Model Based Impact Analysis and Security Measure Allocation for Control Systems

JEZDIMIR MILOŠEVIĆ

Licentiate Thesis
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden 2018

KTH Royal Institute of Technology
School of Electrical Engineering
and Computer Science
Department of Automatic Control
SE-100 44 Stockholm
SWEDEN

TRITA-EECS-AVL-2018:15
ISBN: 978-91-7729-698-0

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av licentiatexamen i elektro- och systemteknik fredagen den 23 mars 2018 klockan 10.00 i L1, Kungliga Tekniska högskolan, Drottning Kristinas väg 30, Stockholm.

Tryck: Universitetsservice US AB

Abstract

Improvement of cyber-security of industrial control systems is of utmost importance for our society. It has been recognized that many security vulnerabilities can be found in these systems, which if exploited may lead to dire consequences. For instance, successful cyber-attacks against industrial control systems may cause loss of electricity, lead to shortage of drinkable water, or disrupt oil and gas production.

Deploying security measures to protect industrial control systems may be costly. Thus, it is expected that we would not be able to prevent all the security vulnerabilities that we find within the systems. In this thesis, we consider two problems related to this issue. The first one is how to determine which combinations of vulnerabilities are the most critical to be prevented. An important part of this classification is estimating the impact of cyber-attacks conducted using these vulnerabilities, which is the first major problem considered in the thesis. The budget for deploying security measures can then be focused on preventing the most critical combinations of vulnerabilities that are found. How to do this in an optimal way once the number of vulnerabilities and measures is large is the second major problem considered.

As our first contribution, we outline a framework for estimating the attack impact in industrial control systems. Here, we consider industrial control systems that have both control and monitoring tasks. For industrial control systems with control tasks, we propose a framework to estimate the impact of several attack strategies. We prove that the estimation of the impact of all possible strategies is reducible to solving a set of convex minimization problems. The solvers for convex minimization problems are well known, so the exact value of the attack impact can be obtained easily. For industrial control systems with monitoring tasks, we analyze the impact of a bias injection attack strategy. We prove that the attack impact can be obtained as the solution of a quadratically constrained quadratic program, for which the exact solution can be found efficiently. We also introduce a lower bound of the attack impact in terms of the number of compromised sensors. The theoretical findings are illustrated in numerical examples.

As our second contribution, we propose a flexible modeling framework for allocating security measures. Our framework is suitable for dynamical models of industrial control systems, and can be used in cases when the number of vulnerabilities and measures is large. The advantages of our framework are the following. Firstly, the framework includes an algorithm for efficiently finding the most dangerous vulnerabilities in the system. Secondly, the problem of eliminating these vulnerabilities can provably be casted as a minimization of a linear function subject to a submodular constraint. This implies that the suboptimal solution of the problem, with guaranteed performance, can be found using a fast greedy algorithm. The applicability of the framework is demonstrated through simulations on an industrial control system used for regulating temperature within a building.

Keywords: Cyber-Security, Cyber-Attacks, Cyber-Physical Systems, Networked Control Systems, Model Based Impact Analysis

Sammanfattning

Att säkerställa och förbättra cybersäkerheten hos industriella styrsystem är av stor vikt för samhällssäkerheten. Det är känt att det förekommer sårbarheter hos den här typen av system, som om de utnyttjas kan leda till allvarliga konsekvenser. Till exempel kan cyberattacker mot industriella styrsystem leda till storskaliga strömavbrott, dricksvattenbrist och störningar i olje- och gasproduktion. Av dessa skäl har de här problemen fått stor uppmärksamhet inom såväl forskningen som hos myndigheter och industrin.

Den här avhandlingen motiveras främst av att de åtgärder som kan vidtas för att skydda industriella styrsystem ofta är kostsamma. I många fall är det därför inte möjligt att åtgärda alla sårbarheter som kan hittas hos systemen. Man behöver därför ta hänsyn till två problem. Till att börja med, hur man kan identifiera vilka sårbarheter, eller kombinationer av dessa, som det är mest kritiskt att förhindra angrepp mot. En viktig del av denna analys är att beräkna effekten av sådana angrepp. Detta problem behandlas i avhandlingens första del. Nästa problem är att fördela en budget för säkerhetsåtgärder för att förebygga de mest kritiska sårbarheterna, vilket behandlas i avhandlingens andra del.

Avhandlingens första bidrag är ett ramverk för att bedöma effekten av angrepp mot industriella styrsystem. Här betraktar vi industriella styrsystem som både har till uppgift att reglera och att övervaka. För system med regleruppgifter föreslår vi ett ramverk för att uppskatta effekterna av flera olika angreppsstrategier. Vi visar att beräkningen av effekten av samtliga strategier kan reduceras till att lösa en uppsättning konvexa minimeringsproblem. Eftersom lösningsalgoritmer för sådana problem är välkända så kan exakta värden enkelt beräknas. För industriella styrsystem med övervakningsuppgifter analyserar vi effekten av angrepp i form av injektion av bias. Vi bevisar att denna effekt kan fås som lösningen till ett så kallat kvadratiskt programmeringsproblem med kvadratiska bivillkor, som går att lösa exakt på ett effektivt sätt. Vi demonstrerar resultaten i numeriska exempel.

Avhandlingens andra bidrag är en flexibel modelleringsmetod för resursfördelning av säkerhetsåtgärder. Metoden vi föreslår är lämpad för dynamiska modeller av industriella styrsystem och löser resursfördelningsproblemet när antalet sårbarbeter och åtgärder är stort. Den föreslagna metoden kan effektivt identifiera de farligaste sårbarheterna i systemet. Sedan visar vi hur problemet med att eliminera dessa sårbarheter kan formuleras som en minimering av en linjär funktion med submodulära bivillkor. Detta innebär att man med garanterad prestanda kan hitta en subotimal lösning med hjälp av en snabb så kallad girig algorithm. Vi demonstrerar att våra metoder går att tillämpa genom simuleringar av ett industriellt temperaturregleringssystem.

Acknowledgements

The path towards earning a PhD degree is full of obstacles. I would never have reached this far without all the support I have had. First and foremost, I would like to express my sincere appreciation towards my main supervisor Professor Henrik Sandberg. Thank you for your support, inspiration, patience, guidance, and encouragement. I am also grateful to my co-supervisor Professor Karl Henrik Johansson, for his support, help, and wise advices.

Special thanks to Takashi Tanaka, David Umsonst, André Teixeira, Mathias Müller, Farhad Farokhi, and Crisitian Rojas for the collaborations we have had so far. I learned a lot by working with you. I am thankful to Professor Mikael Johansson, David Umsonst, Michelle Chong, Mathias Müller, Mladen Čičić, Martin Biel, and Emma Tegling for their feedback on the thesis. I appreciate the administrative support and help from Anneli Ström, Hanna Holmquist, Felicia Gustafsson, Silvia Cardenas, Gerd Franzon, Karin Karlson, and Margreth Hellberg. I am also immensely grateful to my current and former colleagues at the Department of Automatic Control and from CERCES project for making PhD life enjoyable.

This work has been financially supported by the Swedish Civil Contingencies Agency through the CERCES project, and the Swedish Research Council. Their support is greatly acknowledged.

Finally, I would like to express my appreciation to my parents Verica and Milan, and to my sister Milana for their love and unconditional support through whole my life. I dedicate this thesis to you.

Jezdimir Milošević

Contents

Notation

$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	Set of real n–dimensional vectors
$\mathbb{R}^n_+$	Set of real n–dimensional vectors with positive elements
$\mathbb{R}^{n\times m}$	Set of real $(n \times m)$–dimensional matrices
$\mathbb{C}$	Set of complex numbers
$\mathbb{C}^n$	Set of complex n–dimensional vectors
$\mathbb{Z}$	Set of integers
$\mathbf{I}_n$	$(n \times n)$–dimensional identity matrix
$\mathbf{0}_{n\times m}$	$(n \times m)$–dimensional matrix with all elements equal to 0
$\mathrm{diag}(a_1, ..., a_n)$	$(n \times n)$–dimensional diagonal matrix with the elements $a_1, ..., a_n$ on the diagonal
$A \succeq 0$	Positive semi-definite matrix A
$A \succ 0$	Positive definite matrix A
$\mathrm{null}(A)$	Null-space of matrix A
$\dim(U)$	Dimension of the vector space U
$\mathrm{card}(x)$	Cardinality of the vector x
$\mathrm{Tr}\{.\}$	Trace operator
$\mathbb{E}\{.\}$	Expected value of a random variable
$\mathbb{P}(.)$	Probability measure
$2^{\mathcal{V}}$	Set of all subsets of the set $\mathcal{V}$, also known as power set
$\lvert\mathcal{V}\rvert$	Number of elements in the set $\mathcal{V}$

Chapter 1

Introduction

Industrial control systems (ICSs) are vital for the proper functioning of our society. These systems are, for instance, used to operate industrial facilities, power plants, or water distribution networks. In the last two decades, ICSs have become highly digitalized, which enabled more efficient and more reliable operation of these systems. However, the digitalization has negative aspects as well.

Documented cyber-attacks, along with numerous studies, have demonstrated that ICSs have become exposed to cyber-attacks. Given the nature of physical processes operated by ICSs, successfully conducted cyber-attacks can lead to human casualties, may cause considerable economic losses, and endanger the environment. For instance, the cyber-attack against the Maroochy shire's sewage control system lead to an environmental hazard [1, 2], the Stuxnet malware was used to sabotage an uranium enrichment facility in Iran [3–5], while a cyber-attack against the Ukranian power grid companies left thousands of households without electricity [6]. Therefore, improving ICSs cyber-security is of extreme importance.

Unfortunately, securing an ICS has proved to be more complex and more costly than securing an ordinary information technology (IT) system. Thus, a crucial step before implementing any security strategy for these systems is to conduct risk assessment [7–9]. In this assessment, security vulnerabilities within the system are first identified, and the complexity of exploiting these vulnerabilities together with the resulting impact, are estimated. Once the assessment is completed, one can move to the risk response step, where one prioritizes where to invest the security budget based on the results of the risk assessment. In this thesis, our objective is to provide mathematical models and tools that can make the process of risk assessment and risk response for ICS systematic and cost efficient.

Before moving to a more detailed formulation of the problems addressed within the thesis, we motivate the importance of studying these problems in Section 1.1. We then formalize problems and outline our contributions in Section 1.2, and present the structure of the thesis in Section 1.3.

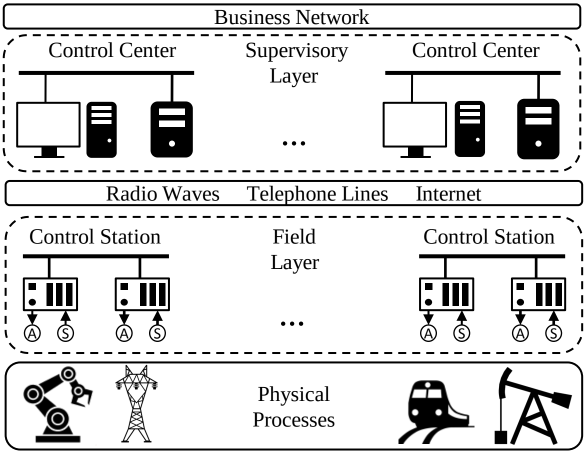


Figure 1.1: Typical architecture of an ICS. The field layer consists of sensors, actuators, and control devices. This layer is used to control the physical process. The supervisory layer models the environment of the control center. The operators in the control center can monitor the physical process, change configuration of controllers, and take manual control over some of the actuators if needed.

