user. The authors presents an experimental result showing the possibility of applying Naive Q-Learning for effectively learning the opponentâs behavior and making a proper decision. Comparing the performance of different decision making algorithms, they present simulation results that show Naive Q-Learning performing better than algorithms with restricted assumptions, especially against irrational attackers.

4.3 Application of stochastic programming in formulating and solving security problems in the field of CPS

In the field of mathematical optimization, stochastic programming is a framework for modeling optimization problems that involve uncertainty. Real world problems almost invariably include some unknown parameters. Here the goal is to find a solution which is feasible for all such data and optimal in some sense. Stochastic programming models take advantage of the fact that probability distributions governing the data are known or can be estimated [32].

The most widely applied and studied stochastic programming models are two-stage (linear) programs. Here the decision maker takes some action in the first stage, after which a random event occurs affecting the outcome of the first-stage decision. A recourse decision can then be made in the second stage to compensate for any bad effects that might have been experienced as a result of the first-stage decision [33]. The optimal policy from such a model is a single first-stage decision and a collection of recourse decisions (a decision rule) defining which second-stage action should be taken in response to each random outcome [34]. The basic idea of two-stage stochastic programming is that (optimal) decisions should be based on data available at the time the decisions are made and should not depend on future observations [35].

The authors in [36] develop a stochastic network interdiction model based on a probabilistic attack graph with uncertain attack success probabilities on the arcs and formulate it as a two-stage stochastic mixed-integer linear program. They have employed the sample average approximation scheme along with the Bender's decomposition approach to solve the resulting problem. The model provides an optimal recommendation for countermeasure deployment in a stochastic environment.

In [37], the author uses two stage stochastic programming to solve non-cooperative games in the field of electric power systems and network communication systems.

The authors combine stochastic programs with Nash equilibrium to deal with certain multi-agent competition problems under uncertainty.

The work presented in this thesis is different from the works mentioned in this chapter in several ways. Firstly, this thesis takes a holistic approach to tackle the security problem in CPS rather than working with a specified area like sensor network or power grid system. This thesis tries to find an optimized framework to defend the CPS against a wide array of possible threats and attacks. This thesis combines game theory and two stage stochastic program to build a robust mathematical model and at the same time compares and validates the results using extensive simulation and implementation on a real world physically functioning testbed which to the best of our knowledge has not been done before.

5. PREVIOUS WORK

The inspiration of this thesis comes from the game theoretic framework in [8] where the problem of protecting the CPS is modeled as a Stackelberg game in which the defender commits first to an assignment of blocks (i.e. assertions) that seeks to minimize the attacker's best response. In [8], a 2-player game between a defender and an adversary is considered. The targets to protect are the different measurement and control signals in the CPS and they will be named as the target signals or the targets in the reminder of this thesis. To illustrate, if we consider an air-conditioning system, the temperature and humidity reading will be measurement signals, the signal to turn on/off the cooler or heater is the control signal. The defender seeks to protect the CPS against various attacks through enabling a set of blocks (i.e. assertions). The adversary seeks to attack the CPS through selecting a target signal to attack and a particular attack method (e.g., spoof, jam, delay). A cost is incurred with each enabled block since it would operate on incoming signals to detect attacks. Such cost is represented by the time to perform the block. Given a certain maximum time performances that the target can tolerate with the presence of all enabled blocks, the defender seeks to assign the blocks to maximize its utility subject to the adversary choosing its best response.

The mathematical programming model in [8] is presented here to help the reader follow the reminder sections of this thesis. Let n_{st} be a binary decision variable that denotes if assertion s is assigned to target signal t. The other decision variables are s_θ (defender's utility for attack type θ) and x_c (probability of an assertion of thwarting and attack). The solution to the following optimization model leads to find an optimal assignment. Table 5.1 lists all the model parameters and Table 5.2 lists all the model decision variables.

Table 5.1: Parameters notation.

Parameter	Definition
z_θ	Probabilities of encountering an adversary type θ
U_t^d	Utility of target t when defended
U_t^u	Utility of target t when undefended
E_a^s	Effectiveness of assertion s against attack a
S	Total number of assertions
N_t	Number of signals to check per target t
T_s	Time to assign assertion s
C_t	Maximum delay capacity for target t

Table 5.2: Decision variables notation.

Decision Variable	Definition
s_θ	Defenders Utility for adversary type θ
x_c	Probability of an assertion of thwarting an attack
n_{st}	Amount of assertion s assigned to target t

$$\max \quad F = \textstyle\sum_{\theta \in \Theta} z_\theta s_\theta \tag{5.1}$$

$$s_\theta \leq x_c U_t^d + (1 - x_c) U_t^u \qquad \forall \theta, c \tag{5.2}$$

$$x_c = 1 - \Pi_{s \in S}(1 - E_s^a \times n_{st}) \qquad \forall c \tag{5.3}$$

$$\sum_{s \in S} n_{st} \times N_t \times T_s \leq C_t \qquad \forall t \tag{5.4}$$

$$n_{st} \in \{0, 1\} \qquad \forall s, t \tag{5.5}$$

The objective function 5.1 maximizes the defender's worst case possible utility, s_θ, given the probability z_θ of encountering an adversary of type θ. Constraint 5.2 enforces that the defender's utility, s_θ, is the worst possible over all possible attack categories c (over all a and t), that the adversary could choose from. Equation 5.3 calculates the probability x_c of thwarting an attack of category c, given the assignment of blocks to targets n_{st}. Given that a subset of blocks are used to detect

an attack a, the attack will go undetected only if all the blocks fail on detecting the attack and hence, the product is used in 5.3 making the program non-linear. The term E_s^a is the effectiveness of assertion s against attack a. Constraint 5.4 enforces that the performance of the set of enabled assertions to protect target t would not exceed the capacity C_t for any target t. Equation 5.5 ensures a valid assignment of each block as being enabled (i.e. 1) or disabled (i.e. 0).

Two relaxations were done to linearize the above model. Constraint 5.3 was relaxed to $x_c = \max_{s \in S} E_s^a n_{st}$. The relaxation means the probability of defeating an attack is approximately computed as the probability of thwarting the attack with the most effective block. Sign constraint 5.5 was relaxed to become a continuous variable between $[0, 1]$. Equation 5.5 makes the optimization model a Mixed Integer Program (MIP) and by replacing constraint 5.5 with $0 \leq n_{st} \leq 1 \quad \forall st$, we obtain a marginal assignment of blocks to targets. This marginal assignment is implementable in two ways: (1) The check blocks are not operating at their full scales (e.g., encrypting a signal with fewer bits, averaging over a smaller number of signals, or watermarking with fewer resolution), and (2) the block is using a sampling approach to only check a subset of the signals and that percentage is given by n_{st} directly. To solve the above model, a small system with one adversary type, six targets, three assertions, three attacks and particular parameter values was artificially generated. For this sample problem, this deterministic model had 114 constraints and 127 decision variables.

6. OVERVIEW OF GAME THEORY AND MIXED INTEGER PROGRAMMING

This chapter presents an overview of the operations research methodologies applied in this research work. Firstly, it provides a synopsis of game theory. Secondly, it describes a branch of optimization named mixed integer programming and the branch and cut algorithm, the exact discrete optimization method the software uses to solve the deterministic and stochastic models researched in this work.

6.1 Overview of Game Theory and Stackelberg Game

Game theory is the process of modeling the strategic interaction between two or more players in a situation containing a set rules and outcomes [38]. Here, we will provide an introduction to game theory, its terminology,and a short description of a Stackelberg Game.

6.1.1 Definition of Game Theory

Game theory is the science of strategy, or at least the optimal decision-making of independent and competing actors in a strategic setting [39]. It is also defined as the study of mathematical models of conflict and cooperation between intelligent rational decision-makers [25]. Game theory is mainly used in economics, political science, and psychology, as well as in logic and computer science.The key pioneers of game theory were the mathematicians John von Neumann and John Nash, as well as the economist Oskar Morgenstern.

Prisoners' Dilemma is a classic example of a game in a game theoretic context. It shows why two completely rational individuals might not cooperate, even if it appears that it is in their best interests to do so. It was originally proposed

by Merrill Flood and Melvin Dresher while working at RAND (**R**esearch **AN**d **D**evelopment Corporation) in 1950. Albert W. Tucker formalized the game with prison sentence rewards and named it "prisoner's dilemma" (Poundstone, 1992). Tucker presented

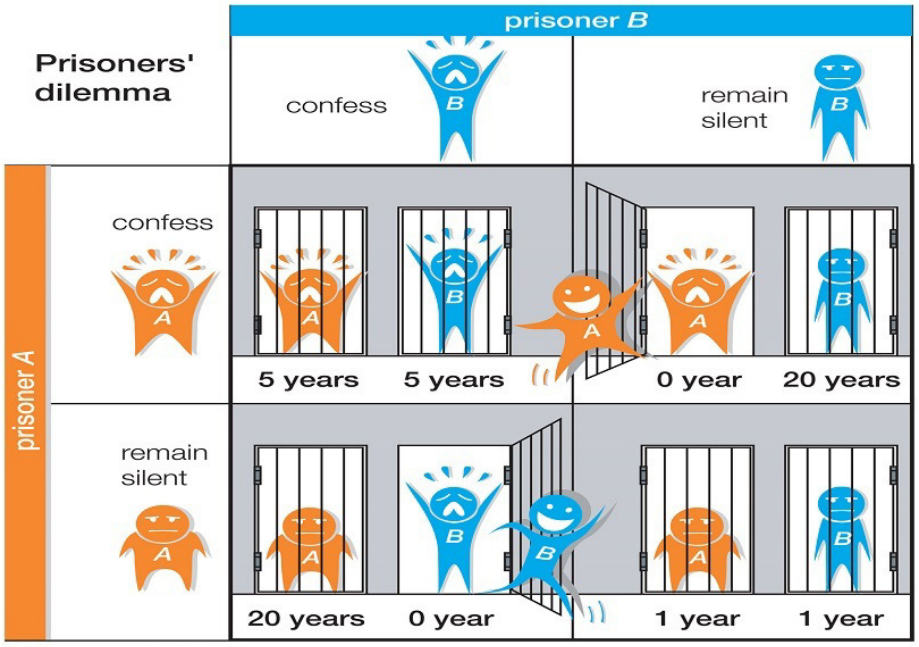

Figure 6.1: A graphic representation of prisoners' dilemma [1].

Two members of a criminal gang are arrested and imprisoned. Each prisoner is in solitary confinement with no means of communicating with the other. The prosecutors lack sufficient evidence to convict the pair on the principal charge. They hope to get both sentenced to a year in prison on a lesser charge. Simultaneously, the prosecutors offer each prisoner a bargain. Each prisoner is given the opportunity either to betray the other by testifying that the other committed the crime, or to cooperate with the other by remaining silent. The offer is:

- If A and B each betray the other (i.e. if they both confess), each of them serves 5 years in prison.
- If A betrays B but B remains silent, A will be set free and B will serve 20 years in prison (and vice versa).

- If A and B both remain silent, both of them will only serve 1 year in prison (on the lesser charge).

It is implied that the prisoners will have no opportunity to reward or punish their partner other than the prison sentences they get and that their decision will not affect their reputation in the future. Because betraying a partner offers a greater reward than cooperating with them, all purely rational self-interested prisoners will betray the other, meaning the only possible outcome for two purely rational prisoners is for them to betray each other. The interesting part of this result is that pursuing individual reward logically leads both of the prisoners to betray when they would get a better reward if they both kept silent.
The prisoner's dilemma game can be used as a model for many real world situations involving cooperative behavior. The label "prisoner's dilemma" may be applied to situations not strictly matching the formal criteria of the classic or iterative games; for instance, those in which two entities could gain important benefits from cooperating or suffer from the failure to do so, but find it difficult or expensive but not necessarily impossible to coordinate their activities.

6.1.2 Different Types of Game in Game Theory

There may be several types of games depending on: *i*. number of players, *ii*. strategies per player, *iii*. number of pure strategy Nash Equilibria, *iv*. game sequence, *v*. information availability etc. Different types of games and their examples are given below.

- Cooperative vs. Non Cooperative Games: A game is cooperative, if the players are able to form binding commitments externally enforced (e.g. through contract law). A game is non-cooperative, if players cannot form alliances or if all agreements need to be self-enforcing (e.g. through credible threats) [25]. Cooperative game theory provides a high-level approach as it only describes

the structure, strategies and payoffs of coalitions, whereas non-cooperative game theory also looks at how bargaining procedures will affect the distribution of payoffs within each coalition. As non-cooperative game theory is more general, cooperative games can be analyzed through the approach of non-cooperative game theory provided that sufficient assumptions are made to encompass all the possible strategies available to players due to the possibility of external enforcement of cooperation. However, non-cooperative games cannot be analyzed as cooperative ones.

- Symmetric vs. Asymmetric Games: A symmetric game is a game where the payoffs for playing a particular strategy depend only on the other strategies employed, not on who is playing them. If the identities of the players can be changed without changing the payoff to the strategies, then a game is symmetric. Many of the commonly studied $2X2$ games are symmetric. The standard representations of the prisoner's dilemma are symmetric games. Most commonly studied asymmetric games are games where there are not identical strategy sets for both players. For instance, the ultimatum game and similarly the dictator game [40] have different strategies for each player.

- Zero-sum vs. Non-zero-sum Games: Zero-sum games are a special case of constant-sum games, in which choices by players can neither increase nor decrease the available resources [41]. In zero-sum games the total benefit to all players in the game, for every combination of strategies, always adds to zero (more informally, a player benefits only at the equal expense of others). Poker exemplifies a zero-sum game (ignoring the possibility of the house's cut), because one wins exactly the amount one's opponents lose. Many games studied by game theorists (including prisoner's dilemma) are non-zero-sum games, because the outcome has net results greater or less than zero. Informally, in non-zero-sum games, a gain by one player does

not necessarily correspond with a loss by another. A two-person constant-sum game is a two-player game in which, for any choice of both player's strategies, the row player's reward and the column player's reward add up to a constant value c.

- Simultaneous vs. Sequential Games: Simultaneous games are games where both players move simultaneously, or if they do not move simultaneously, the later players are unaware of the earlier players' actions (making them effectively simultaneous). Sequential games (or dynamic games) are games where later players have some knowledge about earlier actions. This need not be perfect information about every action of earlier players; it might be very little knowledge. For instance, a player may know that an earlier player did not perform one particular action, while he does not know which of the other available actions the first player actually performed.

- Perfect Information vs. Imperfect Information Games: An important subset of sequential games consists of games of those under perfect information. A game is one of perfect information if all players know the moves previously made by all other players [42]. Most games studied in game theory are imperfect-information games. Examples of perfect-information games include tic-tac-toe, checkers, infinite chess etc. Many card games are games of imperfect information, such as poker and bridge.

6.1.3 Stackelberg Game

A special form of game theoretic approach has been taken for the mathematical formulation of this thesis which is called Stackelberg Game. The Stackelberg Game (also known as Stackelberg leadership model) is a strategic game in economics in which the leader moves first and then the followers move sequentially. It is a non-cooperative, sequential game. It is named after the German economist

Heinrich Freiherr von Stackelberg who published a book named Market Structure and Equilibrium (Marktform und Gleichgewicht) in 1934 which described the model [43].

In game theory terms, the players of Stackelberg game are a leader and a follower, and they compete on quantity. The Stackelberg leader is sometimes referred to as the Market Leader. There are some further constraints upon the sustaining of a Stackelberg equilibrium. The leader must know *ex ante* (before the event) that the follower observes its action. The follower must have no means of committing to a future non-Stackelberg follower action and the leader must know this. Indeed, if the 'follower' could commit to a Stackelberg leader action and the 'leader' knew this, the leader's best response would be to play a Stackelberg follower action.

6.2 Overview of Mixed Integer Programming (MIP) and Branch and Cut Method

Mixed Integer Linear Programs maximize a linear objective function subject to linear inequalities and integer sign constraints on some of the decision variables. This chapter presents how Mixed Integer Linear Programs are solved using the Branch and Cut (B&C) algorithm. A B&C algorithm relies on the advantages of a pure Branch and Bound Scheme and incorporates a new idea that makes it faster than Branch and Bound. The idea behind B&C algorithm and its effectiveness is illustrated after providing brief descriptions of the terminology used.

- **Mixed Integer Programming (MIP):** A mixed-integer programming (MIP) problem is one where some of the decision variables are constrained to be integer values (i.e. whole numbers such as -1, 0, 1, 2, etc.) at the optimal solution. The use of integer variables greatly expands the scope of useful optimization problems.

  Even with highly sophisticated algorithms and modern supercomputers, there are models with just a few hundred integer variables that have never been solved to optimality. This is because many combinations of specific integer values for the variables must be tested, and each combination requires the solution of a regular linear or nonlinear optimization problem. The number of combinations rises exponentially with the size of the problem.

  A mixed-integer program with n variables and m constraints has the form:

$$Minimize: \quad C^T x \tag{6.1}$$

$$Subject \quad to: \quad Ax \leq b \tag{6.2}$$

$$x \geq 0 \tag{6.3}$$

$$x \in Z^n \tag{6.4}$$

where A is a $m*n$ matrix. If all the variables are rational, this is a linear programming (LP) problem, which can be solved in polynomial time. In practice linear programs can be solved efficiently for reasonable-sized problems, or even for big problems with special structure. However when some or all of the variables must be integer, corresponding to mixed integer programming or pure integer programs, respectively, the problem becomes NP-complete or formally intractable since not all instances solve in polynomial time.