1.1 Motivation

ICSs are cyber-physical systems, as illustrated in Figure 1.1. These systems interact with the physical world through the sensors and actuators. The sensors collect measurements from the process, and send them to control devices. Based on these measurements, control devices calculate control actions, and execute them through the actuators. The control devices also share information with and execute commands from the control center. The cyber-physical nature of ICSs is the main reason why ICS security is so important to address. By exploiting some of the cyber vulnerabilities, an attacker can gain opportunities to manipulate sensors and actuators, and endanger the physical process being controlled.

Given the importance of physical processes operated by ICSs, it may be surprising that ICS security was not taken seriously into consideration in the past [10]. The main reason for this is that ICSs were isolated and much different from other IT systems. The hardware and software for ICSs were specially designed for these systems, and they were rarely connected to other networks [11]. However, the last two decades have witnessed rapid transformation of ICSs. These systems have become connected to other IT networks and directly or indirectly to the Internet [9]. Moreover, specialized technologies that were used for ICSs are now being more and more merged with ordinary IT technologies, and, in that way, inherited their vulnerabilities [8].

Unfortunately, these changes were not accompanied by applying appropriate security solutions, which resulted in a large number of security vulnerabilities within ICSs. For instance, communication protocols commonly used in ICSs are often lacking basic protection such as encryption and authentication [12], the local area network in the control center may be connected directly or indirectly to the Internet without adequate protection [8], or anti-virus software may not be installed or updated [7]. Besides cyber vulnerabilities, physical security vulnerabilities could also be present. For example, physical access control to some of the devices within the system may be missing [7].

In order to prevent cyber-attacks, different institutions and standardization organizations have issued recommendations for improving the cyber-security of ICSs [7–9, 13]. The overall suggestion is to implement a defense-in-depth strategy, which consists of multiple layers of security measures. Examples include implementing encryption and authentication of communication channels, segmentation of the control network, installing and maintaining anti-virus software, etc.

However, deployment of security measures in an ICS proves to be a very challenging task. In contrast to ordinary IT systems that have a typical life span of two to five years, ICSs are designed to last for decades. Thus, support for some of the equipment found within ICSs may not exist anymore [7]. These systems also have tight real time requirements, and stopping them in order to deploy or update security measures should be planned well in advance [7]. Furthermore, the equipment used in ICSs is in many cases resource constrained. Adding security measures such as encryption could require additional memory and computing resources, which may cause delays in the system and result in instability [14].

In summary, given the potentially large number of vulnerabilities within an ICS, and all the difficulties of implementing security measures, preventing all of the security vulnerabilities at once might not be possible. Thus, we should first prioritize what combinations of vulnerabilities are more dangerous than the others. One criterion used for prioritizing is the impact that may occur if a certain combination of vulnerabilities is exploited. How to estimate the impact of cyber-attacks is the first question we try to answer in this thesis. Once the high risk combinations of vulnerabilities are determined, the security measures should be selected such as to prevent these combinations. How to select the least expensive combination of the security measures is the second question we try to answer.

1.2 Problem Formulation and Contributions

In this thesis, we address two key problems. The first one is the problem of estimating attack impact. The second one is the problem of deploying security measures in a cost-efficient manner. In what follows, we introduce these problems in more details, and summarize our contributions.

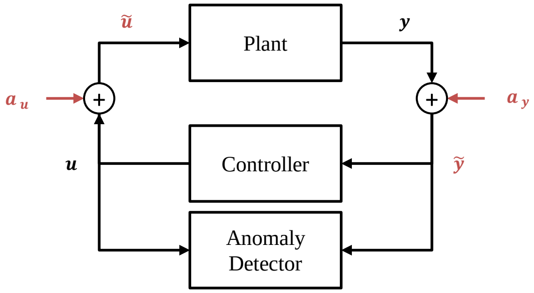


Figure 1.2: A schematic of a control system under attack. The control input u is corrupted by the attack signal a_u, so the signal $\tilde{u}$ is applied to the process instead of u. Similarly, the sensor measurements y are subject to the attack signal a_y, so the controller and anomaly detector receive the false measurement signal $\tilde{y}$.

1.2.1 Problem 1 – Model Based Impact Analysis

The attack impact can be estimated by modeling an ICS and then simulating possible attack strategies. Given that cyber-attacks against ICSs may endanger the physical world, a natural approach is to use a physical model of the system to estimate the attack impact.

A physical part of an ICS under attack is illustrated in Figure 1.2. The plant is modeled as a linear time invariant system of the form

$$\begin{aligned} x(k+1) &= Ax(k) + B\tilde{u}(k) \\ y(k) &= Cx(k) \end{aligned}$$

where $x(k)$ is the state of the system, which contains the physical states such as pressures, flows, or temperatures, $\tilde{u}(k)$ is the control signal applied to the process, and $y(k)$ are the sensor measurements collected from the process.

At certain point, the attacker exploits a group of security vulnerabilities, and gains control over some of the sensors and the actuators. Due to attacks, the signal $\tilde{u}(k)$ is different from the true control signal $u(k)$. Similarly, the measurement signal $\tilde{y}(k)$ received by the controller and the anomaly detector is different from the original one $y(k)$. As illustrated in Figure 1.2, the signals $\tilde{u}(k)$ and $\tilde{y}(k)$ are given by

$$\tilde{u}(k) = u(k) + a_u(k) \qquad\qquad \tilde{y}(k) = y(k) + a_y(k)$$

where $a_u(k)$ and $a_y(k)$ are the attack signals dependent on the sensors and actuators that the attacker controls. These signals could be designed based on the model knowledge, with the purpose to achieve some malicious goal. For instance, this goal can be to increase the pressure in a pipeline above some maximal permitted threshold, or to overflow a tank containing hazardous materials.

However, the behavior of the system under attack usually differs from the one during the normal operation. In case that the measurements $\tilde{y}(k)$ are significantly different from those during the normal operation, an alarm is triggered by the anomaly detector. We can then start some safety procedure to mitigate the attack.

Therefore, from the defender's perspective, we are interested in determining if the attacker can conduct the attack with the high impact while staying undetected by the anomaly detector. Given that we do not know what the attacker's goals are, one way to reason about the possible impact would be to assume an attack strategy, and then to find the worst case impact under this strategy. Based on the previous discussion, the problem of estimating attack impact can be formulated as the following optimization problem.

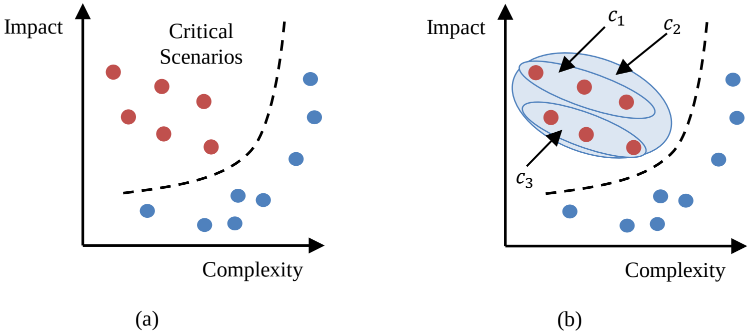
Problem 1. (Attack Impact Estimation Problem)

$$\begin{aligned} \underset{a_u, a_y}{\text{maximize}} \quad & \text{Impact} \\ \text{subject to} \quad & \text{The trajectory of the system satisfies physical laws} \\ & \text{The attack signals } a_u \text{ and } a_y \text{ follow the attack strategy} \\ & \text{The attack is not detected by the anomaly detector} \end{aligned}$$

This problem represents a constrained maximization problem, and it is a non-convex problem in general. Solving these problems is challenging, since the algorithms for finding the exact solution of optimization problems of this type are rarely available. For this reason, it is beneficial to reduce the problem of estimating impact to some of the standard maximization problems that are known to be solvable. We propose several such optimization problems in Chapters 3 and 4.

In Chapter 3, we consider the problem of estimating the impact for an ICS that has a control task. We propose a framework to estimate the impact of denial of service, sign alternation, rerouting, replay, false data injection, and bias injection attack strategies. Our framework is valid for stateless, moving window, cumulative sum, and multivariate exponentially weighted moving average detectors. The attack impact measure is adopted to be the infinity norm of the critical states after a fixed number of time steps. For this measure of the attack impact, we prove that the impact for all of the attack strategies is reducible to the problem of solving a set of convex minimization problems. The solvers for convex minimization problems are well known, so the exact value of the attack impact can be obtained easily. We illustrate how the models of the attacks can be used to estimate the impact on a model of a chemical process.

In Chapter 4, we address the problem of estimating impact of cyber-attacks targeting monitoring systems. The goal of monitoring systems is to estimate the state of the system based on available sensor measurements. For this purpose, a Kalman filter is used. We consider a chi-squared test in this chapter, and a bias injection attack strategy. The attack impact is measured through the steady state mean square estimation error. For this impact metric, we prove that the attack impact can be obtained as a solution of quadratically constrained quadratic program, for which the exact solution can be found efficiently. We also introduce a



(a)

(b)

Figure 1.3: Illustration of the security measure allocation problem. (a) The first step: finding the most critical attack scenarios, which are those that are not complex to conduct and have a large impact. (b) The second step: selecting the least expensive subset of security measures such as to prevent the critical scenarios.

lower bound of the attack impact in terms of the number of compromised sensors. We illustrate theoretical findings on a model of a quadruple tank process.

1.2.2 Problem 2 – Allocating Security Measures

The second major problem we address is how to allocate security measures within an ICS in a cost-efficient manner. Let the set of security vulnerabilities be denoted with $\mathcal{V}$. The elements of the set $\mathcal{V}$ can model a communication link without protection, or insufficient physical security of some components. By exploiting one or more of these vulnerabilities, the attacker can gain control over some of the sensors and actuators and endanger the physical world. Naturally, the high risk combinations of vulnerabilities are those that are not complex to be exploited, and can lead to a large impact, as shown in Figure 1.3 (a).

The vulnerabilities can be prevented by deploying security measures from the set $\mathcal{M}$. The elements of this set can model encryption of communication link, or installation and maintenance of anti-virus software. In order to deploy some of these measures, we need to pay a certain cost. Thus, the goal is to select the least expensive subset $\mathcal{M}_d$ of security measures that prevents all of the high risk vulnerability combinations, as illustrated in Figure 1.3 (b). The security measure allocation problem can then be reduced to the following combinatorial optimization problem.

Problem 2. (Security Measure Allocation)

$$\begin{aligned} \underset{\mathcal{M}_d \subseteq \mathcal{M}}{\text{minimize}} \quad & \text{Cost} \\ \text{subject to} \quad & \text{The subsets of } \mathcal{V} \text{ with high risk are all prevented} \end{aligned}$$

The two main difficulties with solving this problem are the following. Firstly, finding the high risk combinations of vulnerabilities presents an issue. Mainly, the number of combinations of vulnerabilities is equal to the number of subsets of the vulnerability set $\mathcal{V}$. Hence, simply going through all the combinations of vulnerabilities and deciding whether it is with high risk, or not, is not feasible when the cardinality of this set is large. Secondly, even if we find the set of high risk vulnerabilities, the security measure allocation problem is a constrained combinatorial minimization problem. These problems are NP hard in general, thus algorithms that return the optimal solution in polynomial time do not exist except in some special cases.

The aforementioned issues are addressed in Chapter 5. In particular, we propose a flexible modeling framework for allocating security measures suitable for dynamical models of ICSs. Our framework targets allocating security measures once the number of security vulnerabilities and security measures is large. To find the combination of vulnerabilities with the high risk, we adopt a tree representation of the power set, which is used for the purpose of systematic search. We propose the breadth first search algorithm that exploits properties of the risk model and security measures allocation problem to remove branches of this tree. Using this algorithm, the execution time of the search can potentially be reduced considerably.

Once the high risk combinations of vulnerabilities are found, the security measure allocation problem reduces to an integer linear program. Given that integer linear programs are NP hard in general, we show how to reduce the problem to a minimization of a linear function subject to a submodular constraint. In that case, a polynomial-time greedy heuristic can be used to provide a solution with known performance bound. Furthermore, we prove that the set of critical scenarios returned by the aforementioned breadth-first search algorithm provides the best performance guarantees for the greedy algorithm. Applicability of our framework is illustrated on a model of the control system used to regulate temperature within a building.

1.3 Structure of the Thesis

In what follows, we briefly outline the context of each chapter of this thesis. Where relevant, we state peer reviewed scientific contributions on which the chapter is based.

Chapter 2: Background

In this chapter, we introduce the typical architecture of ICSs, summarize some of the cyber-attacks conducted so far, explain the risk management program, and provide a literature review.

Chapter 3: Estimating Attack Impact in Control Systems

In this chapter, we propose several attack strategies that can be used to estimate potential impact of cyber-attacks. The focus is placed on ICSs with control tasks. The chapter is based on the publication:

J. Milošević, D. Umsonst, H. Sandberg, and K. H. Johansson, "Quantifying the impact of cyber-attack strategies for control systems equipped with an anomaly detector," *in Proceedings of the European Control Conference (ECC), 2018. To appear.*

Chapter 4: Estimating Attack Impact in Monitoring Systems

In this chapter, we address the problem of estimating the attack impact in monitoring systems. In particular, we observe a Kalman filter equipped with the chi-squared detector under bias injection attacks. We derive the worst case attack impact, and a lower bound on the attack impact. The chapter is based on the publication:

J. Milošević, T. Tanaka, H. Sandberg, and K. H. Johansson, " Analysis and mitigation of bias injection attacks against Kalman filter," *in Proceedings of the IFAC World Congress*, 2017.

Chapter 5: Allocating Security Measures in ICSs

In this chapter, we study the problem of optimally allocating security measures for dynamical control systems. The chapter is based on the publications:

J. Milošević, A. Teixeira, T. Tanaka, H. Sandberg, and K. H. Johansson, "Security measure allocation for industrial control systems: exploiting systematic search techniques and submodularity," *submitted to International Journal of Robust and Nonlinear Control.*

J. Milošević, T. Tanaka, H. Sandberg, and K. H. Johansson, "Exploiting submodularity in security measure allocation for industrial control systems," *in Proceedings of the 1st ACM Workshop on the Internet of Safe Things (SafeThings'17), 2017.*

Chapter 7: Conclusion and Future Work

In this chapter, we summarize the results presented in the thesis, and outline directions for future work.

Author Contributions and Other Publications

In the aforementioned peer-reviewed articles, the author of the thesis has formulated and solved most of the problems, and written the papers. The results of the coauthors are clearly indicated throughout the thesis.

The following publications in which the author of the thesis participated are not covered in the thesis.

F. Farokhi, J. Milošević, and H. Sandberg, "Optimal state estimation with measurements corrupted by Laplace noise," *in Proceedings of the 55th IEEE Conference on Decision and Control (CDC).*

M. I. Müller, J. Milošević, H. Sandberg, and C. R. Rojas, "A risk-theoretic approach to $\mathcal{H}_2$-optimal control under covert attacks," *to be submitted.*

Chapter 2

Background

In this chapter, we introduce the background that can help the reader to follow the remainder of the thesis. The chapter covers the following topics. In Section 2.1, we explain the typical architecture of an ICS. In Section 2.2, we summarize different types of cyber-attacks. In Section 2.3, we introduce attacks against ICSs conducted so far. In Section 2.4, we describe the risk management program, and explain how are the problems we consider in the thesis related to this program. In Section 2.5, we provide a literature review.

2.1 Architecture of an ICS

An ICS consists of several layers, as shown in Figure 2.1 [15]. The interaction with the physical process is achieved through the field layer. The components of this layer are well known within the control community, and include sensors, actuators, and control devices (e.g., programmable logic controllers, remote terminal units, or intelligent electronic devices [16]). Sensors collect measurements from physical processes (e.g., pressures and temperatures), and send these measurements to a control device. The control device then calculates appropriate control signals based on the sensor measurements and the control algorithm, and sends the signals to actuators (e.g., pumps and valves) for execution. Other roles of control devices include sharing of information such as measurements, control signals, and alarms with the supervisory layer, and executing commands received from the supervisory layer. The communication between the field and the supervisory layer is established in different ways, for instance through telephone lines, radio waves, or the Internet.

The supervisory layer is responsible for monitoring the physical process, collecting and storing process information, centralized alarming, and trend analyses. If required, operators from the control center can take control over some of the components in the field layer. Nowadays, the supervisory layer is usually connected to both the field layer, as well as to other IT systems (e.g., manufacturing execution systems and enterprise resource planning systems).

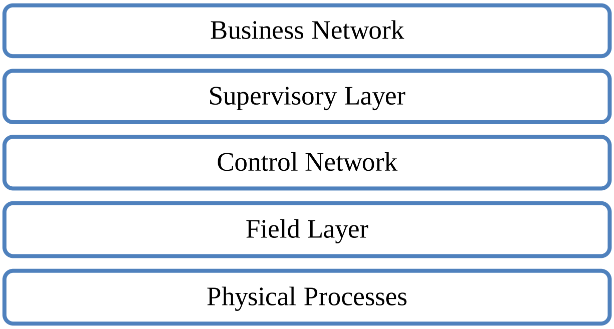


Figure 2.1: Layered architecture of an ICS.

The components of the supervisory layer mostly include IT equipment with different purposes. The most important components of this layer include the engineering workstation and the human machine interface [7]. An engineering workstation is a computer that is used for configuration, maintenance, and diagnostic of control system applications and control system equipment [7]. For instance, control algorithms executed on the control devices in the field layer are designed and stored on the engineering workstation. A human machine interface is a hardware and/or software that is used to display process data in a convenient form, and to enable an operator to manually interact with the control devices from the field layer [7]. Through a human machine interface, an operator can monitor the physical process, modify the control objective, and manually control actuators.

2.2 Types of Cyber-Attacks

The three key properties of IT security are confidentiality, integrity, and availability of the information and services [17]. Confidentiality means that data can be accessed only by authorized users, integrity guarantees that unauthorized change of the information is not possible, and availability implies that data and services are available on a user request.

The attacks against these three properties are illustrated in Figure 2.2. The attacks against confidentiality are called *disclosure attacks*. In these attacks, the attacker gains unauthorized access to the data or service (Figure 2.2 (a)). The attacks against integrity are known as *false data injection attacks*. Due to false data injection attacks, the user may use the false information thinking it is true (Figure 2.2 (b)). Finally, the attacks against availability are known as *denial of service attacks*. These attacks block the user from gaining access to data or service (Figure 2.2 (c)).

From ICS security perspective, false data injection attacks and denial of service attacks are the most important to consider. The reason is that these attack strate-

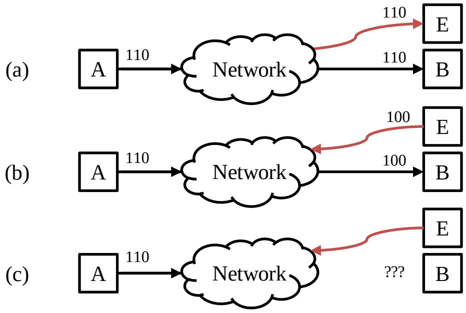


Figure 2.2: Illustration of different types of cyber-attacks: (a) disclosure attacks; (b) false data injection attacks; (c) denial of service attacks. The users are denoted with A and B, and the attacker with E.

gies can endanger the physical world. For instance, false data injection attacks against sensor measurements, can lead to the wrong control signal being calculated and sent to the actuators for execution. Similarly, if the attacker blocks control signals from reaching the actuators, the system may become unstable.

2.3 Documented Attacks Against ICSs

Examples of some of the cyber-attacks against ICSs are provided next.

Example 1. (The Maroochy Water Services Breach [1, 2]) The attacks against the ICS operating sewerage system occurred in the year 2010, in Australia. The attacks lasted approximately for two months. During that time, the system was attacked at least 46 times. Around one million liters of untreated sewage was released into a stormwater drain, and the contaminated water flowed waterways, parks, and grounds of a local hotel. The attack caused death of marine life, and unbearable smell spread over the area.

The attacks were conducted by Vitek Boden, an engineer involved in the installation of the radio equipment for the attacked ICS. He conducted all the attacks from his car, by using stolen radio equipment connected to a computer with specialized software. The attacker was also familiar with the architecture of the system, and knew the vulnerabilities of the system. Using these resources, Mr. Boden was able to inject malicious radio commands, disable alarms in the field stations, and manage to alter the configuration for the field stations. □

Example 2. (Stuxnet [3–5]) Stuxnet is a computer worm specially designed for attacks against the ICS operating an uranium enrichment facility in Iran. Since its discovery in the year 2010, it has attracted considerable attention from the media, industry, and research community.

The Stuxnet attack can be divided into two phases. In the first phase, Stuxnet infiltrated the computer network within the nuclear facility through a USB drive. It then spreads itself through the network until it finds the computers within the control center. Once the communication between the control center and the control devices controlling the uranium enriching centrifuges was established, Stuxnet downloaded itself on the controllers.

In the second phase, Stuxent started collecting sensor measurements of normal system operation. After it has collected enough measurements, Stuxnet started sending the harmful control signals to the centrifuges. At the same time, Stuxnet was masking these actions by sending the previously recorded measurements of the normal operation to operators in the control center. In this way, the operators believed that the process was operating normally, while the harmful controlled signals were damaging the centrifuges. □

Example 3. (The Cyber Attack on the Ukrainian Power Grid [6]) The attack occurred in December 2015, in Ukraine. Three different electricity distribution companies were targets of the attack, which lead to 225,000 customers being without electrical energy.