- **Branch and Cut Method (B&C):** Branch and cut [44] is a method of combinatorial optimization for solving integer linear programs (ILPs), that is, linear programming (LP) problems where some or all the unknowns are restricted to integer values. Branch and cut involves running a branch and bound algorithm and using cutting planes to tighten the linear programming relaxations [45].

- **Idea behind a B&C Algorithm:** The development of the B&C algorithm begins with an understanding of what new constraints may be suitable and helpful to more accurately describe the integer or mixed integer problem feasible region. The following definition is very important to understand the B&C method.

Definition: *A linear inequality is a valid inequality for given discrete optimization model if it holds for all (integer) feasible solutions to the model* [46].

Relaxations can often be strengthened dramatically by including valid inequalities that are not needed for a correct discrete model. Not every valid inequality strengthens a relaxation. For example, all inequality constraints of the original formulation are trivially valid because they are satisfied by every feasible solution. To strengthen a relaxation, a valid inequality must cut off (render infeasible) some feasible solutions to the current LP relaxation that are not feasible in the full ILP model. This need to cut off non-integer relaxation solutions is why valid inequalities are sometimes called **cutting planes**.

- B&C Algorithm

  The steps for the Branch and Cut algorithm for 0-1 Integer Linear Programs are described below and they follow the presentation in [46]:

  Step 0: Initialization. Make the only active partial solution the one with all discrete variables free, and initialize solution index $t \leftarrow 0$. If any feasible solutions are known for the model, also choose the best as incumbent solution $\hat{x}$ with objective value $\hat{v}$. Otherwise, set $\hat{v} \leftarrow -\infty$ if the model maximizes and $\hat{v} \leftarrow +\infty$ if it minimizes.

  Step 1: Stopping. If active partial solutions remain, select one as $x^{(t)}$, and proceed to step 2. Otherwise, stop. If there is an incumbent solution $\hat{x}$, it is optimal, and if not, the model is infeasible.

Step 2: Relaxation. Attempt to solve the linear programming (LP) relaxation of the candidate problem corresponding to $x^{(t)}$.

Step 3: Termination by Infeasibility. If the LP relaxation proved infeasible, there are no feasible completions of partial solution $x^{(t)}$. Terminate $x^{(t)}$, increment $t \leftarrow t + 1$, and return to Step 1.

Step 4: Termination by Bound. If the model maximizes and LP relaxation optimal value $\tilde{v}$ satisfies $\hat{v} \leq \tilde{v}$, or it minimizes and $\hat{v} \geq \tilde{v}$, the best feasible completion of partial solution $x^{(t)}$ cannot improve on the incumbent. Terminate $x^{(t)}$, increment $t \leftarrow t + 1$, and return to Step 1.

Step 5: Termination by Solving. If the LP relaxation optimum $x^{(t)}$ satisfies all binary constraints of the model, it provides the best feasible completion of partial solution $x^{(t)}$. After saving it as new incumbent solution by $\hat{x} \leftarrow \tilde{x^{(t)}}$ and $\hat{v} \leftarrow \tilde{v}$, terminate $x^{(t)}$, incumbent $t \leftarrow t + 1$, and return to Step 1.

Step 6: Valid Inequality. Attempt to identify a valid inequality for the full ILP model that is violated by the current relaxation optimum $\tilde{x^{(t)}}$. If successful, make the constraint a part of the full model, increment $t \leftarrow t + 1$, and return to Step 2.

Step 7: Branching. Choose some free binary-restricted component X_p that was fractional in the last LP relaxation optimum, and branch $x^{(t)}$ by creating two new actives. One is identical to $x^{(t)}$ except that X_p is fixed $= 0$, and the other is the same except X_p is fixed $= 1$. Then increment $t \leftarrow t + 1$ and return to Step 1.

Before turning the current partial solution into two, the branch and cut algorithm would try to improve the relaxation. The idea is to find an inequality satisfied by every binary solution to the full model but violated by $\tilde{x^{(t)}}$. Methods used to find such cutting inequalities vary enormously from one model to another. They can be cuts derived from the observation of problem specific characteristics or cuts from specialized models that also apply to the problem to solve. Besides the cuts can come from families of known valid inequalities such as the Gomory Cuts.

7. RESEARCH METHODOLOGY

This thesis follows a comprehensive methodology to model and solve the cyber security problem and analyze, simulate and validate the model's solution. A three step process that includes: (1) models' design, (2) experimentation and analysis and (3) validation is proposed and described in this chapter.

7.1 Step 1: Design of Mathematical Programming Models

7.1.1 Two-Stage Stochastic Programming (Two-SSP) Model with stochastic number of signals

In the deterministic mathematical programming model presented in Chapter IV (equations 5.1 to 5.5), the number of signals arriving per target in the time horizon T for which the CPS will be protected is assumed to be known. However, in a real situation the number of arriving signals might be known only through a certain probability distribution due to the fact that sometimes multiple signals are sent because of loss of data packets during transmission. Therefore, to solve the problem of protecting a CPS in a more realistic way, a two-stage stochastic programming approach is investigated.

In this Two-SSP, the recourse action a defender take is to modify the initial assignment of assertions considering each scenario (i.e. outcome) for the probability distribution of the number of signals. For instance, if the number of arriving signals ends high, given that there is a maximum delay capacity the CPS needs to abide, the defender would like to modify the original assignments to properly respond to the surge of signals. On the other hand, if the number of arriving signals is low, the defender would like to use the extra capacity available to allocate a higher amount of assertions to those targets where such increase improves the problem's objective function.

Recourse actions are then the fractional amounts of assignment that can be added to or subtracted from the initial assignments for each target in each scenario. A new decision variable Z_{ste} is introduced to represent the recourse action to take for assertion s and target t under scenario e. Note that this decision variable is capitalized to differentiate it from the parameter z_θ, the probability of encountering an attack of type θ. The value of Z_{ste} is restricted to be between $\pm k$,where k is a constant fixed by the model's user. Table 7.1 lists all the parameters and Table 7.2 lists all the decision variables used in this two-stage stochastic model.

Table 7.1: Parameters notation.

Parameter	Definition
U_t^d	Utility of a target t when defended d
U_t^u	Utility of a target t when undefended u
E_a^s	Effectiveness of assertion s against an attack a
P_e	Probability of receiving the number of signals estimated for scenario e
N_{te}	Estimated number of signals arriving to target t under scenario e
T_s	Time to assign and run the initial amount of assertion s assigned
T_a	Time to perform the recourse action
C_t	Delay capacity (i.e. total time available to run all assertions on target t)
k	Maximum value allowed for the recourse action Z_{ste} (i.e. limit on the amount of change on initial assignment implemented by the recourse action)

Table 7.2: Decision variables notation.

Decision Variable	Definition
s_θ	Defenders utility for adversary type θ
x_c	Probability an assertion thwarts an attack category c where c is defined by specifying target and attack (i.e. c is the tuple <t,a>)
n_{st}	Fraction of assertion s assigned to target t
Z_{ste}	Second stage decision variable that represents the recourse action as described in previous paragragh

The mathematical program for the proposed Two-SSP model is shown below. This model corresponds to the extensive form of the stochastic program also known as the deterministic equivalent model. For more details on the stochastic programming models the reader can consult [46].

$$\max \qquad F = \sum_{\theta \in \Theta} \mathbb{E}(z_\theta S_\theta) \tag{7.1}$$

$$\mathbb{E}(s_\theta) \leq \mathbb{E}(x_c) U_t^d + [1 - \mathbb{E}(x_c)] U_t^u \qquad \forall \theta, c \tag{7.2}$$

$$\mathbb{E}(x_c) = \max_{s \in S} \{ n_{st} E_s^a + \sum_e E_s^a p_e Z_{ste} \} \qquad \forall c \tag{7.3}$$

$$\sum_{s \in S} n_{st} N_{te} T_s + \sum_{s \in S} |Z_{ste}| N_{te} T_a \leq C_t \qquad \forall t, e \tag{7.4}$$

$$n_{st} + Z_{ste} \leq 1 \qquad \forall s, t, e \tag{7.5}$$

$$n_{st} + Z_{ste} \geq 0 \qquad \forall s, t, e \tag{7.6}$$

$$0 \leq n_{st} \leq 1 \qquad \forall s, t \tag{7.7}$$

$$-k \leq Z_{ste} \leq k \qquad \forall s, t, e \tag{7.8}$$

Expression 7.1 is the objective function that maximizes the expected worstcase utility for the defenders. Constraint 7.2 calculates the expected worstcase utility over all the attack categories based on the given utility of each target and the expected probability of thwarting an attack. Constraint 7.2 enforces that the expected utility computed is in fact the worstcase one the defender can experience. Equation 7.3 determines the expected probability of thwarting an attack for each target and each attack under each scenario. Constraint 7.4 is the main constraint that controls the amount of first-stage, n_{st}, and second-stage, Z_{ste} assignment of assertion s that can be assigned, given the number of signals, and the times to assign the first and second stage amounts of assertions. The absolute value for Z_{ste} in this constraint ensures that the addition or subtraction of a certain amount of

the original assertion always implies the use of some of the available capacity C_t. Constraints 7.5 and 7.6 ensure that the sum of initial and additional assignments does not exceed 1 or is less than 0 under any probability scenario. Constraint 7.5 permits that z_{ste} be lower or higher than n_{st} meaning that additions of assignment over the original fraction are valid as long as they do not exceed 1. The set of constraints 7.5 and 7.6 are critical since 7.5 permits increases on the amount of assignment that may exceed the first stage assignment as long as they do not exceed 1. Constraint 7.6 assures that decreases on the amount of assignment exceeding the first stage assignment are not permitted. Constraints 7.7 and 7.8 restrict the upper and lower limits for the initial and additional assignment of assertions.

7.1.2 Two-Stage Stochastic Programming (Two-SSP) Model with stochastic number of signals and stochastic effectiveness

The model presented in section 7.1.1 is also adapted to solve the case in which two parameters in the model are random. Here it is assumed that not only the number of signals arriving under each target is known through a probability distribution, but also the effectiveness of a block against an attack, E_b^a. The reason behind this consideration is to make the model more agile and fit more realistic scenarios. Because in a real setting, the effectiveness of a block assignment may vary with the sampling frequency, complexity of encryption etc. The new number of scenarios e' in the model continues to be finite but it is larger. Since the effectiveness of the blocks and the number of arriving signals are two independent random events, the new number of scenarios e' is found through the probability tree of all possible outcome values for these two parameters. The probabilities associated with each scenario are the product of the branches in the probability tree. The new Two-SSP model has same notation as the one in subsection 7.1.1 except for equation 7.2 which changes as shown in equation 7.9. In this new equation, $E_{ae'}^s$ represents the effectiveness of block s to attack a under realization or scenario e'.

$$\mathbb{E}(x_c) = \max_{s \in S} \{E_{se'}^a n_{st} + \sum_{e'} E_{se'}^a p_{e'} Z_{ste'}\} \qquad \forall c \tag{7.9}$$

7.1.3 Piecewise Linear Model

In the model in subsection 7.1.1 the time to assign an assertion, T_s, and the time to perform the recourse action on the assertion, T_a were considered constant. It means that irrespective of a full assignment (i.e. $n_{st} = 1; Z_{ste} = \pm k$) or a partial assignment (i.e. $n_{st} < 1; Z_{ste} = \pm k$) of an assertion s to a target t, the unit time required is the same. This assumption fails if there is an overhead to assign the blocks or if the block's setting time depends on the amount of marginal blocks assigned, like in the assignment of machine learning blocks. In this section, a new model is presented. Here, T_s is assumed to be defined by a piecewise linear function of n_{st} and T_a is assumed to be defined by a piecewise linear function of Z_{ste}.

In a general form, the piecewise linear function used for T_s has n slopes equal to $T_1, T_2, T_3, ..., T_n$ at breakpoints $x_1, x_2, x_3, ..., x_n$ that are fractional values of n_{st}. Similarly, the piecewise linear function for T_a has slopes equal to $T'_1, T'_2, T'_3, ..., T'_n$ at breakpoints $x_1, x_2, x_3, ..., x_n$ that are fractional values of Z_{ste}. Here it is considered that the time required to assign second stage block assignment, T_a is greater than time required to assign first stage block assignment, T_s

The objective function and the constraints for the piecewise linear model are the same as in the stochastic model described by the equations 7.1 to 7.8. The only difference is that, the time T_s and T_a are piecewise functions of n_{st} and Z_{ste} expressed as following.

$$T_s = \begin{cases} T_1 & \text{if} \quad 0 < n_{st} \leq x_1 \\ T_2 & \text{if} \quad x_1 < n_{st} \leq x_2 \\ \ldots & \\ T_n & \text{if} \quad x_{n-1} < n_{st} \leq x_n \end{cases}$$

$$T_a = \begin{cases} T'_1 & \text{if} \quad 0 < Z_{ste} \leq x_1 \\ T'_2 & \text{if} \quad x_1 < Z_{ste} \leq x_2 \\ \ldots & \\ T'_n & \text{if} \quad x_{n-1} < Z_{ste} \leq x_n \end{cases}$$

7.2 Step 2. Experimentation and Analysis

The three models presented in subsection 7.1.1, subsection 7.1.2 and subsection 7.1.3 were solved with the same small CPS used to solve the deterministic model presented in chapter 5. The numerical results and analysis will be presented in the next chapter. Such small CPS had one adversary type, six targets, three assertions, three attacks, and the remaining particular parameter values were artificially generated. For this small sample problem, the model had 114 constraints and 127 decision variables. The small CPS allowed us to easily construct the model and compare the performance of the different models.

However, in reality a much larger problem may arise and the success of the proposed stochastic approach will lie in the ability of solve the model within an acceptable time limit. As it is expected that solving time grows exponentially with the increase in size of the problem, the model presented in the subsection 7.1.1 was progressively expanded to include more targets, assertions or check blocks and attacks and to find if these expansions would permit to observe the hypothesized effect of problem size on solving time . The first step of the expansion was to use 30 targets, 10 check blocks, and 6 attacks. The second step was to increase the number of targets to 50 and the number of check blocks and attacks to 10 and 6 respectively. In the final step, an even larger problem was constructed using 100 targets, 20 check blocks, and 20 attacks.

7.3 Step 3a: Validation of Mathematical Models through the Development of a Discrete Event Simulation Model

In the third step of this research, a simulation model in Matlab Simulink is developed to simulate the block assignments obtained from solving the deterministic model and the one with stochastic number of signals. The simulation approach gives an opportunity to compare the results obtained from mathematical model and also assess the quality of the solution. Matlab simulink is a very strong simulation tool that has a wide range of built-in functions as well as the flexibility of using user defined functions. The Matlab Simulink model simulates the solution from the two stage stochastic model on a small CPS subject to different attack categories. The Simulink model provides the opportunity to validate mathematical model and compare the expected worstcase utility of the defender.

Figure 7.1 represents a simplified diagram of the developed Simulink model. It consist of four target signals (two measurement and two control signals). The Simulink model simulates a simple adaptive cruise control system used in an autonomous driving component to maintain a specified cruise speed as well as a safe distance from the vehicle in front of it. The system has a signal builder block that generates an input signal to the lead vehicle and causes it to accelerate or break. This input signal regulates the velocity of the lead vehicle. The host vehicle has to follow the lead vehicle with a preset cruise speed and also maintain a preset minimum distance. The host vehicle distance from the lead vehicle is the second measurement signal in this CPS. If the distance between the vehicles is greater than a set distance, the host vehicle will accelerate until it reaches cruise speed. Then the host vehicle will maintain the cruise speed unless the distance from the lead vehicle becomes less than the set distance. If the host vehicle reaches the set distance, it will reduce its speed. A control signal for the host vehicle is the

acceleration and it allows to increase or decrease host vehicle speed. Such signal is controlled by a PID controller block and a feedback loop.

The measurement signals and control signals are listed in Table-7.3

Table 7.3: Signal definition of simulink model.