As in the previously described Stuxnet attack, this attack can also be divided into two stages. In the first stage, infected e-mails were sent to the employees in the IT and administrative network of the targeted electricity companies. The e-mails had attached documents containing the malware BlackEnergy. Workers were deceived in opening these documents and installing Black Energy, after which the virus enabled the attacker to establish a connection with the company network. The attacker was then able to explore the company's network, and eventually localize and gain access to the control center. Once there, the attacker started gathering information about the control center environment, and prepared the second stage of the attack.

In the second stage, the attacker disabled operators from controlling the engineering workstation. The attacker additionally blocked communication between the workstation and the field stations, in order to ensure that remote commands could not be sent to the field stations in case the operators recovered control over the engineering workstations. The attacker then used human machine interface devices to issue malicious control commands to circuit breakers, and in that way cause the blackout of the system. □

2.4 Risk Management Program

The risk management program serves for determining the risk, implementing a security strategy to respond to the risk in a cost efficient way, and monitoring how the risk evolves over time. This program can be divided into four steps: risk framing, risk assessment, risk response, and risk monitoring, as shown in Figure 2.3 [18].

The risk framing step is a prerequisite for the remaining steps. It serves to define a strategy for conducting risk assessment, responding to risk, and monitoring risk.

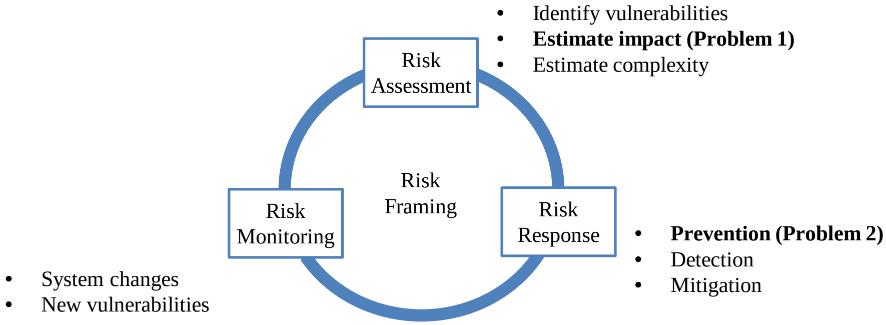


Figure 2.3: Risk management cycle. The risk framing step serves to define a strategy for conducting the remaining three steps. The risk assessment step consists of identifying vulnerabilities within the system, estimating the complexity of exploiting these vulnerabilities, and estimating impact that may occur in that case. Risk response consists of prevention, detection, and mitigation actions, while the risk monitoring step evaluates the risk over time.

The risk assessment step serves to determine a risk. Actions of this step include identifying vulnerabilities within the system, estimating the impact that may occur if the vulnerabilities are exploited, and determining the complexity of exploiting vulnerabilities. For instance, the security vulnerabilities within the system can be identified based on the lists of vulnerabilities commonly found within an ICS [7]. The attack complexity can be estimated based on a security expert knowledge [18–20]. An attack impact can be estimated by modeling an ICS and then simulating possible attack strategies [21]. As we stated earlier, the problem of estimating impact is the first major problem we consider in the thesis.

Once the risk assessment is completed and the risk is determined, one can move to the risk response step. This step serves to deploy an effective security strategy. The first step of risk response is to try to prevent an attacker from exploiting security vulnerabilities by deploying appropriate security measures [22, 23]. As we discussed earlier, it is expected that we would not be able to deploy the security measures that prevent all the vulnerabilities. Thus, the focus should be on preventing the high risk vulnerabilities previously determined in the risk assessment step, and this is the second major problem addressed in the thesis. The other two steps of risk response include detection and mitigation actions. In case the attacker finds a way to exploit some of the vulnerabilities, the anomaly detectors should be deployed to detect the attacks. For instance, the attack can be detected by observing anomalies in the physical behavior of the system [24]. Finally, the way to mitigate the attack once it is detected should be determined. One way to achieve this would be to reconfigure the system such that redundant secured components are used to control the process [25, 26].

The last step of the risk management program serves to monitor risk over time. The purpose of this component is to evaluate the effectiveness of the implemented security strategy, and to determine how the risk is affected by the system changes.

2.5 Cyber-Security of Control Systems

In the past, studies in control theory usually tried to answer how the operation of an ICS is affected by randomly occurring faults [24], noise [27], packet drops [28], or model uncertainty [29]. However, cyber-attacks pose new and fundamentally different challenges [15].

In contrast to noise and faults that are random in nature and without any malicious objective to fulfill, cyber-attacks can be designed based on system knowledge with the purpose of achieving some malicious objective. Moreover, an attack might be conducted at several components at the same time in a coordinated way. Another difference is that disturbances are often assumed to be bounded, while attack signals are dependent on an attacker's choice, and can take arbitrary value. Finally, it was shown that anomaly detectors developed to detect faults prove to be ineffective against carefully designed cyber-attacks. One of the first studies that theoretically proved this was [30]. The authors considered a system used for monitoring power grid equipped with a bad data detector. They proved that an attacker is able to degrade the quality of the state estimate without being detected by the bad data detector.

For the aforementioned reasons, it is not surprising that cyber-security of control systems has become a well established research topic [31, 32]. In what follows, we review some of the major problems considered so far. Literature directly related to the problems we consider in the thesis is reviewed in more detail in Chapters 3–5.

Fundamental Limitations

Fundamental limitations that the cyber-attacks impose have been derived in several theoretical studies. Fawzi *et al.* [33] considered estimation problem in presence of attacks, and established the impossibility of accurately reconstructing the state of the system in the presence of an intelligent cyber-attacker. Pasqualetti *et al.* derived fundamental limitations for detecting cyber-attacks [34]. The authors proved that the attacks hidden in the zero dynamics of the system cannot be distinguished from the normal system operation. Sundaram *et al.* [35] showed that if the connectivity of the communication graph of a networked control system is $2f$ or less, then f malicious nodes may conduct a coordinated attack such that it is not possible to correctly calculate an arbitrary function of all node values.

Experimental Studies

Besides theoretical studies, experimental studies also verified that novel solutions are required for malicious cyber-attacks. Amin *et al.* [36] considered an attacker with the model knowledge and the ability to manipulate some of the measurements and the control signals. The experiment conducted on the ICS controlling a water canal network demonstrated that such an attacker is able to lead to water pilfering from the canal system, without being detected by an anomaly detector.

Teixeira *et al.* [37] considered cyber-attacks against an ICS monitoring a power grid. The experiments were conducted on a realistic energy management systems software. The authors validated experimentally that it is possible to design undetectable cyber-attacks that degrade the estimation quality.

Modeling of Cyber-Attacks

A physical model of an ICS can be used for modeling cyber-attack strategies, and estimating the possible attack impact. Models of attacks in which the attacker avoids being detected by an anomaly detection mechanisms has attracted most of the attention. Examples of these strategies include zero-dynamics attacks [34, 38], covert attacks [39], and replay attacks [40]. However, more simple attack strategies such as denial of service [15, 41], rerouting [42, 43], and sign alternation attacks [44, 45] have also been considered. In order to be able to model and analyze the attacks in a unified way, a modeling framework for ICSs under attack was proposed by Teixeira *et al.* [38].

Security Index

An interesting approach for analyzing ICS vulnerability is by using the so-called security index. The security index was introduced in [46], to characterize the vulnerability of individual sensors monitoring a power network. The security index of a sensor s is equal to the number of sensors that needs to be attacked in addition to sensor s, in order to conduct an undetectable attack such as the one proposed in [30]. What is challenging with this approach is to estimate security index in an efficient and optimal manner once the number of sensors in the network is large. In fact, it was shown in [47] that calculating the security index is an NP hard problem in general. However, if the network topology satisfies certain assumptions, the security index can be efficiently calculated using l_1 relaxation [48], or by solving a min-cut problem [47, 49]. Extensions of the security index to dynamical control systems were proposed in [50–52].

Novel Anomaly Detection Methods

The problem of designing anomaly detectors capable of detecting cyber-attacks has attracted a lot of attention [40, 53–58]. Some of the approaches include the following. Mo *et al.* [40, 53] considered the problem of detecting replay attacks. They proposed the way of designing watermarking signal that reveals replay attacks, without considerably degrading the system performances. Teixeira *et al.* [54] considered the problem of detecting zero dynamics attacks. They showed that the attacks can be revealed by modifying the system structure. Do *et al.* [55] derived a scheme that detects covert and zero dynamics attacks by modulating the control signals.

Game Theoretic Approach

The problem of control system security has also been approached using game theoretic tools [59–62]. Zhu and Basar introduce two zero sum games to model the problem of cyber-security of control systems [59]. The first zero sum game is used to model the problem of designing a robust control strategy for physical layer, while the second one models the problem of designing a security policy. Ugrinovksi *et al.* [62] considered the problem of denial of service attacks against a control system, and modeled it as a zero sum game. Applications for power networks were considered in [60, 61]. In particular, Amin *et al.* [60] proposed a game theoretic model of electricity theft, while Shelar and Amin [61] modeled the problem of protecting a electricity distribution network as a three stage defender-attacker-defender game.

Attack Resilient State Estimators

Besides detectors, novel types of attack-resilient estimators [33, 63–69], controllers [70–77], and consensus protocols [78–81] have been considered in the literature. The problem of designing attack-resilient state estimators proved to be the most interesting one so far.

Fawzi *et al.* [33] considered the problem of reconstructing the state of a noiseless linear system from the corrupted sensor measurements. The estimator minimizes the l_0 norm of the difference between the window of previous sensor measurements and the initial state, over all the combinations of sensors and initial states. The authors proved that this estimator is the optimal one, in the sense that if the state cannot be reconstructed by this estimator, then it cannot be reconstructed by any other. The main issues with this approach are that it does not take noise into consideration, finding the state estimate is an NP hard problem, and a delayed estimate is returned. Approaches that take noise into consideration are considered in [63, 69], a more computationally efficient method was proposed in [68], while the issue of delay was addressed in [64].

Chapter 3

Estimating Attack Impact in Control Systems

This chapter addresses Problem 1, which is on estimating impact of cyber-attacks. In particular, we propose a framework to estimate the impact of several cyber-attack strategies against a dynamical control system equipped with an anomaly detector. Denial of service, sign alternation, rerouting, replay, false data injection, and bias injection attack strategies are considered. The framework is applicable for stateless, moving window, cumulative sum, and multivariate exponentially weighted moving average anomaly detectors. As the attack impact measure, the infinity norm of critical states after a fixed number of time steps is adopted. For this measure and the aforementioned anomaly detectors, we prove that the attack impact for all of the attack strategies can be reduced to the problem of solving a set of convex minimization problems. Thus, the exact value of the attack impact can be obtained easily. We demonstrate how our modeling framework can be used for risk assessment on a numerical example.

3.1 Introduction

3.1.1 Literature Review

In this chapter, we are focused on the problem of estimating the impact of cyber-attacks in control systems equipped with various types of anomaly detectors. Aspects of this problem were earlier investigated in studies [82–84]. Cárdenas *et al.* [82] considered a control system equipped with a *cumulative sum* (CUSUM) anomaly detector, and proposed several attack strategies that can be used to quantify the attack impact. Ahmed *et al.* [83] investigated the performance of *stateless* and CUSUM anomaly detectors in presence of false data injection and zero alarm attacks. In order to compare different types of anomaly detectors, Urbina *et al.* [84] introduced a novel metric and an attack model that can be used for that purpose.

We identify two main directions in which these studies can be extended. Firstly, the aforementioned works mostly considered an attacker that is very resourceful. For instance, the attacker possesses full model knowledge, controls a considerable number of components within the system, and is able to inject arbitrary signals to sensors and actuators it controls. Simpler attack strategies, which are also more likely to happen, were not considered. Some of these strategies include denial of service [15, 41], rerouting [42, 43], sign alternation [44, 45], and replay [40] attacks. Secondly, it is often unclear from the literature how the worst case attack impact is calculated. The problem of estimating the attack impact usually represents a constrained maximization problem, and algorithms that return the *exact* solution of these problems are rarely available. However, in previous work [85], the authors introduced infinity norm of the states as a measure of impact, and formulated the problem of finding the attack impact as an optimization problem that yields the exact solution. Thus, we adopt this metric to quantify the impact of the attack strategies we consider in this chapter.

3.1.2 Contributions

The contributions of this chapter are as follows. Firstly, we propose a unified framework for quantifying the attack impact in control systems equipped with an anomaly detector. Our framework is flexible, and can be used to quantify the impact of both simple attack strategies such as denial of service, sign alternation, rerouting, replay, and bias injection, as well as more complex false data injection attacks with full model knowledge. Our analysis is valid for stateless, moving window, CUSUM, and the multivariate exponentially weighted moving average (MEWMA) detector. Secondly, for the impact measure introduced in [85], we prove that the impact for all attack strategies and all considered detectors can be obtained by solving a set of convex minimization problems (Propositions 1–4). This implies that the exact value of the attack impact can easily be obtained, since the algorithms for solving convex minimization problems are well known. Finally, we illustrate on a numerical example of a chemical process how our framework can be used for risk assessment.

3.1.3 Organization

The remainder of the chapter is organized as follows. In Section 3.2, we introduce a model of the control system under attack, and models for anomaly detectors. In Section 3.3, we introduce criterion based on which we characterize the attack impact and some technical results. In Section 3.4, we formulate several attack strategies, and prove that the impact for these strategies can be obtained by solving a set of convex optimization programs. In Section 3.5, we illustrate on a simulation study how the introduced framework can be used for assessing potential impact of cyber-attacks. In Section 3.6, we briefly summarize the results presented in the chapter.

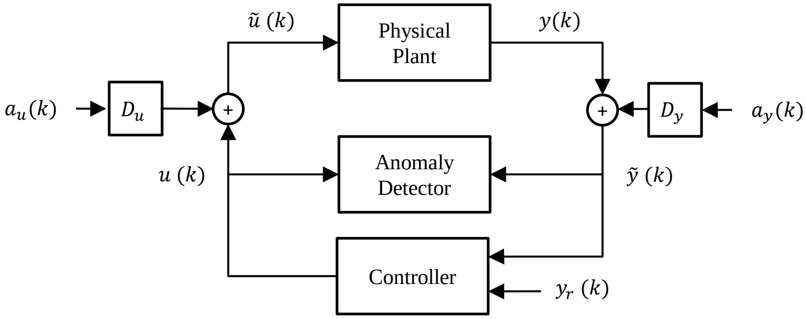


Figure 3.1: A schematic of a control system under attack. The sensor measurements $y(k)$ are subject to the attack signal $D_y a_y(k)$, so the controller and anomaly detector receive corrupted version of the signal $\tilde{y}(k)$. Similarly, the control input $u(k)$ is corrupted by the attack signal $D_u a_u(k)$, so the signal $\tilde{u}(k)$ is applied to the process instead of $u(k)$.

3.2 Model Setup

We consider the modeling framework introduced in [38], where the control system was modeled as an interconnection of the plant, the controller, the anomaly detector, and the attacker, as shown in Figure 3.1. In what follows, we introduce detailed models of each of the blocks. We then combine these blocks into the extended model of the plant.

3.2.1 Plant and Feedback Controller

The physical plant is modeled as

$$\mathcal{P}\colon \begin{cases} x(k+1) = A_p x(k) + B_p \tilde{u}(k) \\ \quad\; y(k) = C_p x(k) \end{cases} \tag{3.1}$$

where $x(k) \in \mathbb{R}^{n_x}$ is the state of the plant at time step k, $y(k) \in \mathbb{R}^{n_y}$ is the vector of sensor measurements, and $\tilde{u}(k) \in \mathbb{R}^{n_u}$ is the control input applied to the process. We denote the set of sensor with $\mathcal{S} = \{1, 2, \dots, n_y\}$, and the set of actuators with $\mathcal{A} = \{1, 2, \dots, n_u\}$.

The plant is controlled with a feedback controller of the form

$$\mathcal{F}\colon \begin{cases} z(k+1) = A_f z(k) + B_f \tilde{y}(k) + K_f y_r(k) \\ \quad\; u(k) = C_f z(k) + D_f \tilde{y}(k) + E_f y_r(k) \end{cases} \tag{3.2}$$

where $z(k) \in \mathbb{R}^{n_z}$ is the state of the controller, $\tilde{y}(k) \in \mathbb{R}^{n_y}$ is the vector of sensor measurements used by the controller to calculate the control signal, $u(k) \in \mathbb{R}^{n_u}$ is the control signal calculated by the controller, and $y_r(k) \in \mathbb{R}^{n_{y_r}}$ is the bounded reference signal. In particular, we assume

$$-\delta_{y_r} \leq y_r(k) \leq \delta_{y_r} \tag{3.3}$$

where $\delta_{y_r} \in \mathbb{R}_+^{n_{y_r}}$. The assumption is that the controller is designed so that stability and acceptable performances are achieved under the nominal (un-attacked) behavior.

3.2.2 Anomaly Detector

In the absence of attacks, the signals $\tilde{y}(k)$ and $\tilde{u}(k)$ are equal to $y(k)$ and $u(k)$, respectively. However, because of an attack or a fault in the system, these values may differ. In order to detect these anomalies, an anomaly detector is used. The first step of the detection procedure is to calculate the so called residual signal. We consider a residual-generating filter of the form

$$\mathcal{D}\colon \begin{cases} s(k+1) = A_d s(k) + B_d u(k) + K_d \tilde{y}(k) \\ \quad\;\; r(k) = C_d s(k) + D_d u(k) + E_d \tilde{y}(k) \end{cases} \tag{3.4}$$

where $s(k) \in \mathbb{R}^{n_s}$ is the state of the filter, and $r(k) \in \mathbb{R}^{n_r}$ is the residual signal evaluated to detect potential anomalies. We assume that the filter is designed such that the following properties are satisfied:

1. the value of the residual $r(k)$ converges asymptotically to zero in absence of anomalies;

2. the residual $r(k)$ is sensitive to attacks and anomalies, and in case when $\tilde{u}(k) \neq u(k)$ and/or $\tilde{y}(k) \neq y(k)$, $r(k)$ is different from zero except in pathological cases such as zero dynamic attacks (see [38]).

These are standard assumptions adopted from the fault-diagnosis literature [86].

The second step of the detection procedure is to process the residual signal $r(k)$ to obtain a security metric $S(k+1)$. When this metric exceeds a certain threshold $\delta_r > 0$, an alarm is raised. How $S(k+1)$ is determined depends on the detector used. In this chapter, we are focused on the following four anomaly detectors.

Stateless Detector

A stateless detector is defined as

$$S(k+1) = ||Q_r r(k)||_p^2$$

where $Q_r \in \mathbb{R}^{n_r \times n_r}$ represents a scaling matrix, and $||(.)||_p$ represents the p-norm. The common values for p used throughout the literature are 2 or ∞.

Moving Window Detector

Instead of using only a single value of $r(k)$ in decision making, the moving window detector uses weighted sum of the last $N_w > 1$ values of $r(k)$. In other words, this detector is defined as

$$S(k+1) = \sum_{j=0}^{N_w - 1} w_j ||Q_r r(k-j)||_p^2$$

where w_j are non-negative weighting coefficients.

CUSUM Detector

The CUSUM detector is a stateful detector, which in its non-parametric form is defined as

$$S(k+1) = \max\{S(k) + ||Q_r r(k)||_p^2 - \delta, 0\}$$

where $\delta > 0$ is the forgetting factor of the CUSUM detector. The metric $S(k+1)$ is reset to zero once an alarm occurs, that is, when $S(k+1) > \delta_r$.

MEWMA Detector

The MEWMA detector is another stateful detector, which is defined as

$$\begin{aligned} \tilde{S}(k+1) &= \beta Q_r r(k) + (1-\beta)\tilde{S}(k) \\ S(k+1) &= \frac{2-\beta}{\beta} ||\tilde{S}(k+1)||_2^2 \end{aligned}$$

where $\beta \in (0, 1]$ is the forgetting factor of the detector. As for the CUSUM detector, $\tilde{S}(k+1)$ is reset to zero, if an alarm occurs.

3.2.3 Attack Model

By exploiting some security vulnerability, the attacker is able to manipulate the subsets of sensors $\mathcal{S}_a \subseteq \mathcal{S}$ and actuators $\mathcal{A}_a \subseteq \mathcal{A}$. The influence of attack on signals $y(k)$ and $u(k)$ is modeled as

$$\tilde{y}(k) = y(k) + D_y a_y(k) \qquad \tilde{u}(k) = u(k) + D_u a_u(k) \tag{3.5}$$

where $a_u(k) \in \mathbb{R}^{n_{a_u}}$ represents the attack against actuators, $a_y(k) \in \mathbb{R}^{n_{a_y}}$ represents the attack against sensors, and the matrices $D_u \in \mathbb{R}^{n_u \times n_{a_u}}$ and $D_y \in \mathbb{R}^{n_y \times n_{a_y}}$ model the influence of attacks on actuators and sensors, respectively. We remark that the matrices D_u and D_y depend on the sets of $\mathcal{S}_a$ and $\mathcal{A}_a$. If the attacker is able to manipulate the sensors measurements $\mathcal{S}_a = \{j_1, j_2, \ldots, j_{n_{a_y}}\}$, then the elements $(j_1, 1), (j_2, 2), \ldots, (j_{n_{a_y}}, n_{a_y})$ of the matrix D_y are equal to one, and the remaining elements are equal to zero. The matrix D_u is defined in an analogous way, as illustrated in the following example.