Sl.	Signal Name	Signal Definition
1	Measurement Signal-1	Speed of host vehicle
2	Measurement Signal-2	Distance of host vehicle from lead vehicle
3	Control Signal-1	Signal to host to accelerate
4	Control Signal-2	Signal to host to brake

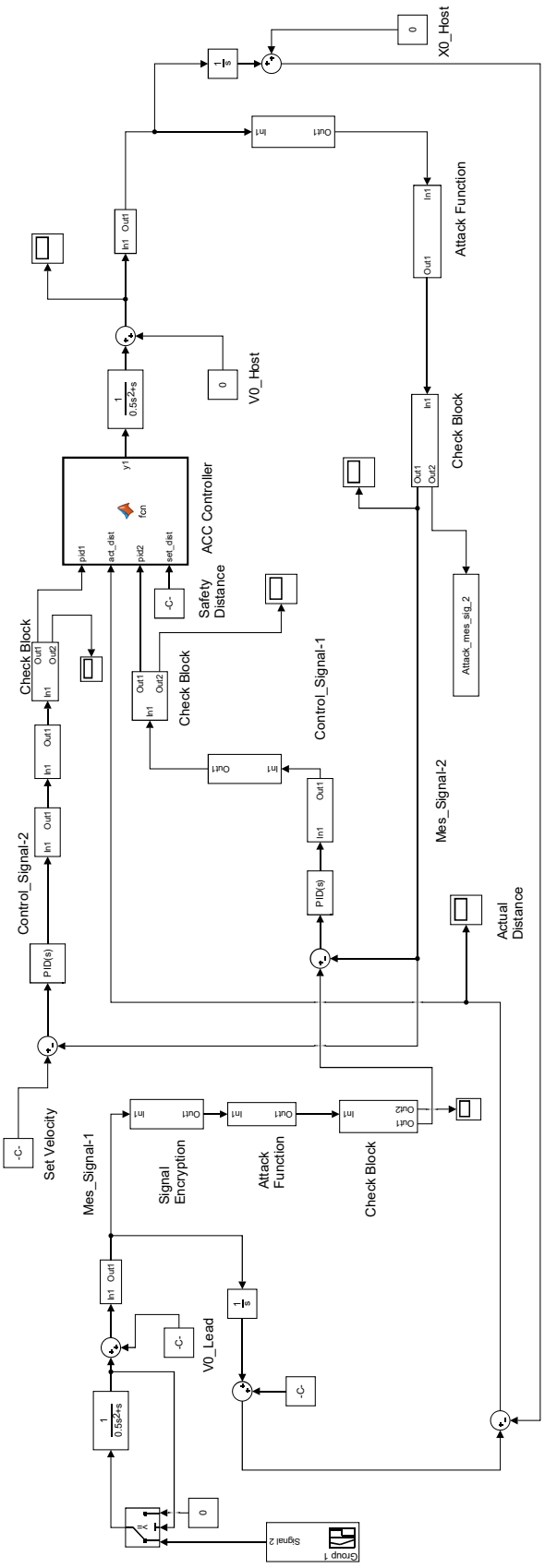

Figure 7.1: Simulation of an adaptive cruise control system.

This proof-of-concept CPS model has four target signals (two measurement signals $[Mes\_Signal{-}1, Mes\_Signal{-}2]$ and two control signals $[Control\_Signal{-}1, Control\_Signal-2]$). The stochastic number of signals is generated through conditional rate transition blocks that follow a probability distribution for the number of signals per target and scenario N_{te} defined as $N_{te} = [5, 7, 10]$ with probabilities $p_e = [0.2, 0.3, 0.5]$. The conditional rate transition block is a built in simulink function block that is able to generate signals at different rates which can be set by user.

The following three types of cyber-attack functions were programmed in the Simulink model to manipulate the measurement and control signals and to prevent the CPS from operating optimally. The attack functions are:

- **M**inimum Value Attack (MVA):

$$\bar{y} = \begin{cases} y & \text{if} \qquad y > k \\ k & \text{if} \qquad y \leq k \end{cases} \tag{7.10}$$

  MVA fixes the input signal y to an arbitrary minimum k. If the value of incoming signal y is less than k, it denies the signal and sends the minimum value which creates a temporary instability in the CPS because for a certain time the system will receive the same value.

- **O**ffset Attack (OA):

$$\bar{y} = y + rand[-y, y] \tag{7.11}$$

OA adds an arbitrary random number in the range $[-y, y]$, where y is the value of the incoming signal. Then, this attack abruptly changes a signal and tends to change the behavior of the CPS.

- **R**andom Noise Attack (RNA):

$$\bar{y} = y + N(\mu, \sigma^2) \tag{7.12}$$

RNA adds a noise to the original signal, it is assumed that the signal follows a normal distribution with mean μ and variance σ^2. Then this attack can tweak the signal to deceive the check blocks and destabilize the CPS.

The Simulink model has three check blocks that are deployed to detect the attacks and consequently protect the CPS.

- **P**arity-based Check Block: This check block marks even signal and odd signal differently and can detect the attack by reversing the marking procedure. This check block has a high chance of catching a MVA and a fair chance to protect the CPS against RNA and OA.

- **T**hreshold-based Check Block: This check block compares the incoming signal with a preset value and decides that an attack has happened if the value exceeds the threshold value. This check block is moderately effective against MVA or OA where the attack function largely manipulates the signal. However, this check block is not very effective against RNA.

- **W**atermarking Block: This check block adds a checksum digit in the least significant digit of the signal. It has a very negligible effect on the signal property and behavior but can be checked for attack detection. Watermarking is very effective against RNA because this attack adds a noise to the

signal which changes the checksum digit in the signal. However, this check block can be less effective against MVA or OA, as random attack can pick values in a wider range meaning that MVA or OA can manipulate the signal without changing the checksum digit of the signal thus preventing the check block form detecting an attack.

The different parts of the CPS in Simulink are interconnected through a network that can be subject to different cyber-attacks. The attack nodes have different attack functions and will be activated when the attacker decides to attack. Check blocks are employed before the signals enter the control module thus they can detect whether a signal has been attacked or not.

For easier understanding of the readers a simplified version of the simulink model is presented in figure-7.2

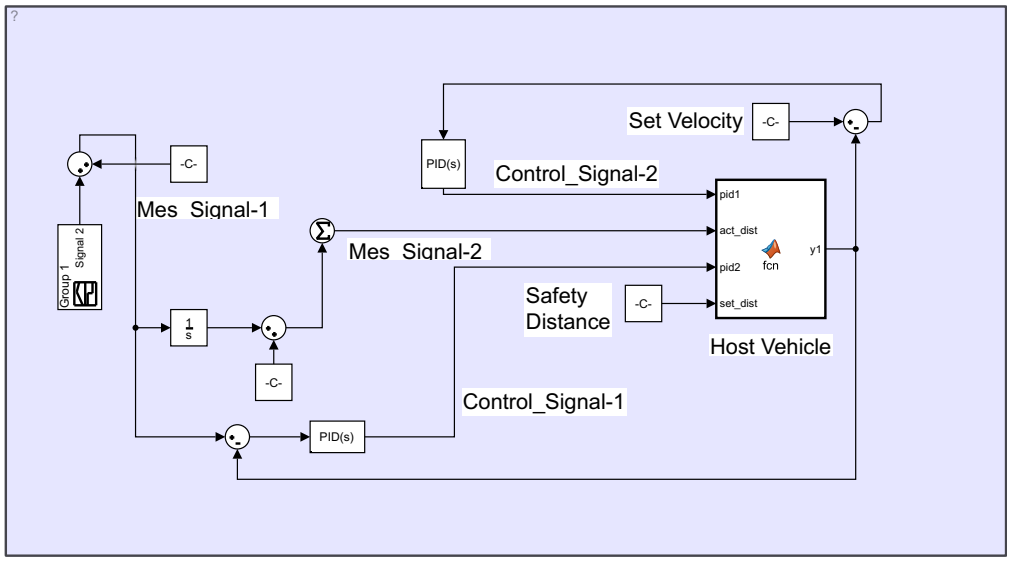

Figure 7.2: A simplified version of simulink model presented in figure-7.1.

7.4 Step 3b: Validation of Models using a Real-World Testbed

To assert the outcome of this research and validate the applicability of the results found via mathematical programming and simulation models, the author of this thesis constructed a testbed to experiment on a realistic CPS. Such testbed is a simplified version of a real world CPS that has an appropriate size to be managed in a laboratory . It has all of the functional physical and cyber components of a real world CPS. This CPS testbed allows to test and simulate cyber attacks and observe their effect without compromising a real system.

The physical system of the testbed mainly consists of a small scale steam turbine coupled with a boiler (steam generator). The turbine is subsequently coupled with a DC generator. The steam from the boiler rotates the turbine and the turbine rotates the generator shaft producing a DC voltage. The cyber component of the testbed is emulated by interconnecting arduino boards (Uno and Mega 2560) and a PC using the Matlab-Arduino interface. Some sensors and actuators bridge the communication between the cyber system and physical system. Figure-7.3 shows a simplified diagram of the CPS testbed. Steam temperature, steam pressure and voltage are the measurement signals and PWM of heater on/off is the control signal.

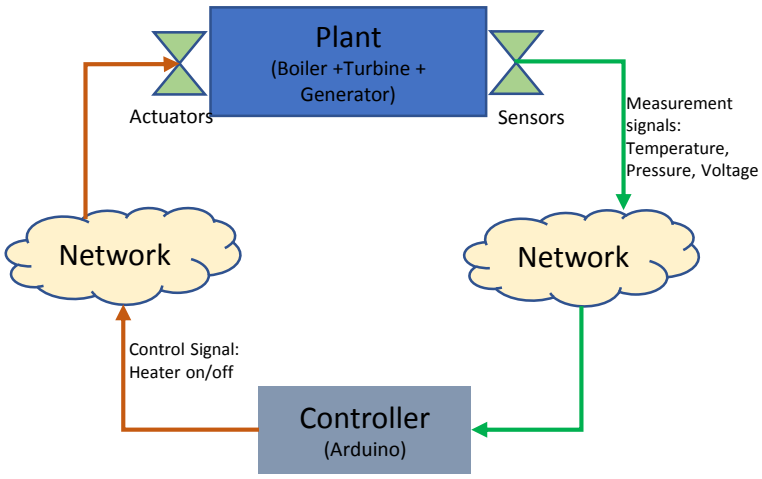

Figure 7.3: A simplified diagram of CPS testbed.

Static and dynamic data of different signals (temperature, pressure, voltage etc.) are collected under different attack scenarios used in the mathematical programming and simulation models. The data is analyzed to conclude about system performance. The results from the mathematical approach and the testbed operation are compared to establish an optimized security framework for a real world CPS. Figure 7.4 shows the block diagram of the testbed constructed and its different cyber and physical components. A short description of the components of the CPS testbed is given immediately after the graph.

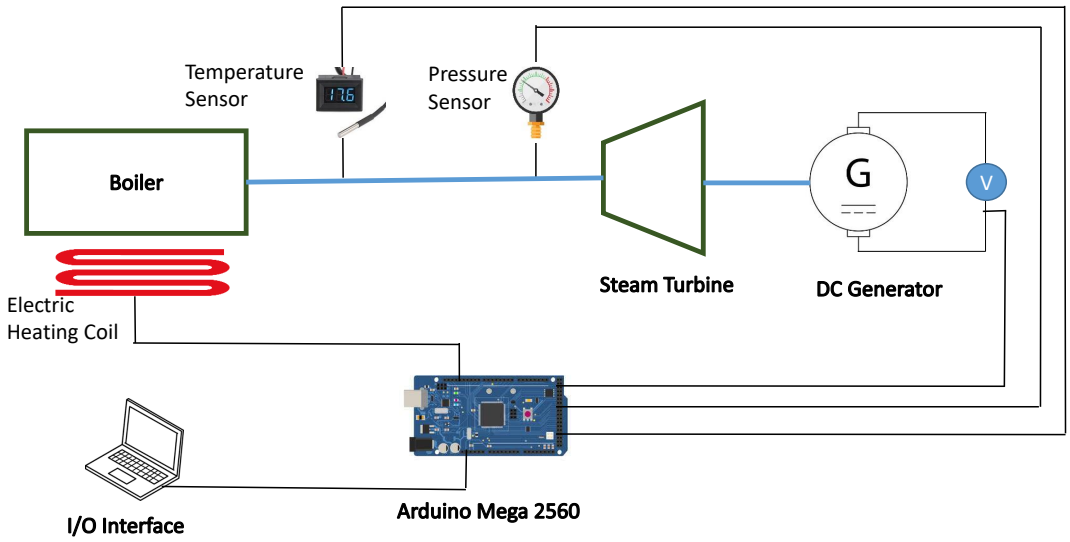

Figure 7.4: CPS testbed diagram.

- Boiler: The boiler is made of aluminum. Its length is 6.5 inches. Its diameter is 3.5 inches and its volume is 500 ml. The boiler can generate steam up to 20 psi gauge pressure. It has two safety pressure release valves to avoid any accident. The standard boiler comes with four alcohol lamps to generate steam. However, for this experiment an electric heater was used instead since it let a better control of the steam generation.

Figure 7.5: Steam generator used to build the CPS tesbed.

- Steam Turbine and Generator: The standard steam turbine came coupled with a DC generator. The turbine blades, shaft and gears are made of brass. The DC generator has a rated voltage of 12V but since the Arduino can handle a maximum of 5V, the generator ran at maximum 5V capacity.

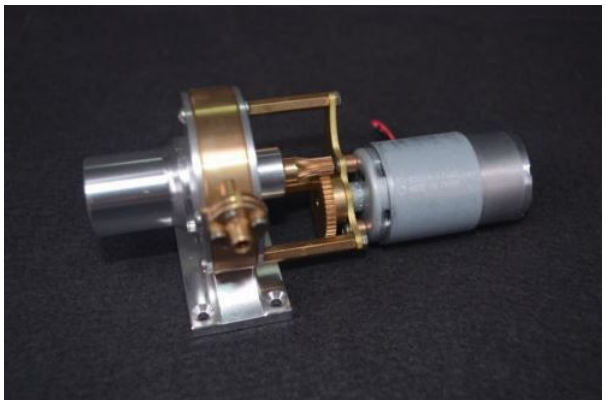

Figure 7.6: Steam turbine and DC generator used to build the CPS testbed.

- Temperature Sensor: To measure the steam temperature, a K type thermocouple temperature sensor was used. The thermocouple is attached to an Arduino compatible MAX6675 module that converts voltage generated by the thermocouple into a digital signal that can be read using an Arduino digital input pin. Its operating temperature range is 0 to 600^0C.

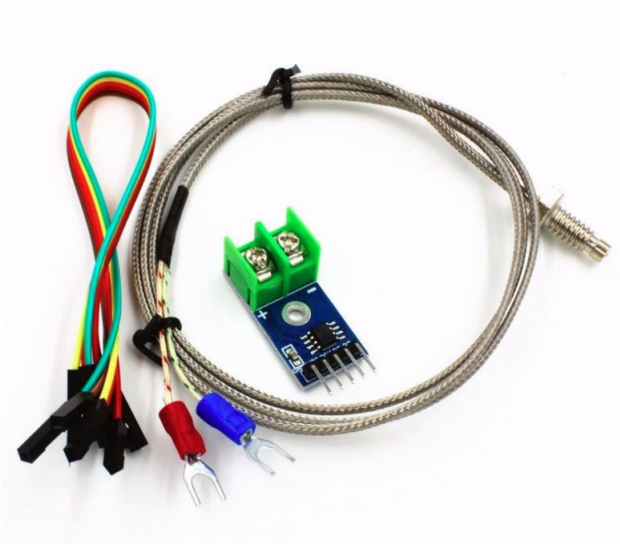

Figure 7.7: MAX6675 temperature sensor used to build the CPS tesbed.

- Pressure Sensor: The pressure sensor is an advanced strain gauge pressure transducer that converts strain from liquid pressure into a DC voltage ranging between $0.5V$ to $5.0V$. This voltage is a linear function of the steam pressure. The voltage can be read using the Arduino analog input pin and converted into pressure in psi. The operating range of this pressure sensor is 0 to 200 psi.

Figure 7.8: Pressure sensor used to build the CPS testbed.

- Arduino Boards: Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single-board micro-controllers and micro-controller kits for building digital devices and interactive objects that can sense and control objects in the phys-

ical and digital world [47]. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in pre-assembled form, or as do-it-yourself (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards or Breadboards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The micro-controllers are typically programmed using a dialect of features from the programming languages C and C++. Matlab also has hardware support package for Arduino and an Arduino can be controlled from the Matlab environment using Matlab code. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing Language project.

Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform and their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike.

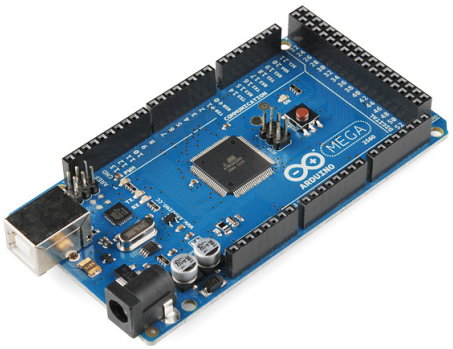

Figure 7.9: Arduino board used to build the CPS testbed.

The heater heats up the water in the boiler and generates steam. The steam passes through the pipe and is fed into the inlet of a steam turbine. The steam rotates the turbine and exists through the outlet. The rotating turbine drives the DC generator and generates electricity. Steam pressure and temperature are read using a steam pressure transducer and k-type thermocouple temperature sensor. The signal from both sensors are sent to an Arduino MEGA 2560 board. The generated voltage is also read by Arduino directly using analog input pin. The heater underneath the boiler is controlled by pulse width modulation using solid state relay and output from Arduino digital output pin. The Arduino boards are programmed to monitor and run the system in a closed loop feedback system autonomously.