Example 4. Consider a system with four sensors $\mathcal{S} = \{1, 2, 3, 4\}$ and three actuators $\mathcal{A} = \{1, 2, 3\}$. Assume that the attacker controls the sensors two and three, and the first actuator. The sets $\mathcal{S}_a$ and $\mathcal{A}_a$ are then

$$\mathcal{S}_a = \{2, 3\} \qquad \mathcal{A}_a = \{1\}$$

and the matrices D_y and D_u are given by

$$D_y = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \qquad D_u = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

3.2.4 Extended System Model

In order to formulate some of the attack strategies in a more compact form, we introduce the augmented vectors

$$x_e(k) = [x^T(k)\ z^T(k)\ s^T(k)]^T \qquad a(k) = [a_u^T(k)\ a_y^T(k)]^T$$

which represent the extended state of the system and extended attack vector, respectively. We denote the dimension of the vector $x_e(k)$ by $n_e = n_x + n_z + n_s$, and the dimension of the vector $a(k)$ by $n_a = n_{a_u} + n_{a_y}$. By combining (3.1), (3.2), (3.4), and (3.5), the dynamics of the system under the attack can be expressed as

$$\begin{aligned} x_e(k+1) &= A_e x_e(k) + B_e a(k) + K_e y_r(k) \\ r(k) &= C_e x_e(k) + D_e a(k) + E_e y_r(k) \end{aligned} \tag{3.6}$$

where

$$A_e = \begin{bmatrix} A_p + B_p D_f C_p & B_p C_f & \mathbf{0}_{n_x \times n_s} \\ B_f C_p & A_f & \mathbf{0}_{n_z \times n_s} \\ (K_d + B_d D_f) C_p & B_d C_f & A_d \end{bmatrix} \qquad B_e = \begin{bmatrix} B_p D_u & B_p D_f D_y \\ \mathbf{0}_{n_z \times n_{a_u}} & B_f D_y \\ \mathbf{0}_{n_s \times n_{a_u}} & (B_d D_f + K_d) D_y \end{bmatrix}$$

$$C_e = \begin{bmatrix} (D_d D_f + E_d) C_p & D_d C_f & C_d \end{bmatrix} \qquad D_e = \begin{bmatrix} \mathbf{0}_{n_r \times n_{a_u}} & (D_d D_f + E_d) D_y \end{bmatrix}$$

$$K_e = \begin{bmatrix} B_p E_f \\ K_f \\ B_d E_f \end{bmatrix} \qquad E_e = D_d E_f.$$

3.3 Quantifying Attack Impact

The main goal of this chapter is to estimate the impact that can occur once the attacker exploits some security vulnerability. In order to do that, we first introduce the criterion based on which we characterize the impact of cyber-attacks. We then introduce the convex optimization framework that is used for estimating the impact of different attack strategies that we define in the next section.

3.3.1 Criterion for Characterizing the Attack Impact

In order to estimate the impact of possible attacks, we use the concept of critical states. Let $Q_c \in \mathbb{R}^{n_c \times n_e}$ be a matrix that maps the extended state vector to a subset of critical states

$$x_c(k) = Q_c x_e(k).$$

These critical states may model levels in tanks with hazardous materials that must not be overflown, or pressures that should not exceed some safety limit. From the perspective of the defender, we want to prevent the attacker in driving any of these states far from the steady state. Therefore, one way to estimate the impact would be to check if the attacker is able to drive the critical states far from the steady state during some time interval. For simplicity, we assume that the attack starts at $k = 0$, and we observe how far the attacker can drive system in the time interval $[0, N]$. The attack impact $I\big(\mathcal{S}_a, \mathcal{A}_a\big)$ measure can then be defined as

$$I\big(\mathcal{S}_a, \mathcal{A}_a\big) := ||x_c(N)||_\infty.$$

Besides driving the critical states as far as possible from the steady state, we are interested to check if the attack can stay undetected by an anomaly detector. The assumption is that if we are able to detect the attack, we can start safety procedures in order to prevent the attacker from causing a large damage to the system. Therefore, we also want to check if the attack can be conducted without triggering an alarm. Hence, we impose the constraints

$$S(k+1) \leq \delta_r \qquad\qquad k = 0, \ldots, N \tag{3.7}$$

where $S(k+1)$ is calculated by using one of the detectors we introduced in the previous section.

From (3.6), we see that the system has two input signals–the reference signal $y_r(k)$ and the attack signal $a(k)$. Thus, during the attack, the system trajectory depends on both of these signals. Given that the reference signal is often a constant signal, we adopt the following standing assumption.

Assumption 1. The reference signal is constant and equal to the reference prior to attack $y_r(k) = y_r$, $k = 0, 1, \ldots, N$. The system has reached steady state before the attack happens, that is

$$S(0) = 0 \qquad\qquad r(0) = 0 \qquad\qquad x_e(0) = Q_{ss} y_r$$

where $Q_{ss} \in \mathbb{R}^{n_e \times n_{y_r}}$ represents the steady state gain of the transfer function from the reference signal $y_r(k)$ to the extended state $x_e(k)$ of the system. □

In what follows, we are performing off-line analysis of the attack impact, so the exact value of reference signal from the interval (3.3) at the beginning of the attack is unknown to us. Thus, throughout the chapter we identify the worst possible value of the reference y_r when estimating the attack impact. Same holds for the attack signals.

3.3.2 Convex Approach for Calculating Attack Impact

The attacker can use different attack strategies in order to conduct an attack. In this chapter, we observe denial of service, rerouting, sign alternation, replay, false data injection, and bias injection attack strategies. We show that for all of these attack strategies and for all of the anomaly detectors we observe, the impact can be obtained by solving an optimization problem of the following form.

Problem 3. (Maximizing infinity norm subject to symmetric convex constraints)

$$\begin{aligned} \underset{d}{\text{maximize}} \quad & ||Td||_\infty \\ \text{subject to} \quad & f_i(d) \leq \delta_i \qquad i = 1, \dots, n_i \end{aligned}$$

where $d \in \mathbb{R}^{n_d}$ is the decision variable, T is a matrix from $\mathbb{R}^{n_c \times n_d}$, and the constraint functions $f_i(d) : \mathbb{R}^{n_d} \to \mathbb{R}$, $i = 1, \dots, n_i$, are symmetric and convex.

A convenient property of the problems of this type is that the optimal solution can be obtained by solving n_c convex minimization problems. Given that the algorithms that return the global solution of the convex minimization problems are well known, we are able to use these algorithms for finding the *exact* value of the attack impact. In the next lemma, we formally prove the aforementioned claim.

Lemma 1. Let I be the optimal value of Problem 3, and I' be the optimal value of the following set of convex minimization problems

$$\begin{aligned} \underset{l \in \{1, \dots, n_c\}}{\text{minimize}} \quad \underset{d}{\text{minimize}} \quad & -T(l,:)d \\ \text{subject to} \quad & f_i(d) \leq \delta_i \qquad i = 1, \dots, n_i. \end{aligned}$$

The equality $I = |I'|$ then holds.

Proof. Let d^* be an optimal solution of Problem 3, and let the optimal value of this problem be defined with

$$I = ||Td^*||_\infty = |T(l^*,:)d^*|.$$

Thus $-|T(l^*,:)d^*| \leq -T(l,:)d$ for every $l \in \{1, 2, \dots, n_c\}$, and for every d that satisfies the constraints. Given that the constraints are equivalent for both of the problems, it follows $-I \leq I'$. Assume that $-I < I'$. By the symmetry of the constraints, we have that both d^* and $-d^*$ are feasible points for both of the problems. However, that implies that either $T(l^*,:)d^*$ or $T(l^*,:)(-d^*)$ is less than 0. If we define

$$I'' = \min\{-T(l^*,:)d^*, T(l^*,:)d^*\}$$

then it follows $I'' = -I < I'$. This contradicts the assumption that I' is the optimal value of the problem stated in the lemma. Therefore, the only possibility is $-I = I'$, which concludes the proof. □

In order to reduce the attack strategies to the form of Problem 3, we use that the detector constraints are convex and symmetric under a certain condition, as stated in the following lemma[1].

Lemma 2. Assume that $S(0) = 0$. If the residuals can be expressed as $r(k) = T_r(k)d$, where $T_r(k) \in \mathbb{R}^{n_r \times n_d}$ and $d \in \mathbb{R}^{n_d}$, then the constraints

$$S(k+1) \leq \delta_r \qquad\qquad k = 0, \dots, N$$

represent convex and symmetric constraints in d for the stateless, moving window, CUSUM, and MEWMA detector.

Proof. We first show that the stateless detector is convex and symmetric in d. By using the definition of the stateless detector, we have

$$S(k+1) = ||Q_r r(k)||_p^2 = ||Q_r T_r(k)d||_p^2 \leq \delta_r.$$

Since every norm is symmetric and convex, and the square of a convex function is still convex, we see that the stateless detector represents convex and symmetric constraints in d.

Note that the moving window detector represents the nonnegative sum of at most N_w stateless detectors $||Q_r T_r(k)d||_p^2$, which we prove to be symmetric and convex. Thus, it follows that this constraint is symmetric and convex itself.

Using $r(k) = T_r(k)d$ in the definition of the CUSUM detector leads to

$$S(k+1) = \max\{S(k) + ||Q_r T_r(k)d||_p^2 - \delta, 0\}.$$

Due to the symmetry of $||Q_r T_r(k)d||_p^2$ in d we immediately see that $S(k+1)$ is also symmetric in d for every k. To show that $S(k)$ is convex we will use induction. The base step for $k = 0$ shows that $S(0) = 0$ is convex because it is constant. For the induction step we assume that $S(k)$ is convex and show that $S(k+1)$ is convex. First note that the max of two convex functions is also convex and that the sum of convex functions is also convex [87]. By assumption $S(k)$ is convex and $-\delta$ as well since it is constant. Here, $||Q_r r(k)||_p^2 = ||Q_r T_r(k)d||_p^2$ is convex in d which was already shown for the stateless detector case. Hence, $S(k) + ||Q_r r(k)||_p^2 - \delta$ is convex and therefore $S(k+1)$ is also convex in d.

For the MEWMA detector we rewrite $\tilde{S}(k)$ in terms of d

$$\tilde{S}(k) = \beta \sum_{i=0}^{k-1} (1-\beta)^{k-1-i} Q_r r(i) = \beta \sum_{i=0}^{k-1} (1-\beta)^{k-1-i} Q_r T_r(i) d$$

and we see that if $d \to -d$, $\tilde{S}(k) \to -\tilde{S}(k)$. Therefore, $S(k)$ represents a symmetric constraint in d due to the symmetry of the squared Euclidean norm. Since $\tilde{S}(k)$ is a linear transformation of d, it is a convex function. Hence, $S(k) = \frac{2-\beta}{\beta} ||\tilde{S}(k)||_2^2$ is also convex in d for all k. □

[1] The lemma is formulated and proved by David Umsonst, one of the coauthors of the paper based on which this chapter is written.

3.4 Attack Strategies

We now introduce the attack strategies, and prove that the problem of finding the attack impact can be reduced to the form of Problem 3 for all of the strategies.

3.4.1 Denial of Service Attacks

In this attack strategy, the attacker starts blocking some of the signals of the sensors and actuators from reaching their destination. This can be achieved by gaining unauthorized access to system and making physical damage to devices, but as well as by overflowing communication network with large amount of traffic, or jamming the network [15].