A Matlab script was written to simulate the attacks (Random noise attack, Minimum value attack and offset attack) and implement the check blocks (Parity based check block, Threshold based check block and Watermarking check block) in the CPS testbed. The definition of the attack functions and the check blocks are the same as presented in section- 7.3, chapter-7. All the readings from the Arduino board were sent to a Matlab simulation model by using the Arduino add-in in matlab and USB cable. The actual simulation of attacks and check blocks happens inside the Matlab script and the command signal is then passed to the Arduino to control the heater of the boiler. There are three measurement signals (steam pressure, steam temperature, and generated DC voltage) and one control

signal (boiler heater signal) in the CPS. First the signals are read from the analog pin of Arduino. Then each of the signals may face one of the three attacks based on random selection. After that check blocks are applied to the signals Zero, one or more check blocks may be active based on the solution found from mathematical model. The check blocks checks each signal for any sign of potential attack. If the check blocks can detect an attack then it is recorded. If the check blocks can not detect an attack it is also recorded. The percentage of attack detection under a certain policy or method contrasted (deterministic GT , two-stage stochastic, greedy or random) determines the success of that policy.

Figure-7.10 shows the final set up of the CPS that has been built.

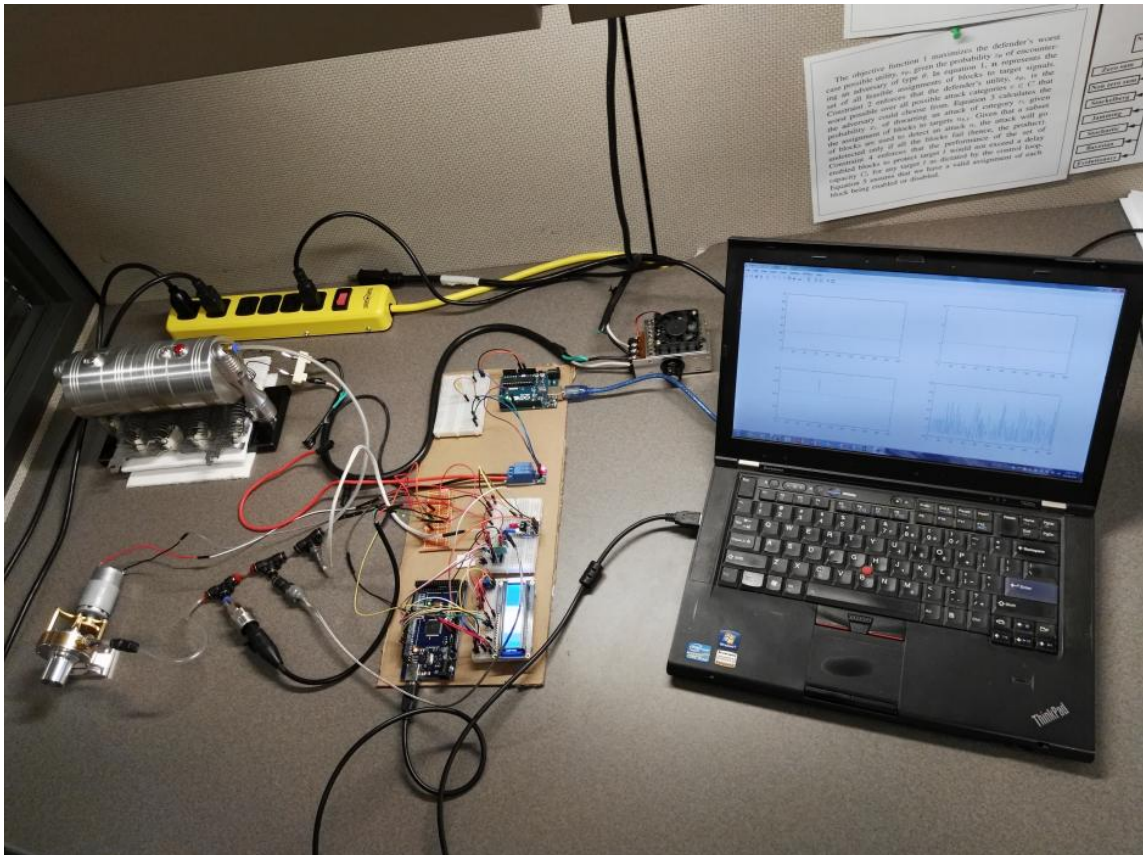

Figure 7.10: Final setup of the CPS testbed.

8. RESULTS

8.1 Results from the Deterministic Model

All the mathematical models were programmed in IBM Ilog CPLEX Studio 12.7.1 using OPL language and solved in a PC running on windows 10 professional operating system with core i7 processor with 3.2 GHz clock speed and 16GB DDR3 RAM. To assess the superiority of the Game Theory formulation, the outcome from the Game Theory approach is compared to the outcome of four other methods. The methods compared are:

1. Game theory- two relaxations, as explained in the chapter 5 are done in equations-5.3 and 5.5 to the model given by equations 5.1-5.5.

2. Game theory Mixed Integer, that solves the model comprised of Equations 5.1-5.5 but with only one relaxation of Equation 5.3. It means n_{st} is considered to be integer 0,1 for this model.

3. Random assignment, in which the assignment of blocks to targets is done randomly while satisfying the capacity constraints given in Equation 5.4. The randomly generated assignment is used to calculate the probability x_c and the worst case possible utility s_θ.The procedure for randomly finding the assignment of blocks to targets is repeated a pre-determined number of times to assess the variability on the results.

4. Greedy assignment, in which the following procedure is followed. A relative effectiveness for each assertion was calculated by dividing the effectiveness

of each assertion by the time to assert the assertion. Then total effectiveness of the assertions is computed by adding their relative effectiveness over the different attacks. Then the blocks are sorted by total relative effectiveness on descending order and kept on a list. The first ranked block is assigned at the highest possible amount to the most valuable target first and consecutively to the remaining less valuable targets if the capacity constraints are satisfied.The procedure continues attempting to assigning the next blocks in the list if the capacity constraint is not violated. The resulting greedy assignment permits to find x_c and the worst case possible utility s_θ.

5. Greedy-LP, in which the linear relaxation to the following mathematical programming model is solved to find the assignment of blocks to targets n_{st} and subsequently this assignment is plugged into the model in chapter 5 to find the resulting worst case utility

$$\max \quad \sum_{t\in T}\sum_{c\in C}(x_c U_t^d + (1-x_c)U_t^u) \tag{8.1}$$

$$x_c = \max_{s\in S} E_s^a n_{st} \qquad \forall c \tag{8.2}$$

$$\sum_{s\in S} n_{st} N_t T_s \leq C_t \qquad \forall t \tag{8.3}$$

$$\sum_{s\in S} E_s^a n_{st} \geq k \qquad \forall t, a \tag{8.4}$$

$$n_{st} \in 0, 1 \qquad \forall s, t \tag{8.5}$$

In the model above, the objective function 8.1 maximizes the utility of the defender when the target is undefended. Equation 8.2 is the relaxed constraint of Equation 5.3 that once the maximum function is transformed to a set of linear constraints causes the model to become an IP Equation 8.3 is the same capacity constraint as in the model in chapter 5. Equation 8.4 guarantees that for each

target there is a minimum level of effective coverage or protection against each attack. In Equation 8.4, k is a coverage constant whose value is set to a small number in the experiments (i.e., 0.2). By relaxing Equation-8.5 we obtain a fractional assignment of assertions to targets.

A 3^2 general full-factorial design was run to compare the performance of methods 1-5. The three different levels of effectiveness considered for the blocks S were Low, Medium and High. The three different levels of defenders' utility for the targets were Low, Medium and High. The low level of effectiveness was a random number less than or equal to 0.5 for all blocks against all attacks. The medium level of effectiveness was a random number between 0.5 and 0.7 and the high level of effectiveness was a random number greater or equal to 0.7.

The total utility (i.e. value) of the targets was set to add to 700. At the low experimental level, all the targets were considered to have equal value (i.e. 116.67). At the medium experimental level, one of the targets had higher value randomly chosen between 200 and 300 and the rest had equal value.At the high experimental level one target had a higher value randomly chosen between 300 and 450 and the remainder targets had equal value. The values for the rest of the parameters are mentioned in Table 8.1

Table 8.1: Parameter values.

Parameter	Value
N_t	[10, 10, 10, 10, 10, 10]
T_s	[2, 1, 1.5]
C_t	[40, 40, 40, 40, 40, 40]

The random assignment procedure was performed 5 times over each one of the nine experimental conditions. Besides, at each one of the conditions, the random selection of values for effectiveness and utility was replicated 5 times. Then, the

approaches 1, 2, 4,and 5 were run a total of $3^2 * 5 = 45$ times and the random approach (i.e., approach 3) was run $3^2 * 5 * 5 = 225$ times. Consequently, in each experimental condition the resulting worst case utility is averaged over 5 runs for approaches 1,2,4 and 5 ad over 25 runs for approach 3. A summary of these average worst case utility outcomes is presented in Figure 8.1.

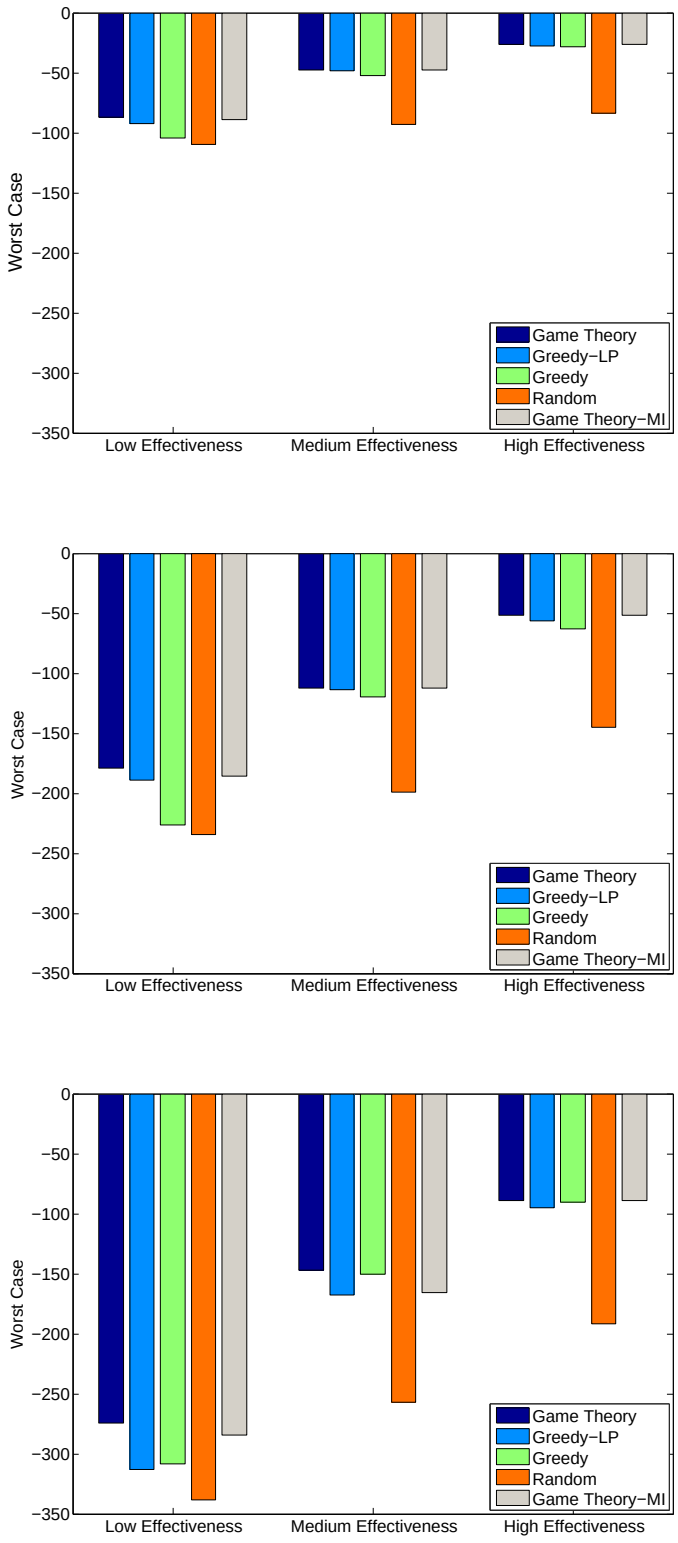

Figure 8.1: Average worstcase utility for different levels of Utility (top to bottom Low, Medium and High).

The graph in the top of Figure-8.1 presents the average worst case utility for different effectiveness levels when the utility of the targets is low. As expected from the linear programming theory, it can be seen that Game Theory two relaxation method has the best worstcase utility in all the scenarios. Greedy-LP works better than Greedy or Random approach when the effectiveness is low but for medium and high effectiveness Greedy gives better results than Greedy-LP or Random approach. As expected. random approach gives the most negative worst case utility in all the effectiveness levels.

The graph in the middle in Figure-8.1 shows the average worst case utility for different effectiveness levels when utility is medium. Again, the Game Theory two relaxation approach has the best worst case utility in all the scenarios; Greedy-LP works better than Greedy or Random approach when the effectiveness is low but for medium and high effectiveness Greedy gives better result than Greedy-LP or Random approach. For all the scenarios in this figure, the Random approach gives the less desirable worst case utility.

The graph in the bottom of Figure-8.1 represents the average worst case for different effectiveness level when utility is high. It is evident that the Game Theory two relaxation approach has the best worst case utility in all the scenarios. The Greedy approach gives the second highest worst case value. Note that for medium and high effectiveness, the worst case utility for the Greedy approach is very close to the one for Game Theory. Greedy-LP gives better worst case utility than Random but less than Game Theory or Greedy.

A one way ANOVA test was done to determine whether there is any statistically significant differences between the five approaches. The p-value of the ANOVA test is 0.00 (figure-8.2), which means there is statistically significant difference in means between different approaches. To further understand the difference in

means, a Tukey pairwise comparison is performed and the results are presented in figure-8.4. It can be seen from the pairwise comparison graph that our game-theoretic approaches (GT and GT-MI) and Greedy, form a group that has the least difference in means among themselves. Random and Greedy-LP form another group.

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Approach	4	182400	45600	6.05	0.000
Error	220	1658279	7538		
Total	224	1840680			

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
86.8195	9.91%	8.27%	5.77%

Means

Approach	N	Mean	StDev	95% CI
Game Theory	45	-111.9	77.4	(-137.4, -86.4)
Greedy	45	-128.2	91.1	(-153.7, -102.7)
Greedy-LP	45	-169.0	96.8	(-194.5, -143.5)
Random	45	-181.9	86.1	(-207.4, -156.4)
Game Theory-MI	45	-116.7	81.2	(-142.2, -91.2)

Pooled StDev = 86.8195

Figure 8.2: One way ANOVA result

Tukey Pairwise Comparisons

Grouping Information Using the Tukey Method and 95% Confidence

Approach	N	Mean	Grouping
Game Theory	45	-111.9	A
Game Theory-MI	45	-116.7	A
Greedy	45	-128.2	A B
Greedy-LP	45	-169.0	B C
Random	45	-181.9	C

Means that do not share a letter are significantly different.

Figure 8.3: Tukey test Summary.

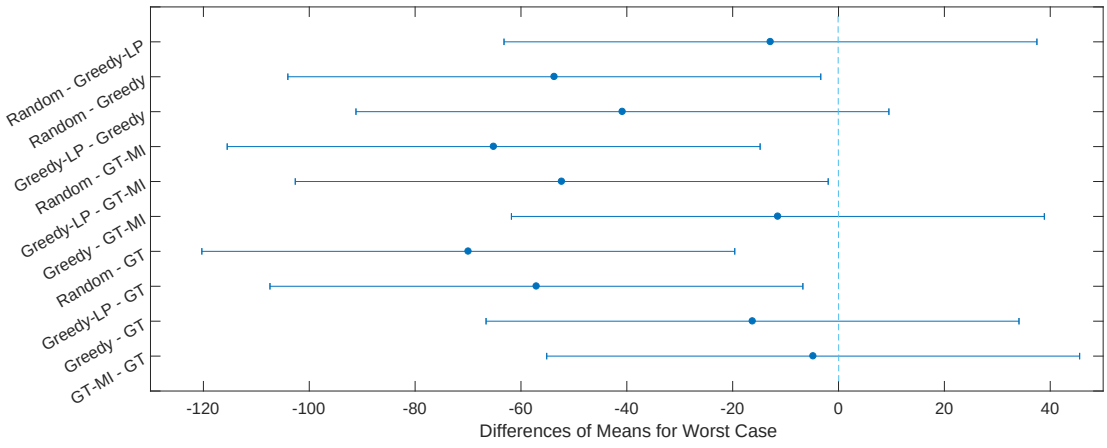

Figure 8.4: Tukey pairwise graph.

In summary, Game Theory gives the best (i.e. largest negative value) worst case utility among all the approaches compared and for all the test scenarios. Greedy-LP gives second highest worst case when effectiveness is low but for medium and high effectiveness Greedy gives better results. Random approach gives the worst worstcase utility for all test scenarios.