One possible way of modeling this type of attacks was suggested in [15, 41], where the control signals and measurements during the attack were modeled as

$$\tilde{u}(k) = \Lambda_u u(k) \qquad\qquad \tilde{y}(k) = \Lambda_y y(k) \tag{3.8}$$

where $\Lambda_u \in \mathbb{R}^{n_u \times n_u}$ and $\Lambda_y \in \mathbb{R}^{n_y \times n_y}$ are diagonal matrices defined as follows

$$\Lambda_u(i,i) = \begin{cases} 0 & i \in \mathcal{A}_a \\ 1 & i \notin \mathcal{A}_a \end{cases} \qquad\qquad \Lambda_y(i,i) = \begin{cases} 0 & i \in \mathcal{S}_a \\ 1 & i \notin \mathcal{S}_a \end{cases} . \tag{3.9}$$

By combining (3.1), (3.2), (3.4), and (3.8), the dynamics of the extended system under the denial of service attack can be expressed as

$$\begin{aligned} x_e(k+1) &= \tilde{A}_e x_e(k) + \tilde{B}_e y_r \\ r(k) &= \tilde{C}_e x_e(k) + \tilde{D}_e y_r \end{aligned} \tag{3.10}$$

where

$$\begin{aligned} \tilde{A}_e &= \begin{bmatrix} A_p + B_p\Lambda_u D_f \Lambda_y C_p & B_p \Lambda_u C_f & \mathbf{0}_{n_x \times n_s} \\ B_f \Lambda_y C_p & A_f & \mathbf{0}_{n_z \times n_s} \\ (B_d D_f + K_d)\Lambda_y C_p & B_d C_f & A_d \end{bmatrix} & \qquad \tilde{B}_e &= \begin{bmatrix} B_p \Lambda_u E_f \\ K_f \\ B_d E_f \end{bmatrix} \\ \tilde{C}_e &= \begin{bmatrix} (D_d D_f + E_d)\Lambda_y C_p & D_d C_f & C_d \end{bmatrix} & \qquad \tilde{D}_e &= D_d E_f . \end{aligned}$$

From (3.10), and by using the fact that $x_e(0) = Q_{ss} y_r$, the critical states after N steps and the residual signal after k steps can be expressed as

$$x_c(N) = Q_c x_e(N) := T_x y_r \qquad\qquad r(k) := T_r(k) y_r \tag{3.11}$$

where

$$T_x = Q_c \left(\tilde{A}_e^N Q_{ss} + \sum_{i=0}^{N-1} \tilde{A}_e^i \tilde{B}_e \right) \qquad T_r(k) = \tilde{C}_e \left(\tilde{A}_e^k Q_{ss} + \sum_{i=0}^{k-1} \tilde{A}_e^i \tilde{B}_e \right) + \tilde{D}_e .$$

Note that the evolution of the system under the denial of service attack is only dependent on the value of the reference signal y_r. Thus, what we need to investigate is if there exists an y_r inside of the operating region defined by (3.3), such that the denial of service attack strategy drives some of the critical states far from the steady state while remaining undetected at the same time. Therefore, the problem of finding the worst case impact $I\big(\mathcal{S}_a, \mathcal{A}_a\big)$ in the case of the denial of service attack strategy can be formulated as the following optimization problem.

Problem 4. (Estimating impact of denial of service attacks)

$$\begin{aligned}
\underset{y_r}{\text{maximize}} \quad & I\big(\mathcal{S}_a, \mathcal{A}_a\big) = ||T_x y_r||_\infty \\
\text{subject to} \quad & -\delta_{y_r} \leq y_r \leq \delta_{y_r} \\
& S(k+1) \leq \delta_r \qquad\quad k = 0, \ldots, N.
\end{aligned}$$

In what follows, we prove that Problem 4 can be reduced to the form of Problem 3.

Proposition 1. Problem 4 is an instance of Problem 3 for the stateless, moving window, CUSUM, and MEWMA detector.

Proof. The only decision variable in Problem 4 is y_r, so $d = y_r$. The objective functions are of the same form, thus we only need to prove that all the constraints of the problem are convex and symmetric. Let $f_k(d) = S(k+1) \leq \delta_r$, $k = 0, \ldots, N$. We know from (3.11) that $r(k) = T_r(k)d$ for every k, so it follows from Lemma 2 that $f_0(d), ..., f_N(d)$ represent convex and symmetric constraints in d. It remains to prove that the reference constraint is symmetric and convex in d. Let Q_{y_r} be the diagonal matrix from $\mathbb{R}^{n_{y_r} \times n_{y_r}}$ whose elements are defined by $Q_{y_r}(i,i) = 1/\delta_{y_r}(i)$. We can then represent the reference constraint (3.3) as

$$f_{y_r}(d) = ||Q_{y_r} d||_\infty \leq 1.$$

This constraint is a convex and symmetric constraint in d due to the infinity norm, which concludes the proof. □

3.4.2 Rerouting Attacks

In this attack strategy, the attacker permutes the values of the measurements/control signals under its control. As stated in [42, 43], the attacker can conduct this attack by physically re-wiring the sensor cables, or by modifying the senders address.

The control inputs and measurements during the rerouting attack can be expressed as

$$\tilde{u}(k) = \Lambda_u u(k) \qquad\qquad \tilde{y}(k) = \Lambda_y y(k)$$

where $\Lambda_u \in \mathbb{R}^{n_u \times n_u}$ and $\Lambda_y \in \mathbb{R}^{n_y \times n_y}$ are any permutation matrices that satisfy the following constraint

$$\Lambda_u(i,i) = 1 \;\; i \notin \mathcal{A}_a \qquad\qquad \Lambda_y(i,i) = 1 \;\; i \notin \mathcal{S}_a.$$

Note that the way we define $\tilde{u}(k)$ and $\tilde{y}(k)$ in this attack strategy is identical to the way we defined them for the denial of service attack strategy. The only difference is that Λ_u and Λ_y this time represent permutation matrices. Therefore, for the fixed permutation matrices Λ_u and Λ_y, the problem of finding the worst case impact of the rerouting attack strategy can be reduced to Problem 4.

Remark 1. The attacker can permute the sensor measurements in $|\mathcal{S}_a|!$ number of ways, and control signals in $|\mathcal{A}_a|!$ number of ways. Hence, the total number of possible choices for the permutation matrices Λ_u, Λ_y is equal to $|\mathcal{A}_a|!|\mathcal{S}_a|!$. Thus, finding the worst case attack impact for all possible rerouting attack strategies can be expensive for larger cardinalities of sets $\mathcal{S}_a$ and $\mathcal{A}_a$. The number of combinations can be reduced by selecting combinations that are more likely to happen. For instance, in order to avoid easy detection, the attacker would probably exchange two measurements or control signals that have the same nature.

3.4.3 Sign Alternation Attacks

In this attack strategy, the attacker simply flips the sign of the measurement and control signals under its control. Although simple, these attacks can turn negative feedback into positive, and in that way destabilize the system. Moreover, it was shown that in certain configurations with Kalman filter, this attack strategy leads to strictly stealthy attacks [44, 45]. The control signal and measurement signal during the attack are given by

$$\tilde{u}(k) = \Lambda_u u(k) \qquad\qquad \tilde{y}(k) = \Lambda_y y(k)$$

where $\Lambda_u \in \mathbb{R}^{n_u \times n_u}$ and $\Lambda_y \in \mathbb{R}^{n_y \times n_y}$ are in this case defined as

$$\Lambda_u(i,i) = \begin{cases} -1 & i \in \mathcal{A}_a \\ 1 & i \notin \mathcal{A}_a \end{cases} \qquad\qquad \Lambda_y(i,i) = \begin{cases} -1 & i \in \mathcal{S}_a \\ 1 & i \notin \mathcal{S}_a \end{cases} .$$

Note that the way we define $\tilde{u}(k)$ and $\tilde{y}(k)$ is again the same as in the case of denial of service attack strategy. Thus, the problem of estimating impact of this attack strategy can also be reduced to Problem 4.

3.4.4 Replay Attacks

This attack strategy is inspired by well known Stuxnet malware [3]. The attacker keeps steady sensor measurements at sensors under its control, while at the same time sends malicious control signals to actuators it controls. We consider two possible attack scenarios. In the first attack scenario, the attack signal sent to actuators is assumed to be a constant bias. This can model the case where the attacker sends high value control signal to actuators. In the second scenario, the attacker blocks the original control signals of reaching the actuators.

Replay Attacks with Additive Bias

The control signals and measurements during the attack can be modeled as

$$\tilde{u}(k) = u(k) + D_u a_u \qquad\qquad \tilde{y}(k) = \tilde{\Lambda}_y y(k) + \Lambda_y y(0) \tag{3.12}$$

where $a_u \in \mathbb{R}^{n_{a_u}}$ is the malicious control signal sent to actuators, and $\Lambda_y \in \mathbb{R}^{n_y \times n_y}$ and $\tilde{\Lambda}_y \in \mathbb{R}^{n_y \times n_y}$ are diagonal matrices defined as

$$\Lambda_y(i,i) = \begin{cases} 1 & i \in \mathcal{S}_a \\ 0 & i \notin \mathcal{S}_a \end{cases} \qquad\qquad \tilde{\Lambda}_y = \mathbf{I}_{n_y} - \Lambda_y .$$

From (3.1), (3.2), (3.4), (3.12), and

$$y(0) = [C_p \ \mathbf{0}_{n_y \times n_z} \ \mathbf{0}_{n_y \times n_s}] Q_{ss} y_r := Q^y_{ss} y_r$$

the system under the attack can be expressed in the form

$$\begin{aligned} x_e(k+1) &= \tilde{A}_e x_e(k) + \tilde{B}_e a_u + \tilde{K}_e y_r \\ r(k) &= \tilde{C}_e x_e(k) + \tilde{D}_e y_r \end{aligned}$$

where

$$\tilde{A}_e = \begin{bmatrix} A_p + B_p D_f \tilde{\Lambda}_y C_p & B_p C_f & \mathbf{0}_{n_x \times n_s} \\ B_f \tilde{\Lambda}_y C_p & A_f & \mathbf{0}_{n_z \times n_s} \\ (K_d + B_d D_f) \tilde{\Lambda}_y C_p & B_d C_f & A_d \end{bmatrix} \quad \tilde{B}_e = \begin{bmatrix} B_p D_u \\ \mathbf{0}_{n_z \times n_{a_u}} \\ \mathbf{0}_{n_s \times n_{a_u}} \end{bmatrix}$$

$$\tilde{C}_e = \begin{bmatrix} (D_d D_f + E_d) \tilde{\Lambda}_y C_p & D_d C_f & C_d \end{bmatrix} \qquad \tilde{D}_e = D_d E_f + (D_d D_f + E_d) \Lambda_y Q^y_{ss}$$

$$\tilde{K}_e = \begin{bmatrix} B_p (E_f + D_f \Lambda_y Q^y_{ss}) \\ K_f + B_f \Lambda_y Q^y_{ss} \\ B_d E_f + (B_d D_f + K_d) \Lambda_y Q^y_{ss} \end{bmatrix} .$$

The critical states after N steps and the residual signal $r(k)$ after k steps are then given by

$$x_c(N) := T_x \begin{bmatrix} y_r \\ a_u \end{bmatrix} \qquad\qquad r(k) := T_r(k) \begin{bmatrix} y_r \\ a_u \end{bmatrix}$$

where

$$T_x = Q_c \left[\tilde{A}_e^N Q_{ss} + \sum_{i=0}^{N-1} \tilde{A}_e^i \tilde{K}_e \quad \sum_{i=0}^{N-1} \tilde{A}_e^i \tilde{B}_e \right] \tag{3.13}$$

$$T_r(k) = \left[\tilde{C}_e (\tilde{A}_e^k Q_{ss} + \sum_{i=0}^{k-1} \tilde{A}_e^i \tilde{K}_e) + \tilde{D}_e \quad \tilde{C}_e \sum_{i=0}^{k-1} \tilde{A}_e^i \tilde{B}_e \right] . \tag{3.14}$$

In what follows, we formulate the problem of finding the worst case impact of this type of replay attacks, and then prove that this problem represents an instance of Problem 3.