The relative performance of the Greedy, Greedy-LP, Random, and Game Theory Mixed Integer approaches compared to Game Theory Linear Relaxation is

presented in the following Performance Profile graph [48]. Figure-8.5 shows that Random and Greedy LP gave the same worst case utility as Game Theory Linear Relaxation only in less than 20% of the experimental runs performed while Greedy did it in about 80% of them. Also, near 95% of the instances run for Greedy algorithm have less than 50% error vs. the Game Theory Linear Relaxation approach. For Greedy LP and Random the percentage of instances with error below 50% vs. Game Theory Linear Relaxation is only about 40-50%, Game Theory Mixed Integer gave 95% data points within 20% error compared to Game Theory Linear relaxation

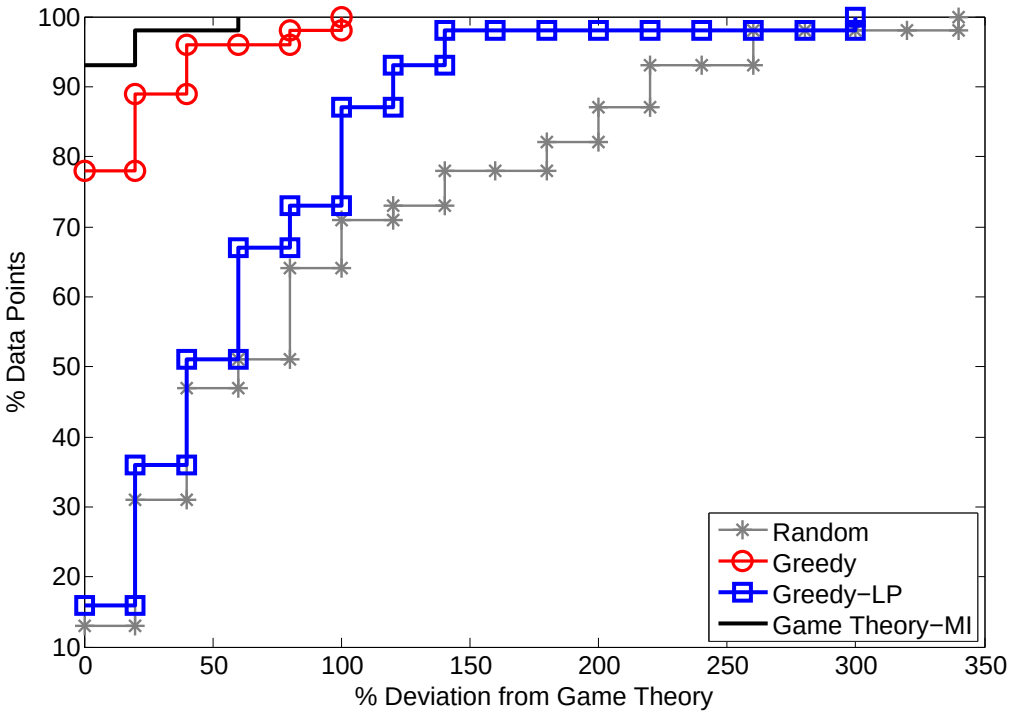

Figure 8.5: Performance profile graph.

8.2 Results from Two-Stage Stochastic Programming (Two-SSP) Model with stochastic number of signals

The Two-SSP stated in section 7.1.1 was solved for nine different cases (three levels of utility for targets and three levels of effectiveness for the assertions) as the deterministic model in chapter 5. The value of the parameters are presented in Table 8.2. To assert the effectiveness of the model, value of stochastic solution (VSS) was calculated. VSS is defined by the difference between the worstcase of two-stage stochastic programming model solved for first stage assignment along with second stage recourse action and two-stage stochastic model where the first stage assignment is given from previous deterministic model (presented in chapter 5) and solved for only second stage recourse action. The worstcase utility under each of the 9 experimental settings were averaged over the 5 runs and they are presented in Figure 8.6 and the VSS is presented in figure-8.7

Table 8.2: Parameter values for Two-SSP.

Parameter	Value
P_e	[0.2, 0.3, 0.5]
N_{te}	$\begin{bmatrix} [7,7,7,7,7,7] \\ [10,10,10,10,10,10] \\ [11,11,11,11,11,11] \end{bmatrix}$
T_s	[2, 1, 1.5]
T_a	[2.4, 1.2, 1.8]
C_t	[40, 40, 40, 40, 40, 40]
k	$(-0.1, 0.1)$

Figure-8.6 shows the comparison between the worstcase utility of the two-SSP solution and the deterministic solution plugged into two-SSP model. It can be seen that the worstcase utility improves when solved as two-SSP than when solution from deterministic model is plugged into two-SSP model. The gain in utility when solved as two-SSP is the VSS. The VSS is presented in Figure 8.7. The inner caption in the figure 'Low' 'Mid' and 'High' indicates low, medium and high effec-

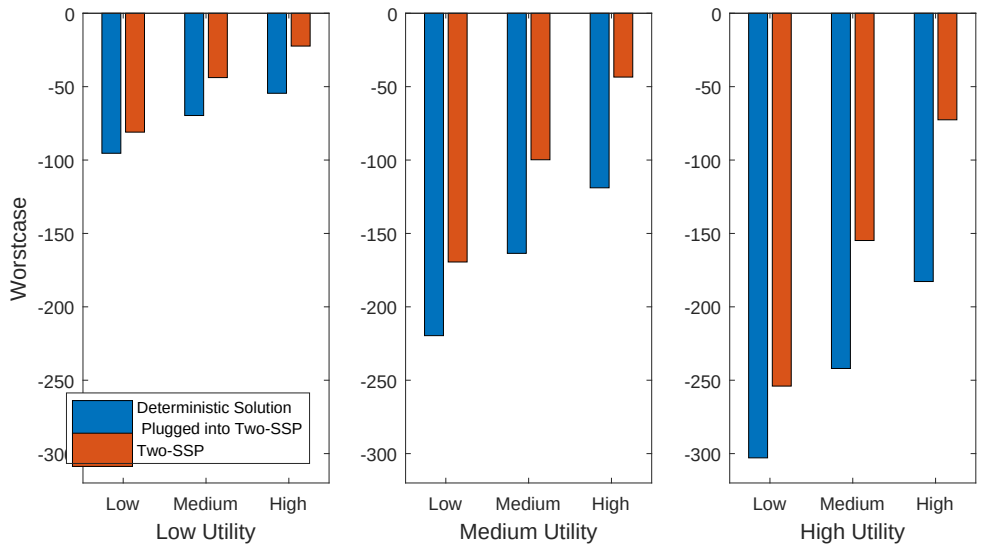

Figure 8.6: Worstcase comparison of two-SSP and two-SSP with solution plugged in from deterministic model.

tiveness of assertions. It can be seen that the VSS increases with the increase in effectiveness and utility. A two way ANOVA test was performed to study whether changing the utility and effectiveness had a statistically significant effect on VSS. The effect of both effectiveness of blocks and utility of targets on VSS was found to be statistically significant. The details of ANOVA can be found in appendix B.

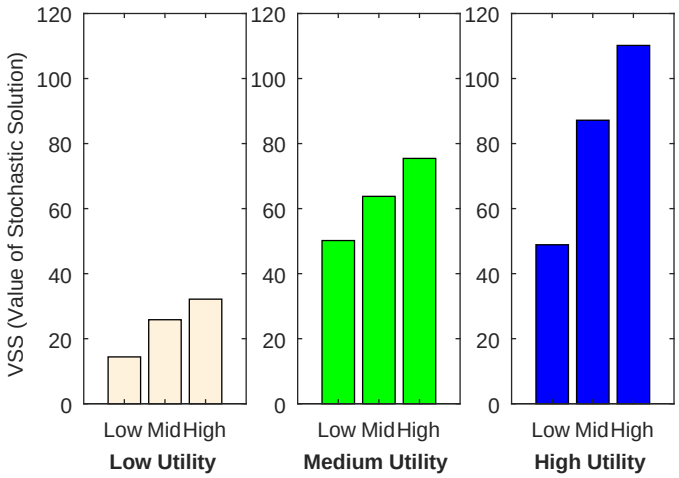

Figure 8.7: VSS of different utility for targets and different effectiveness of blocks.

A sample solution of the two-SSP model is presented here-

Worstcase $S_\theta = -40.0$

$$n_{st}= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0.086 & 0 & 0.1 & 0.1 & 0.1 & 0.1 \\ 1 & 0.9 & 1 & 0.81 & 0.94 & 0.82 \end{bmatrix}$$

$$Z_{ste}= \begin{bmatrix} \begin{bmatrix} 0 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 \\ -0.086 & 0 & -0.1 & -0.1 & -0.1 & -0.1 \\ 0 & -0.1 & -0.1 & 0 & -0.1 & -0.1 \end{bmatrix} \\ \begin{bmatrix} 0 & -0.1 & -0.1 & -0.1 & -0.1 & 0 \\ 0 & 0 & -0.1 & -0.1 & 0 & -0.1 \\ 0 & -0.1 & 0 & -0.1 & -0.1 & 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & -0.1 & 0 & -0.046 \\ 0.1 & 0 & 0 & -0.1 & -0.1 & -0.1 \\ 0 & 0.1 & 0 & -0.02 & 0 & -0.1 \end{bmatrix} \end{bmatrix}$$

8.3 Results from the Piecewise Linear Model

The slopes and breakpoints for the piecewise linear functions that model the times T_s to enable the 3 blocks in the first stage ($s = 1, 2, 3$) and the times T_a to implement recourse actions are in Table 8.3.

Table 8.3: Slopes and breakpoints for the piecewise linear functions.

Breakpoints	T_s			T_a		
	T_1	T_2	T_3	T_{a1}	T_{a2}	T_{a3}
0.10	0.10	0.05	0.08	0.12	0.06	0.09
0.20	0.25	0.13	0.19	0.30	0.15	0.23
0.30	0.45	0.23	0.34	0.54	0.27	0.41
0.40	0.75	0.38	0.56	0.90	0.45	0.68
0.50	1.10	0.55	0.83	1.32	0.66	0.99
0.60	1.50	0.75	1.13	1.80	0.90	1.35
0.70	1.68	0.84	1.26	2.02	1.01	1.51
0.80	1.85	0.93	1.39	2.22	1.11	1.67
0.90	2.00	1.00	1.50	2.40	1.20	1.80

The piecewise model solves in about the same computational time as stochastic model (80 to 100 ms). The worst-case utility and the matrices for the first-stage marginal block assignments, n_{st} and the second-stage recourse actions for the block assignments, Z_{ste}, are given below.

Worstcase $= -40.0$

$$n_{st} = \begin{bmatrix} 1 & 0.8 & 0.8 & 0.8 & 0.8 & 0.8 \\ 0 & 0 & 0.1 & 0.886 & 0 & 0.1 \\ 1 & 0.85 & 0.85 & 0.7 & 0.75 & 0.8 \end{bmatrix}$$

$$Z_{ste}= \begin{bmatrix} \begin{bmatrix} 0 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 \\ 0 & 0 & -0.1 & 0 & 0 & -0.1 \\ 0 & -0.1 & -0.1 & 0 & -0.1 & -0.1 \end{bmatrix} \\ \begin{bmatrix} 0 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 \\ 0 & 0 & -0.1 & 0 & 0 & -0.1 \\ 0 & -0.1 & -0.1 & 0 & -0.1 & -0.1 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.1 & 0 & 0 & -0.1 \\ 0 & -0.1 & -0.1 & 0.1 & 0.1 & 0 \end{bmatrix} \end{bmatrix}$$

Additional experiments were performed to observe the effect of the ratio T_a/T_s on the expected worstcase utility of model in subsection 7.1.3 with varying the delay capacity C_t at 3 levels, 10, 20, and 40. Results are presented in figure-8.8. The results show that when the second stage assignment takes less time than first stage assignment ($T_a/T_s < 1$), the expected worstcase utility is better than when $T_a/T_s > 1$. This trend is more observable if the system delay capacity C_t is low (e.g. 10). This can be explained by the fact that at lower capacity levels the program can assign a much lower fraction of assignments compared to higher capacity level. So, the ratio of Z_{ste}/n_{st} is lower at higher capacity levels and higher at lower capacity levels. As the ratio of Z_{ste}/n_{st} goes higher, the higher value of t_a (time to assign second stage assignment) affects the worstcase utility more.

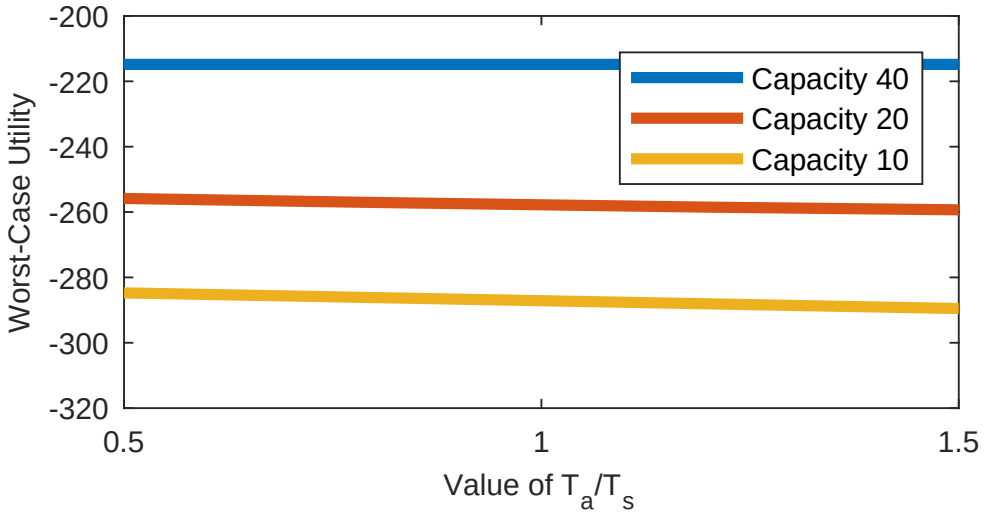

Figure 8.8: Effect of the ratio T_a/T_s on worstcase utility at 3 different capacity levels.

8.4 Results from Two-Stage Stochastic Programming (Two-SSP) Model with stochastic number of signals and stochastic effectiveness

The model presented in section 7.1.2 was solved considering three random scenarios for the effectiveness of the blocks, E^a_{se}, and three different scenarios for the number of signals,N_{te}. The values for N_{te} and its probabilities are the same as in Section 8.2. Thus, the total number of scenarios in this model is $e' = 9$. The solution to this model is given below. The worstcase utility obtained is similar to the one in section-8.2 but the assignment matrix is different. The reason is that due to stochastic effectiveness of the blocks the program assigns them differently to achieve maximum worstcase. As other parameters were same as section 8.2 the worstcase utility is same.

S_θ=-40.374

$$n_{st}=\begin{bmatrix} 1 & 1 & 1 & 0.99 & 1 & 1 \\ 0 & 0.12 & 0 & 0 & 0 & 0 \\ 1 & 0.88 & 0.84 & 0.86 & 0.95 & 0.91 \end{bmatrix}$$

$$Z_{ste}=\begin{bmatrix}\begin{bmatrix}0 & 0 & -0.1 & -0.1 & 0 & -0.1\\ 0 & -0.1 & 0 & 0 & 0 & 0\\ 0 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1\end{bmatrix}\\ \begin{bmatrix}0 & 0 & -0.1 & -0.1 & 0 & -0.1\\ 0 & -0.1 & 0 & 0 & 0 & 0\\ 0 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1\end{bmatrix}\\ \begin{bmatrix}0 & 0 & -0.1 & -0.1 & 0 & -0.1\\ 0 & -0.1 & 0 & 0 & 0 & 0\\ 0 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1\end{bmatrix}\\ \begin{bmatrix}0.1 & -0.1 & 0 & -0.1 & 0 & -0.1\\ 0 & 0 & -0.1 & -0.1 & -0.1 & 0\\ 0.1 & -0.1 & 0 & -0.1 & -0.1 & -0.1\end{bmatrix}\\ \begin{bmatrix}0.1 & -0.1 & 0 & -0.1 & 0 & 0\\ 0 & -0.1 & -0.1 & 0 & 0 & -0.1\\ 0.1 & -0.1 & -0.1 & 0 & 0 & 0\end{bmatrix}\\ \begin{bmatrix}0.1 & -0.1 & -0.1 & -0.1 & 0 & 0\\ 0 & 0 & -0.1 & -0.1 & 0 & 0\\ 0.1 & -0.1 & 0 & -0.1 & 0 & 0\end{bmatrix}\\ \begin{bmatrix}0.05 & 0 & -0.1 & -0.068 & 0 & 0\\ 0 & 0 & 0 & 0 & -0.1 & 0\\ 0.053 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1\end{bmatrix}\\ \begin{bmatrix}0.08 & 0 & -0.1 & -0.1 & 0 & 0\\ 0 & 0 & 0 & 0 & -0.1 & -0.1\\ 0 & -0.1 & 0 & -0.1 & 0 & -0.1\end{bmatrix}\\ \begin{bmatrix}0.017 & 0 & 0 & -0.1 & 0 & 0\\ 0 & -0.1 & -0.1 & -0.1 & -0.1 & 0\\ 0.1 & -0.03 & -0.052 & 0 & -0.07 & 0\end{bmatrix}\end{bmatrix}$$

The model was also run 45 times by doing 5 runs under 9 different experimental settings that correspond to the 3 levels of utility or importance for the targets (low, medium, high) and 3 levels of effectiveness for the blocks described in Section 8.1. Figure 8.9 shows the worst case utilities for each experimental setting.

Figure 8.9 shows the average worstcase utilities over 45 runs for the different mathematical programming models presented in this thesis. The figure shows that the model including the piece-wise linear functions to model the times to enable blocks, produces a better value for the worst case utility if compared to the ones with fixed times to enable the block assignments. The explanation to this result is that in piece-wise linear model for the fractional assignment the time required to enable the block is lower than in two-SSP models. And if N_{te} is low, according to equation-7.4, n_{st} and Z_{ste} will be higher resulting in a better worstcase utility.

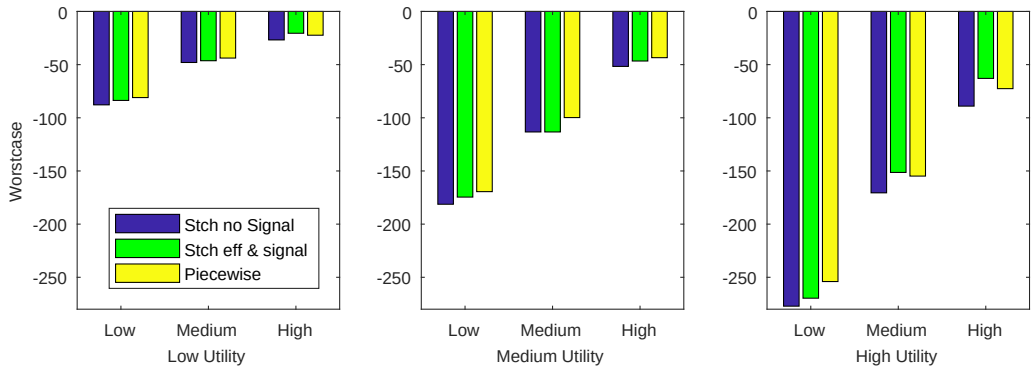

Figure 8.9: Worstcase Utility for Two-SSP- Stochastic Number of Signals, Two-SSP - Stochastic Effectiveness and Stochastic Number of Signals, and Two-SSP - Stochastic Number of Signals with Piece-wise Linear Functions for the times to enable blocks.

8.5 Results from the Model Expansion

The success of the approach taken in this thesis largely depends on the processing time the program takes to solve the optimization problem. To compare the processing time of different problem sizes a problem expansion experiment was done. With three different problem size and three different approaches (deterministic, two stage stochastic with stochastic number of signal and two stage stochastic with stochastic number of signal and stochastic effectiveness). The three different problem sizes are-

Case-1: A model with 30 targets, 10 assertions and 6 attacks.
Case-2: A model with 50 targets, 10 assertions and 10 attacks.
Case-3: A model with 100 targets, 20 assertions and 20 attacks.

Table 8.4: Summary of Model Expansion experiments.

	Deterministic Model			Two-SSP with Stochastic Number of Signal			Two-SSP with Stochastic Number of Signal ad Effectiveness		
	$Case-1$	$Case-2$	$Case-3$	$Case-1$	$Case-2$	$Case-3$	$Case-1$	$Case-2$	$Case-3$
Time (s)	5.01	56.12	810.5	19.81	8.71	875.78	136	323	6606
# of Decision Variables	2642	7002	48002	4591	10251	59501	11642	22002	108002
# of Constraints	2970	6560	46100	5490	11750	66500	11790	22250	107500
# of Nodes Inspected by $B\&C$	2	15	41	250	141	2618	288	40685	187120
# of Iterations	312	1316	8370	103310	43785	448202	713813	2187890	8236412

The summary of the results for the model expansion test is presented in the table-8.4. This table presents the computational times in seconds and other relevant resulting features from the Branch and Cut solution method for each model tested, such as the number of nodes inspected and the number of branch and cut iterations performed. The computer used in the test was a PC running on a 64bit Windows 10 professional operating system with Intel core i7 processor with 3.2GHz of clock speed and 16GB of RAM. Each of the cases was run three times

and average time was taken. The table confirms that it is possible to solve problems with up to 100 targets, 20 assertions and 20 attacks in 810, 876 and 6,606 seconds (110 minutes) for each one of the models compared.

The results show that in general, when the problem size is increased, the computational time increases. One exception occurred for $Case - 2$ in the Two-SSP with Stochastic Number of Signals model. The running time behavior in this case is consistent with the fact that the program inspected less $Nodes$ and went over less $Iterations$. This may happen for various reasons; i.e. the B&C may terminate a branch based on in-feasibility or the solution is not promising vs incumbent solution etc. Thus, the experiment corroborates that the computational time for the models is not only dependent of problem size (i.e. number of targets, attacks and check blocks) but also related to the number of $Nodes$ the program ends inspecting and the number of branch and cut $Iterations$.

Figure 8.10 and figure 8.11 shows snapshots of the statistics window of the Cplex IDE when the program was running a problem (case-3 of Two-SSP with stochastic number of signals and effectiveness) and after it was solved, respectively. Figure 8.11 shows that the B&C searches through nodes to find integer solutions. If during the exploration of active partial solutions (i.e. nodes pending to explore), an integer solution is found, it is marked by the yellow dot on the green line. The best integer solution found so far is also depicted (red line). When the best node (red line) and best integer (green line) coincide it means there are no more nodes to explore and the best integer solution to the problem is found. At that time, the screen displays the cell "Remaining nodes" as zero. The cell "Nodes" presents the total number of nodes explored.

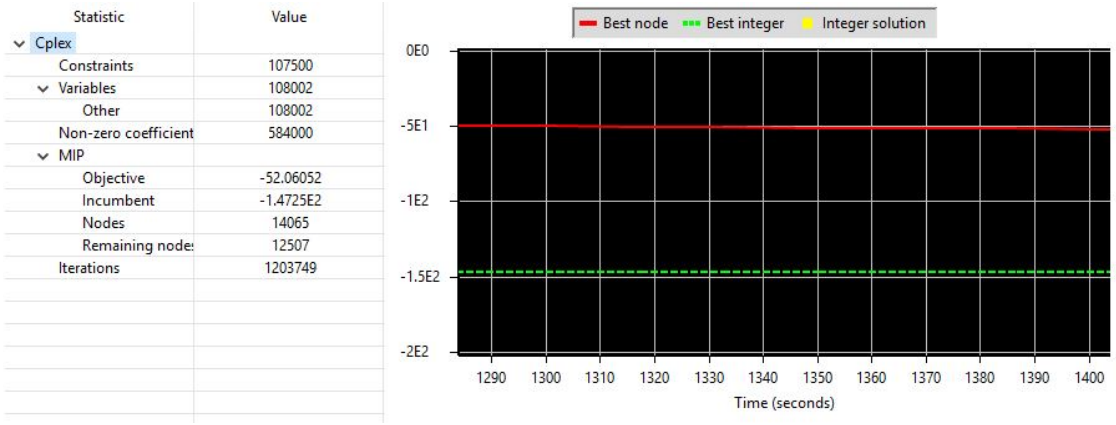

Figure 8.10: Statistics window of IBM OPL Cplex IDE when the program is running.

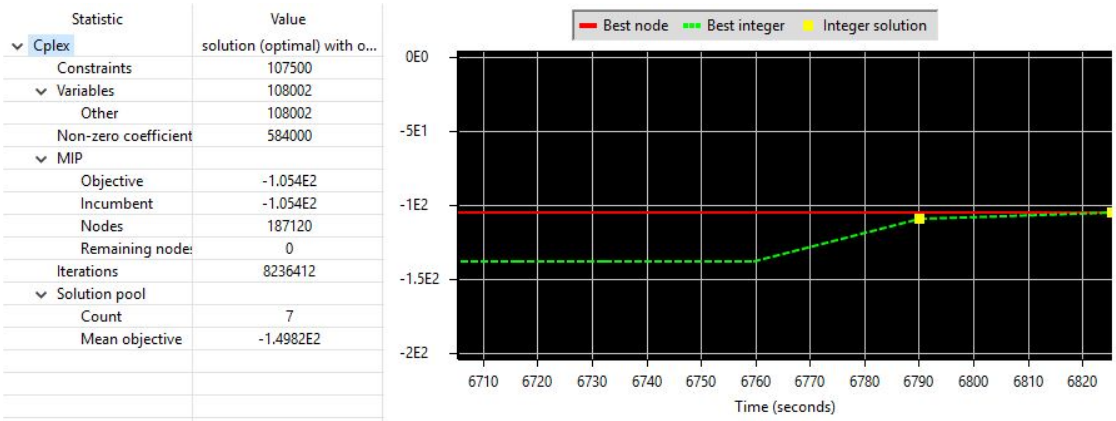

Figure 8.11: Statistics window of IBM OPL Cplex IDE when the program is solved.

8.6 Results from the Simulation Model

Figure 8.12 displays the behavior of the velocity of the lead vehicle (left graph), velocity of the host vehicle (middle graph), and the distance of the host vehicle from the lead vehicle (right graph) over time. The graphs contrast the behavior of the system in under attack vs. under no attack. Under no attack, the model behaves as designed. The host vehicle maintains a preset cruise speed of 30 mph while maintaining a safe distance (200 ft) from the lead vehicle. However, when the attack functions are activated the system no longer behaves as designed. The speed of the host vehicle increases beyond the set speed and the distance also goes below the set distance. In the graph, the point where the distance becomes negative indicates an accident occurred.

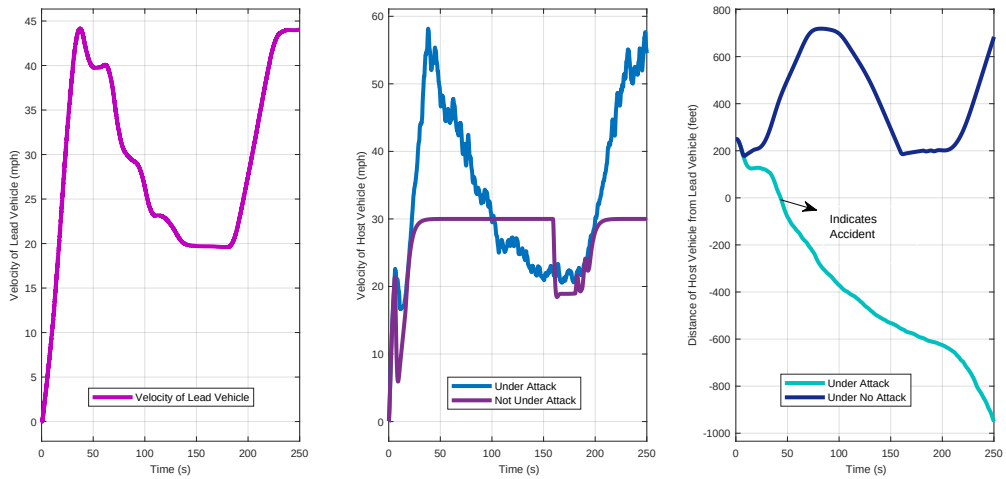

Figure 8.12: Behavior of the system under Cyber-Attack.

The Simulink model and the Two-Stage Stochastic Programming (Two-SSP) model were calibrated to have similar values for the utility of the targets, time to run the check blocks and delay capacity of the targets. The data in the models is also similar for number of signals, number of attacks, number of check blocks, and effectiveness matrix. The effectiveness matrix E_s^a of the simulation model comes from simulating each attack against each of the check blocks and taking the percentage of times the check blocks can detect an attack. This matrix was

then used as E_s^a in solving the mathematical models. The block assignments come from solving the following four different models/methods: (i) Two-SSP (ii) deterministic model with known number of arriving signals, N_t to each target (iii) a method where the check blocks are assigned randomly and (iv) a Greedy method that seeks to protect the target with highest utility first using the most effective check block available and then looks to protect the next most valuable target. The block assignment matrix is then plugged into simulink model to simulate the system.

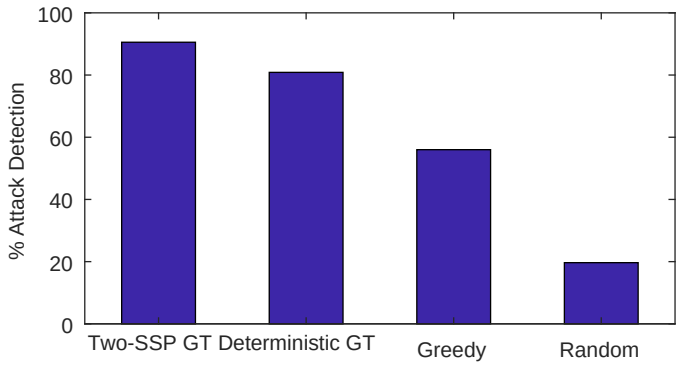

Figure 8.13: Rate of attack detection of Simulink model.

The Simulink model was run for 250 seconds with a discrete time step of 0.05 second which is sufficiently long time to observe the system behavior given that in each time step an attack was simulated. While the simulation was running the check blocks were active or inactive according to the assignments obtained from the models. The Simulink model recorded the number of times the check blocks detected an attack. These results are presented in Figure 8.13. The comparison of the percentage of attacks detected by the different models indicates that the Two-SSP provides the best results closely followed by Deterministic game theoretic method. The Greedy approach seeks to protect the most valuable target but can leave other targets vulnerable, thus failing to detect several attacks. The Random method assigns the check blocks randomly and has the worst performance among the four models.

Further model comparisons were done by estimating the probability of thwarting an attack through the percentage of times an attack was detected by the Simulink model. This estimated probability permitted to calculate the worst-case utility in each of the methods compared. Figure 8.14 presents the worst-case utility results.

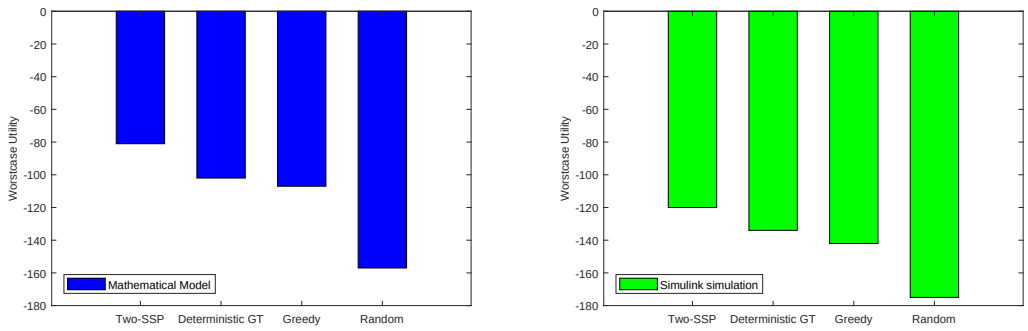

Figure 8.14: Worstcase comparison.

Figure 8.14 corroborates that in the Simulink model the assignment from the Two-SSP gives the "best worst case utility" and that the assignment from the random approach gives the worst "worst case utility" (see blue bars). This trend is observed also when comparing the worstcase utility of the Two-SSP to the worst case utilities from plugging into the Two-SSP as first-stage assignments the Deterministic GT, Greedy and Random methods solutions (yellow bars). Figure 8.14 also shows a difference in the worst case utility provided by the Two-SSP model and the Simulink model The difference between Simulink and Two-SSP partly arises from the estimation of the probability of thwarting an attack done in the Simulink model. In mathematical model the probability of thwarting an attack comes from multiplying the assertions (n_{st}) with the effectiveness (E_s^a) of check block from effectiveness matrix and taking the maximum of the results. But in simulink simulation the probability of thwarting an attack is directly calculated from the percentage of instances the check blocks can detect an attack. This approximation in simulink model is responsible for the difference in the values of worstcase utility.

It was also tested how the different methods to assign blocks perform under false attacks. The false positive percentage (i.e. the probability of saying that the CPS is under attack when it is not) for different approaches is presented in Fig-8.15. It can be seen that the Two SSP (i.e. Stochastic GT) approach gives the least percentage of false positive among the different approaches compared. The Deterministic Game Theory model is the second best.

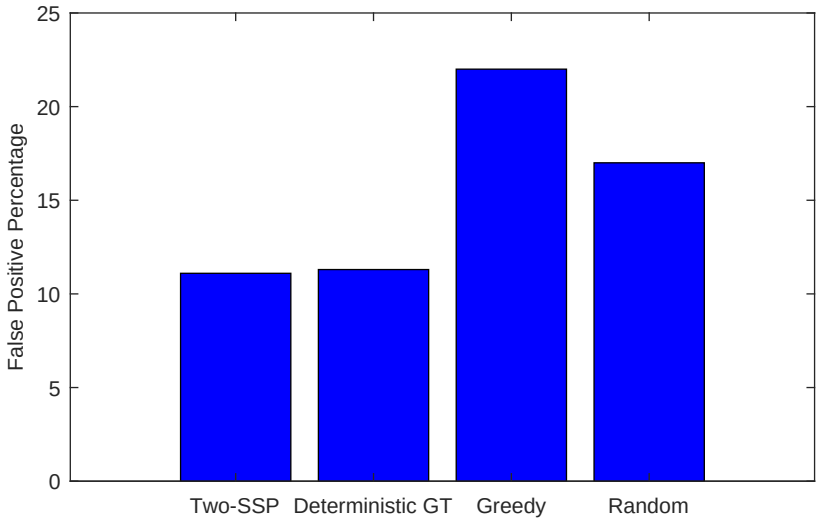

Figure 8.15: False positive percentage for the simulated methods.

8.7 Results from the CPS Testbed

Figure-8.16 shows the behavior of the CPS testbed under no-attack and under attack situations (when no check blocks are applied). It can be seen that when there is no attack the values of the signals are stable (i.e. the steam pressure remains between 22 psi and 24 psi, steam temperature remains between 110 °C and 120 °C and generated voltage remains between 0.5V and 0.7V). However, when the attacks are introduced it can be seen that the the attack functions intercept the signals, change the values and feed the changed values to the control module. Thus the cyber-attacks can destabilize a CPS and lead to catastrophic accident.

Further investigations were done on the CPS by implementing the different check block assignments found from solving mathematical model for the different policies or methods studied (i.e. Deterministic Game Theory, Two-Stage Stochastic Game Theory, Greedy, and Random). The check block assignments were found from solving the mathematical models using the parameters from the CPS testbed. The model solutions (i.e. the resulting check block assignments) were implemented while running the Matlab script used to simulate the attacks on the CPS testbed. While the Matlab script was running along with the CPS testbed, if an attack was detected by the check blocks it was recorded. A total of three runs were performed and average values were taken. Figure 8.18 shows the percentage of attack detection for each one of the methods implemented in the CPS testbed. It can be seen that Two-SSP model can detect highest percentage of attacks closely followed by deterministic GT model. Greedy and random model follows subsequently. The trend observed here is consistent with one found from the mathematical solution of the models and also with the Simulink simulation. Figure-8.17 highlights the time steps of detection of attack when check blocks are applied. In the figure black circles indicate that an attack has been detected and the red dots indicate that no-attack has been detected. It is also visible that non-detection pattern is fairly

random.

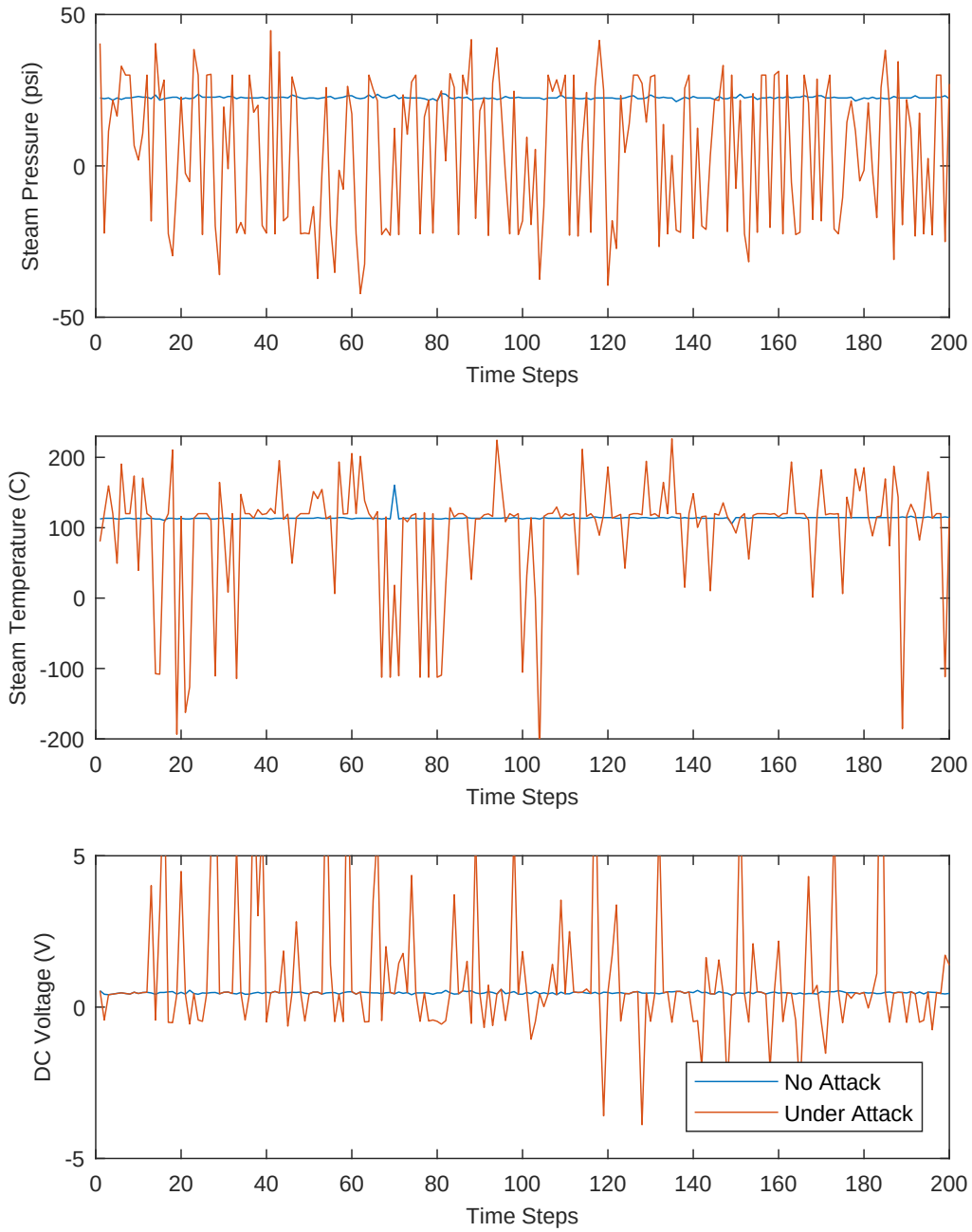

Figure 8.16: Behavior of CPS testbed under attack and no-attack condition.

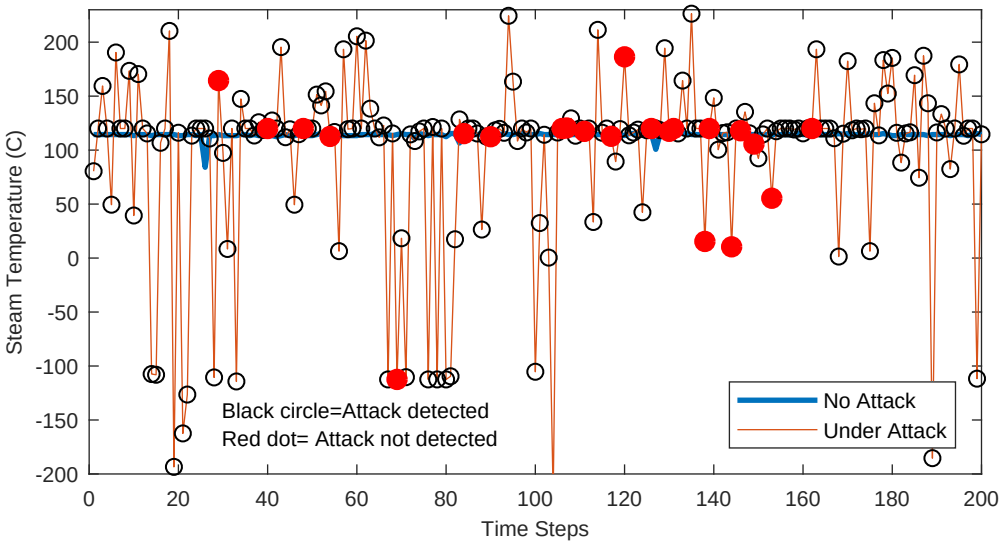

Figure 8.17: Attack detection on a target signal.

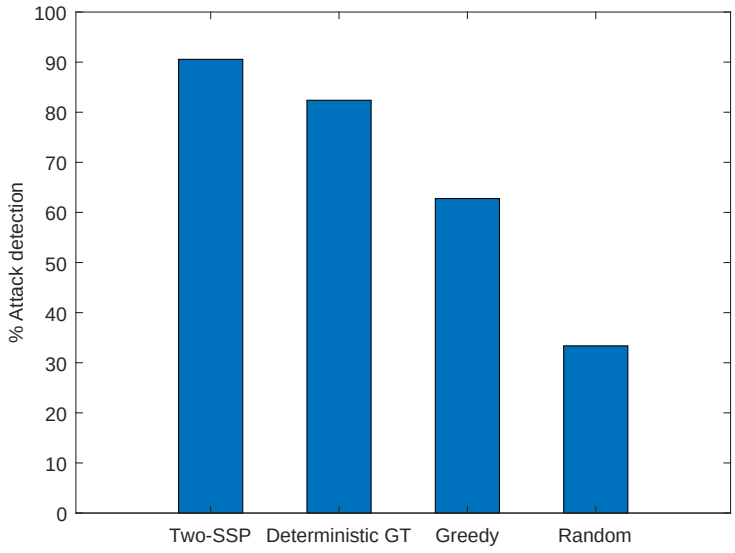

Figure 8.18: Percentage of attack detection in CPS testbed.

To calculate the worstcase utility of the models, the attack detection percentage can be used to approximate the probability of thwarting an attack, x_c and such estimated probability is plugged into equation 5.2 to finally compute the worscase utility using the models objective function. Figure 8.19 shows the comparison for worstcase utilities among the different models. It is evident that Two-SSP model has the best worstcase utility than all other models. Deterministic GT closely follow two-SSP and then greedy and random models occupy the third and fourth

place. A gap between the worst case values from the mathematical models and the ones from the CPS testbed is observed in the figure. The reason behind is that the 'percentage of attack detection' was approximated as 'probability of thwarting an attack x_c' where as in mathematical model it was calculated differently.

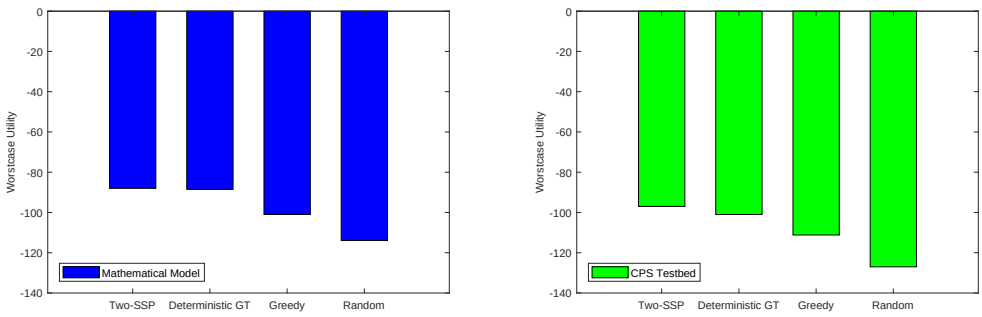

Figure 8.19: Worstcase comparison of CPS testbed.

Figure-8.20 shows the comparison of the type one error or percentage of false positives for the models contrasted. The type one error or percentage of false positives is the probability of rejecting the null hypothesis when it is true, and for this cybersecurity problem it is the probability that the check blocks detect an attack when there was no attack. The figure also shows that Two-SSP gives the least percentage of false positives closely followed by deterministic GT.

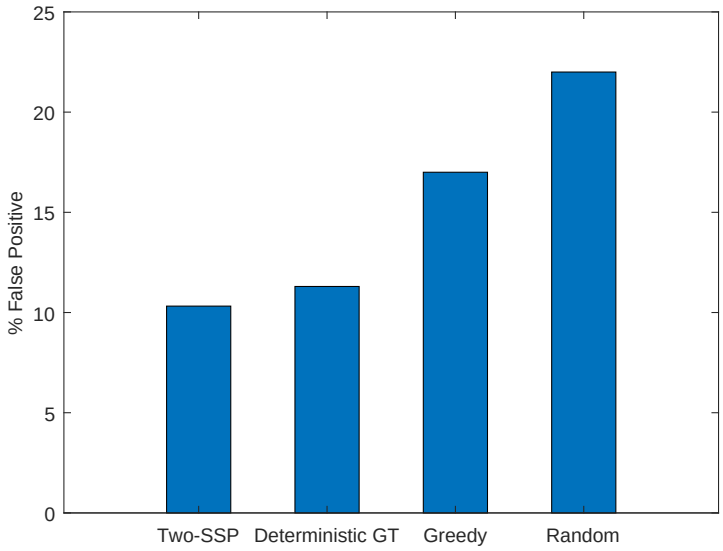

Figure 8.20: Percentage of false positives in CPS testbed .

9. CONCLUSION

An important component in securing CPS is ensuring that the correct control and measurement signals are propagated within the CPS control loop. This thesis was intended to provide a rigorous game theoretic approach in which check blocks can be integrated efficiently in the control loop. The approach considers delay constraints of the control loop, effectiveness of the blocks, and uncertainty in the number of signals through a two-stage stochastic Stackelberg model. The solutions are evaluated through extensive numerical analysis and comparison with several other mathematical approaches as well as numeric simulation using Simulink and implementation on a real world CPS testbed.

The results demonstrate the superiority of the stochastic approach when compared to a deterministic mathematical programming model, a greedy heuristic, and a random method. Other in depth analysis like a variant of two stage stochastic model with stochastic effectiveness, a piece-wise linear model, and an extended model with larger number of targets and attacks was also performed to cement the findings of the thesis. Further developments of this thesis includes to: (1) assess the model results under multiple type of adversaries (2) put precedence constraint on the way the blocks can be assigned (3) implement more complex attack functions and check blocks to assess the performance of two-SSP model (4) study how to integrate the models with algorithms that protect the CPS against coordinated attacks (e.g. replay, covert, false data injection).

The Cyber-Physical System (CPS) is a promising paradigm for the design of current and future engineered systems and is expected to make an important impact on our interactions with the real world. But the success of this new era of automation and engineering depends a lot on our ability to protect the CPS against cyber attacks. This thesis would help to orchestrate some new approaches

against cyber attacks and also help future researchers to find more efficient and effective ways of cyber security.

APPENDIX SECTION

APPENDIX A: Sample CPLEX Code

```
/*********************************************
* OPL 12.7.1.0 Model
* Author:  Shafi Ahad
* Creation Date:  Aug 13, 2017 at 5:29:50 PM
Title:  Stochastic Two-SSP Model with High Utility and High
Effectiveness
*********************************************/

float Beffectiveness[1..3][1..3];
float Bcapacity[1..6]=[40,40,40,40,40,40];
int BN[1..3][1..6]=[[7,7,7,7,7,7],[10,10, 10, 10, 10, 10],
[11,11,11,11,11,11]];
float BU[1..6][1..4]=[[-200, 0, 200, 0],[-100, 0, 100, 0],
[-100, 0, 100, 0],[-100, 0, 100, 0],[-100, 0, 100, 0],[-100,
0, 100, 0]];
float BTs[1..3]=[2, 1, 1.5];
float tempU[1..6];
float Btemp[1..6];
float Bassignment[1..3][1..6]=[[0.5714, 0, 0, 0, 0, 0],
[0, 1, 1, 1, 1, 1],
[0.5714, 0.625, 0.625, 0.625, 0.625, 0.625]];
float Bprobability[1..3]=[0.2,0.3,0.5];
float Bz[1..3][1..3][1..6];
float BTa[1..3][1..3]=
```

```
[[1.5,0.8,1.2],[2,1,1.5], [2.2, 1.2,1.8]];
main
{ var time0=new Date();
var source = new IloOplModelSource
("Stch_New_W-Assignment.mod");
var cplex = new IloCplex();
var def = new IloOplModelDefinition(source);

  for(var k=1;k<=5;k++)
{

var eff = new IloOplOutputFile
  ("Effectiveness-Stch_HU-HE_WA"+k+".txt");
var uty = new IloOplOutputFile
  ("Utility-Stch_HU-HE_WA"+k+".txt");
var ass = new IloOplOutputFile
  ("Assignment-Stch_HU-HE_WA"+k+".txt");
var prob = new IloOplOutputFile
  ("Probability-Stch_HU-HE_WA"+k+".txt");
var wor = new IloOplOutputFile
  ("Worstcase-Stch_HU-HE_WA"+k+".txt");
var Z_ste = new IloOplOutputFile
  ("Additional_Assignment-Stch_HU-HE_WA"+k+".txt");

  var opl = new IloOplModel(def,cplex);
var data2= new IloOplDataElements();

data2.effectiveness=thisOplModel.Beffectiveness;
data2.capacity=thisOplModel.Bcapacity;
```

```
data2.N=thisOplModel.BN;
data2.U=thisOplModel.BU;
data2.Ts=thisOplModel.BTs;
data2.assignment=thisOplModel.Bassignment;
data2.probability=thisOplModel.Bprobability;
data2.Ta=thisOplModel.BTa;
data2.temp=thisOplModel.Btemp;

for (var i=1; i<=3; i++)
for(var j=1; j<=3; j++)
while(1)
{ data2.effectiveness[i][j]=Opl.rand()/9999999999999999;
if(data2.effectiveness[i][j]>0.7)
break;
}
var totU=700
var tempU

  for(var i=1;i<=1;i++)
{ while(1)
{ data2.temp[i]=Opl.rand(totU);
if(data2.temp[i]>=300)
if(data2.temp[i]<=450)
break;

  }
  totU=totU-data2.temp[i];

  }
```

```
  for(var i=2;i<=6;i++)
data2.temp[i]=totU/5;

  for(var i=1;i<=6;i++)
{
  data2.U[i][1]=-data2.temp[i];
data2.U[i][3]=data2.temp[i];
}
  opl.addDataSource(data2);

  opl.generate();
if (cplex.solve())
{ writeln("OBJ = ",cplex.getObjValue());
writeln("Effectiveness= ", data2.effectiveness);
writeln(opl.assignment);

  for(var i=1;i<=3;i++)
{
  for(var j=1;j<=3;j++)
eff.write(data2.effectiveness[i][j]);
eff.write(" ");
eff.writeln();

  }
  for(var i=1;i<=6;i++)
{
  for(var j=1;j<=4;j++)
uty.write(data2.U[i][j]);
uty.write(" ");
```

```
uty.writeln(); }
for(var i=1;i<=6;i++)
{ for(var j=1;j<=3;j++)
ass.write(opl.assignment[j][i]);
ass.write(" ");}
ass.writeln();

  }
  for(var i=1;i<=6;i++)
{
  for(var j=1;j<=3;j++)
prob.write(opl.x[i][j]);
prob.write(" ");
prob.writeln();

  }
  wor.writeln(cplex.getObjValue());
Z_ste.writeln(opl.z);
writeln("Z_ste= ",opl.z);

  } else
{ writeln("No solution");
}
  } var time1=new Date();
writeln("Solve Time 1= ", time1-time0);
}
```

APPENDIX B: Two Way ANOVA results of section 8.2

General Linear Model: VSS versus Utility, Effectiveness

Analysis of Variance

Table 0.1: Summary of ANOVA

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Utility	2	26192	13096.0	50.80	0.000
Effectiveness	2	9206	4603.0	17.86	0.000
Utility*Effectiveness	4	2788	696.9	2.70	0.046
Error	36	9280	257.8		
Total	44	47466			
Model Summary					
S	R-sq	R-sq(adj)	R-sq(pred)		
16.0555	80.45%	76.10%	69.45%		

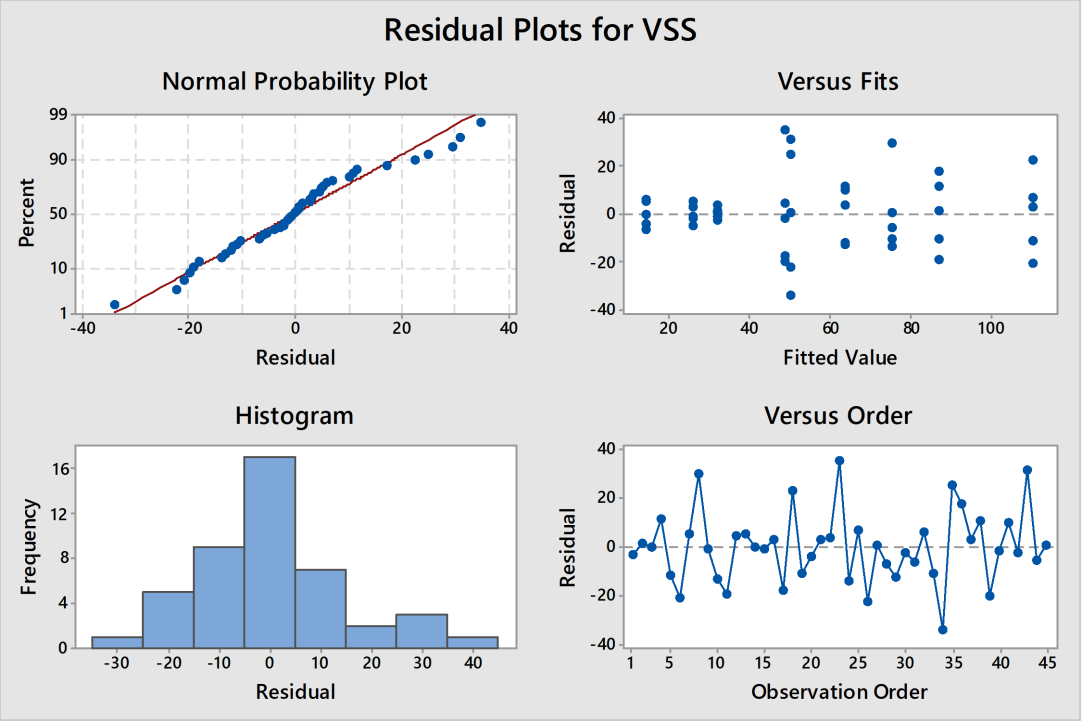

Figure 0.1: Residual plots of VSS from ANOVA.

APPENDIX C: How to Linearize "Max" function in MIP

The model presented in chapter 5 is non-linear because of equation-5.3. The equation was relaxed to $x_c = \max_{s \in S} E_s^a n_{st}$ to reduce the complexity of the model. But the "max" function is also not linear but can be linearized to form a MILP. The linearization process is described step-by-step below [49]-

Let's consider a constraint,
$y = max\{x_1, x_2, x_3, ..., x_n\}$ for continuous variables $x_1, x_2, x_3, ..., x_n$

- The upper and lower bound of the variables must have to be known

  $L_i \leq x_i \leq U_i$ [1.i]

- Introduce binary variables $d_1, d_2, ...d_n$

  $d_i = 1$ if x_i is the maximum value, 0 otherwise

- MIP formulation

  $L_i \leq x_i \leq U_i$ [1.i]

  $y \geq x_i$ [2.i]

  $y \leq x_i + (U_{max} - L_i)(1 - d_i)$ [3.i]

  $\sum_i d_i = 1$ [4]

```
// include the library code:

#include <LiquidCrystal.h>
#include <max6675.h>

  // initialize the library by associating any needed
LCD interface pin with the arduino pin number
it is connected to

const int rs = 7, en = 8, d4 = 9, d5 = 10, d6 = 11, d7 = 12;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int sensorPin=A0;
int sensorPin2=A2;
int sensorPin5=A5;

float sensorValue=0;
float pressure=0;
float voltValue=0;
float tempVolt=0;
float x=0;

int ktcSO = 5;
int ktcCS = 3;
int ktcCLK = 4;

MAX6675 ktc(ktcCLK, ktcCS, ktcSO);
```

```
  void setup()
{
// set up the LCD's number of columns and rows:
Serial.begin(9600);

  delay(500);
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.print("Volt");

  }
  void loop()
{
//set the cursor to column 0, line 1
//(note:  line 1 is the second row, since counting begins with
0):
//lcd.setCursor(0, 1);
// print the number of seconds since reset:
//lcd.print(millis() / 1000);

  sensorValue=analogRead(sensorPin)*(5.0/1023.0);
pressure=(sensorValue-0.50)*(200.0/4.0)+14.70;
lcd.setCursor(5,0);
lcd.print("P(psi)");
lcd.setCursor(5, 1);
lcd.print(pressure,2);

  voltValue=analogRead(sensorPin2)*(12.0/1023);
Serial.print("Voltage = ");
```

```
Serial.println(voltValue,4);
lcd.setCursor(0, 1);
lcd.print(voltValue,2);

  tempVolt=(analogRead(sensorPin5)*5.0/1023.0-1.34)/0.005;

  Serial.print("Deg C = ");
Serial.print(ktc.readCelsius());
Serial.print("Deg F = ");
Serial.println(ktc.readFahrenheit());
Serial.print("Pressure= ");
Serial.println(pressure,3);
Serial.println("Temp Voltage= ");
Serial.print(tempVolt);

  //delay(1500);

  lcd.setCursor(11, 0);
lcd.print("Temp");
lcd.setCursor(11, 1);
lcd.print(tempVolt,2);

  delay(1000);
}
```

REFERENCES

[1] (2018) The prisoner's dilemma: Solution and implications. [Online]. Available: https://sites.google.com/site/gametheorybasics/contact

[2] N. S. Foundation". (1999) Cyber-Physical Systems (CPS). Available at . [Online]. Available: https://www.nsf.gov/pubs/2014/nsf14542/nsf14542.htm

[3] V. Gunes, S. Peter, T. Givargis, and F. Vahid, "A survey on concepts, applications, and challenges in cyber-physical systems." *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS*, vol. 8, no. 12, pp. 4242 – 4268, n.d. [Online]. Available: http://libproxy.txstate.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edswsc&AN=000348048600001&site=eds-live&scope=site

[4] S. Morgan. (2015) Cybersecurity market reaches $75 billion in 2015;expected to reach $170 billion by 2020. [Online]. Available: https://www.forbes.com/sites/stevemorgan/2015/12/20/cybersecurity%E2%80%8B-%E2%80%8Bmarket-reaches-75-billion-in-2015%E2%80%8B%E2%80%8B-%E2%80%8Bexpected-to-reach-170-billion-by-2020/#72f091cb30d6

[5] Statista. (2017) Size of the cyber security market worldwide, from 2017 to 2022. [Online]. Available: https://www.statista.com/statistics/595182/worldwide-security-as-a-service-market-size/

[6] D. of Homeland Security. (2016) Cyber Physical Systems Security. Available at https://www.dhs.gov/science-and-technology/csd-cpssec. [Online]. Available: https://www.dhs.gov/science-and-technology/csd-cpssec

[7] H. Ye, "Security protection technology of cyber-physical systems." *International Journal of Security and its Applications*, vol. 9, no. 2, pp. 159–168, 2015. [Online]. Available: http://libproxy.txstate.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-84924505936&site=eds-live&scope=site

[8] M. Guirguis, A. Tahsini, K. Siddique, C. Novoa, J. Moore, C. Julien, and N. Dunstatter, "Bloc: A game-theoretic approach to orchestrate CPS against cyber attacks," in *Proceedings of the IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2018.

[9] H. Jin, *Handbook of Research on Developments and Trends in Wireless Sensor Networks: From Principle to Practice: From Principle to Practice*. IGI Global, 2010.

[10] D. Senie and P. Ferguson, "Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing," *Network*, 1998.

[11] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry, "Challenges for securing cyber physical systems," vol. 5, 2009.

[12] Y. Liu, P. Ning, and M. Reiter, "False Data Injection Attacks against State Estimation in Electric Power Grids," in *the 18th ACM Conference on Computer and Communications Security*, Chicago, IL, November 2009.

[13] Y. Mo and B. Sinopoli, "False Data Injection Attacks in Control Systems," in *Proceedings of the 1st workshop on Secure Control Systems*, 2010, pp. 1–6.

[14] N. Trcka, M. Moulin, S. Bopardikar, and A. Speranzon, "A Formal Verification Approach to Revealing Stealth Attacks on Networked Control Systems," in *Proceedings of the 3rd International Conference on High Confidence Networked Systems*, Chicago, IL, April 2014.

[15] A. Tiwari, B. Dutertre, D. Jovanović, T. de Candia, P. Lincoln, J. Rushby, D. Sadigh, and S. Seshia, "Safety Envelope for Security," in *Proceedings of the 3rd international conference on High confidence networked systems*. ACM, 2014, pp. 85–94.

[16] Y. Wang, Z. Xu, J. Zhang, L. Xu, H. Wang, and G. Gu, "SRID: State Relation Based Intrusion Detection for False Data Injection Attacks in SCADA," in *Computer Security-ESORICS*. Springer, 2014, pp. 401–418.

[17] R. Bobba, K. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. Overbye, "Detecting False Data Injection Attacks on DC State Estimation," in *The First Workshop on Secure Control Systems, CPS Week*, 2010.

[18] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "A Survey of Recent Results in Networked Control Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[19] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of Control and Estimation Over Lossy Networks," *IEEE*, vol. 95, no. 1, p. 163, 2007.

[20] S. Amin, A. Cárdenas, and S. Sastry, "Safe and Secure Networked Control Systems under Denial-of-Service Attacks," *Hybrid Systems: Computation and Control*, pp. 31–45, 2009.

[21] M. Wilhelm, I. Martinovic, J. Schmitt, and V. Lenders, "Short Paper: Reactive Jamming in Wireless Networks: How Realistic is the Threat?" in *Proceedings of the fourth ACM conference on Wireless network security*, 2011.

[22] K. Pelechrinis, M. Iliofotou, and V. Krishnamurthy, "Denial of Service Attacks in Wireless Networks: The Case of Jammers," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, 2011.

[23] V. Nguyen, M. Guirguis and G. Atia, "A Unifying Approach for the Identification of Application-driven Stealthy Attacks on Mobile CPS," in *Proceedings of the 52nd Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 2014.

[24] P. K. Mishra, "Analysis of mitm attack in secure simple pairing." *Journal of Global Research in Computer Science*, 2015. [Online]. Available: https://www.rroij.com/open-access/analysis-of-mitm-attack-in-secure-simple-pairing-42-45.php?aid=38069

[25] R. Myerson, "Game theory: Analysis of conflictharvard univ," *Press, Cambridge*, 1991.

[26] Y. Wadhawan and C. Neuman, "Defending cyber-physical attacks on oil pipeline systems: A game-theoretic approach," in *Proceedings of the 1st International Workshop on AI for Privacy and Security*. ACM, 2016, p. 7.

[27] Z. Xu and Q. Zhu, "A game-theoretic approach to secure control of communication-based train control systems under jamming attacks," in *Proceedings of the 1st International Workshop on Safe Control of Connected and Autonomous Vehicles*. ACM, 2017, pp. 27–34.

[28] H. Fang, L. Xu, and K.-K. R. Choo, "Stackelberg game based relay selection for physical layer security and energy efficiency enhancement in cognitive radio networks," *Applied Mathematics and Computation*, vol. 296, no. Supplement C, pp. 153 – 167, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0096300316306208

[29] T. Spyridopoulos, G. Karanikas, T. Tryfonas, and G. Oikonomou, "A game theoretic defence framework against dos/ddos cyber attacks," *Computers and Security*, vol. 38, no. Supplement C, pp. 39 – 50, 2013, cybercrime in the Digital Economy. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016740481300059X

[30] Z. Ismail, J. Leneutre, D. Bateman, and L. Chen, "A game-theoretical model for security risk management of interdependent ict and electrical infrastructures," in *High Assurance Systems Engineering (HASE), 2015 IEEE 16th International Symposium on*. IEEE, 2015, pp. 101–109.

[31] K. Chung, C. A. Kamhoua, K. A. Kwiat, Z. T. Kalbarczyk, and R. K. Iyer, "Game theory with learning for cyber security monitoring." *[2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE), High Assurance Systems Engineering (HASE), 2016 IEEE 17th International Symposium on, High-Assurance Systems Engineering, 2005. HASE 2005. Ninth IEEE International Symposium on]*, p. 1, 2016. [Online]. Available: http://libproxy.txstate.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.7423125&site=eds-live&scope=site

[32] D. Dentcheva, A. Ruszczynski, and A. Shapiro, "Lectures on stochastic programming: modeling and theory," *The society for industrial and applied mathematics and the mathematical programming society, Philadelphia, USA*, 2009.

[33] A. Shapiro and A. Philpott, "A tutorial on stochastic programming," *Manuscript. Available at www2. isye. gatech. edu/ashapiro/publications. html*, vol. 17, 2007.

[34] C. Novoa, K. Siddique, M. Guirguis, and A. Tahsini, "A game-theoretic two-stage stochastic programing model to protect CPS against attacks," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*. IEEE, 2018, pp. 15–22.

[35] A. P. Giddings, R. L. Rardin, and R. Uzsoy, "Statistical optimum estimation techniques for combinatorial optimization problems: a review and critique," *Journal of Heuristics*, vol. 20, no. 3, pp. 329–358, 2014.

[36] T. H. Bhuiyan, A. K. Nandi, H. Medal, and M. Halappanavar, "Minimizing expected maximum risk from cyber-attacks with probabilistic attack success," in *2016 IEEE Symposium on Technologies for Homeland Security (HST)*, May 2016.

[37] F. Yan, "Two-stage nash equilibrium problems," *ProQuest Dissertations and Theses Global*, 2007.

[38] D. McNulty. (2018) The basics of game theory. [Online]. Available: https://www.investopedia.com/articles/financial-theory/08/game-theory-basics.asp

[39] Game theory. [Online]. Available: https://www.investopedia.com/terms/g/gametheory.asp

[40] M. S. Alvard, "The ultimatum game, fairness, and cooperation among big game hunters." 2004. [Online]. Available: http://psycnet.apa.org/record/2005-01445-014

[41] G. Owen, *Game Theory*. Bingley: Emerald Group Publishing, 1995.

[42] T. S. Ferguson. Game theory, part ii. [Online]. Available: https://www.math.ucla.edu/~tom/math167.html

[43] H. Von Stackelberg, *Market structure and equilibrium*. Springer Science & Business Media, 2010.

[44] M. Padberg and G. Rinaldi, "A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems," *SIAM review*, vol. 33, no. 1, pp. 60–100, 1991.

[45] J. E. Mitchell, "Branch-and-cut algorithms for combinatorial optimization problems," *Handbook of applied optimization*, pp. 65–77, 2002.

[46] R. L. Rardin and R. L. Rardin, *Optimization in operations research*. Prentice Hall, 2016.

[47] Arduino introduction. [Online]. Available: https://www.arduino.cc/en/Guide/Introduction

[48] E. Dolan and J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, pp. 201–213, 2002.

[49] (2009) MIP formulations and linearizations. [Online]. Available: https://www.artelys.com/uploads/pdfs/Xpress/mipformref-1.pdf