Problem 5. (Estimating Impact of Replay Attacks)

$$\begin{aligned}
\underset{y_r, a_u}{\text{maximize}} \quad & I(\mathcal{S}_a, \mathcal{A}_a) = \left|\left| T_x \begin{bmatrix} y_r \\ a_u \end{bmatrix} \right|\right|_\infty \\
\text{subject to} \quad & -\delta_{y_r} \leq y_r \leq \delta_{y_r} \\
& S(k+1) \leq \delta_r \qquad k = 0, \ldots, N.
\end{aligned}$$

Proposition 2. Problem 5 is an instance of Problem 3 for the stateless, moving window, CUSUM, and MEWMA detector.

Proof. By defining $d = [y_r^T \; a_u^T]^T$, we see that the objective functions for both of the problems are of the same form. It follows from (3.13) that $r(k) = T_r(k)d$, so all the constraints $f_k(d) = S(k+1) \leq \delta_r$, $k = 0, \ldots, N$ are convex and symmetric in d due to Lemma 2. The reference can be expressed as $y_r = [\mathbf{I}_{n_{y_r}} \; \mathbf{0}_{n_{y_r} \times n_{a_u}}]d$, and the reference constraint is then given by

$$f_{y_r}(d) = ||Q_{y_r}[\mathbf{I}_{n_{y_r}} \; \mathbf{0}_{n_{y_r} \times n_{a_u}}]d||_\infty \leq 1$$

where Q_{y_r} is defined in the same way as in the proof of Proposition 1. This constraint is a convex and symmetric constraint in d due to the infinity norm. Thus, we conclude that Problem 5 is an instance of Problem 3. □

Replay Attacks with Blocking the Control Signals

The control signals and measurements in this variant of replay attack can be modeled by

$$\tilde{u}(k) = \Lambda_u u(k) \qquad\qquad \tilde{y}(k) = \tilde{\Lambda}_y y(k) + \Lambda_y y(0) \tag{3.15}$$

where $\Lambda_u \in \mathbb{R}^{n_u \times n_u}$ is defined in the same way as in (3.9), and $\Lambda_y, \tilde{\Lambda}_y \in \mathbb{R}^{n_y \times n_y}$ are given by

$$\Lambda_y(i,i) = \begin{cases} 1 & i \in \mathcal{S}_a \\ 0 & i \notin \mathcal{S}_a \end{cases} \qquad\qquad \tilde{\Lambda}_y = \mathbf{I}_{n_y} - \Lambda_y.$$

By combining (3.1), (3.2), (3.4), (3.15), and using the fact that

$$y(0) = [C_p \; \mathbf{0}_{n_y \times n_z} \; \mathbf{0}_{n_y \times n_s}]Q_{ss}y_r := Q_{ss}^y y_r$$

the system under the replay attack can be expressed in the form

$$\begin{aligned}
x_e(k+1) &= \tilde{A}_e x_e(k) + \tilde{B}_e y_r \\
r(k) &= \tilde{C}_e x_e(k) + \tilde{D}_e y_r
\end{aligned} \tag{3.16}$$

where

$$\tilde{A}_e = \begin{bmatrix} A_p + \tilde{B}_p D_f \tilde{C}_p & \tilde{B}_p C_f & \mathbf{0}_{n_x \times n_s} \\ B_f \tilde{C}_p & A_f & \mathbf{0}_{n_z \times n_s} \\ (K_d + B_d D_f)\tilde{C}_p & B_d C_f & A_d \end{bmatrix} \quad \tilde{B}_e = \begin{bmatrix} \tilde{B}_p(E_f + D_f \Lambda_y Q_{ss}^y) \\ K_f + B_f \Lambda_y Q_{ss}^y \\ B_d E_f + (B_d D_f + K_d)\Lambda_y Q_{ss}^y \end{bmatrix}$$

$$\tilde{C}_e = \begin{bmatrix} (D_d D_f + E_d)\tilde{\Lambda}_y C_p & D_d C_f & C_d \end{bmatrix} \qquad \tilde{D}_e = D_d E_f + (D_d D_f + E_d)\Lambda_y Q_{ss}^y$$

and $\tilde{B}_p = B_p\Lambda_u$, $\tilde{C}_p = \tilde{\Lambda}_y C_p$. By comparing equations (3.16) and (3.10), we conclude that the system under this variant of replay attack is described by the same equations as under the denial of service attacks. Thus, it follows that the problem of finding the attack impact in the case of replay attacks can be reduced to Problem 4, which is an instance of Problem 3.

3.4.5 Additive False Data Injection Attacks

In what follows, we introduce two attack strategies where the attacker injects carefully designed signals into sensors and actuators under its control.

The Worst Case False Data Injection Attacks

The worst case false data injection attacks represent very sophisticated attack strategy. The attack signal $a(0), \ldots, a(N)$ is calculated based on the full model knowledge and then fed into the system through the corrupted sensors and actuators. Although very powerful, this attack is more unlikely than for example denial of service attack due to the need of full model knowledge.

This attack is additive in nature, thus the attack trajectory of the extended system (3.6) caused by the attack can be divided into the trajectory $x_e^0(k), r^0(k)$ driven by the steady state value and the reference, and the trajectory $x_e^a(k), r^a(k)$ driven by the attack signal

$$x_e(k) = x_e^0(k) + x_e^a(k) \qquad\qquad r(k) = r^0(k) + r^a(k).$$

From Assumption 1, the system has reached steady state before the attack starts. Since the attack does not change the system structure, it follows that during the attack

$$x_e^0(k) = x_e(0) \qquad\qquad r^0(k) = 0.$$

Based on the previous discussion, and using the extended system equations (3.6), the critical states after N steps and the residual signal $r(k)$ after k steps can be expressed as

$$x_c(N) := T_x \begin{bmatrix} y_r \\ a_{0:N} \end{bmatrix} \qquad\qquad r(k) := T_r(k) a_{0:k}$$

where $a_{0:k} = [a(0)^T \ \ldots \ a(k)^T]^T$, and

$$\begin{aligned} T_x &= Q_c \begin{bmatrix} Q_{ss} \ A_e^{N-1}B_e \ \ldots \ B_e \ \mathbf{0}_{n_e \times n_a} \end{bmatrix} \\ T_r(k) &= \begin{bmatrix} C_e A_e^{k-1} B_e & \ldots & C_e B_e & D_e \end{bmatrix}. \end{aligned} \tag{3.17}$$

The worst case impact of the false data injection attacks can then be obtained by solving the following optimization problem.

Problem 6. (Estimating Impact of False Data Injection Attacks)

$$\begin{aligned} \underset{y_r, a_{0:N}}{\text{maximize}} \quad & I(\mathcal{S}_a, \mathcal{A}_a) = \left|\left| T_x \begin{bmatrix} y_r \\ a_{0:N} \end{bmatrix} \right|\right|_\infty \\ \text{subject to} \quad & -\delta_{y_r} \leq y_r \leq \delta_{y_r} \\ & S(k+1) \leq \delta_r \qquad k = 0, \ldots, N. \end{aligned}$$

This problem is also an instance of Problem 3, as proved in the following proposition.

Proposition 3. Problem 6 is an instance of Problem 3 for the stateless, moving window, CUSUM, and MEWMA detector.

Proof. The decision variables in Problem 6 are y_r and $a_{0:N}$, so $d = [y_r^T \; a_{0:N}^T]^T$. Thus, it follows that the problems have objective functions of the same form. If we define matrix

$$T'_r(k) = [\mathbf{0}_{n_r \times n_{y_r}} \; T_r(k) \; \mathbf{0}_{n_r \times (N-k)n_a}]$$

we can express the residual signal $r(k)$ during the attack as $r(k) = T'_r(k)d$, $k = 0, \ldots, N$. The constraints $f_k(d) = S(k+1) \leq \delta_r$, $k = 0, \ldots, N$ are then convex and symmetric constraints in d due to Lemma 2. If we define the reference as $y_r = [\mathbf{I}_{n_{y_r}} \; \mathbf{0}_{n_{y_r} \times (N+1)n_a}]d$, then the reference constraint can be rewritten as

$$f_{y_r}(d) = ||Q_{y_r}[\mathbf{I}_{n_{y_r}} \; \mathbf{0}_{n_{y_r} \times (N+1)n_a}]d||_\infty \leq 1$$

where Q_{y_r} is defined in the same way as in the proof of Proposition 1. Since this is a convex and symmetric constraint in d, we conclude that Problem 6 is an instance of Problem 3 for all the investigated detectors. □

Bias Injection Attacks

Compared to the false data injection attack, the bias injection attack is less sophisticated since the attacker injects a constant bias in the corrupted signals instead of a time varying signal. The full model knowledge is still needed to design the worst-case bias.

The control inputs and measurements during the bias injection attack can then be expressed as

$$\tilde{u}(k) = D_u a_u + u(k) \qquad\qquad \tilde{y}(k) = D_y a_y + y(k).$$

Let $a = [a_u^T \;\; a_y^T]^T$. By inserting $a(0) = \ldots = a(N) = a$ in (3.17), the critical states after N steps and the residual signal $r(k)$ after k steps can be expressed as

$$x_c(N) = Q_c \left(Q_{ss} y_r + \sum_{i=0}^{N-1} A_e^i B_e a \right) := T_x \begin{bmatrix} y_r \\ a \end{bmatrix}$$

$$r(k) = \left(C_e \sum_{i=0}^{k-1} A_e^i B_e + D_e \right) a := T_r(k) a.$$

The problem of finding the worst case bias injection attack can then be formulated as follows.

Problem 7. (Estimating Impact of Bias Injection Attacks)

$$\begin{aligned} \underset{y_r, a}{\text{maximize}} \quad & I(\mathcal{S}_a, \mathcal{A}_a) = \left|\left| T_x \begin{bmatrix} y_r \\ a \end{bmatrix} \right|\right|_\infty \\ \text{subject to} \quad & -\delta_{y_r} \leq y_r \leq \delta_{y_r} \\ & S(k+1) \leq \delta_r \quad k = 0, \ldots, N. \end{aligned}$$

Proposition 4. Problem 7 is an instance of Problem 3 for the stateless, moving window, CUSUM, and MEWMA detector.

Proof. Decision variable in this problem is given by $d = [y_r^T \ a^T]^T$, hence the objective function is of the form $||T_x d||_\infty$. The proof that all the constraints of Problem 7 are convex and symmetric is similar to the proof of Proposition 3. If we define the matrix $T'_r(k) = [\mathbf{0}_{n_r \times n_{y_r}} \ T_r(k)]$, we can then express the residual signal $r(k)$ during the attack as $r(k) = T'_r(k)d$. If we let $f_k(d) = S(k+1) \leq \delta_r$, $k = 0, \ldots, N$, then we have that these constraints are convex and symmetric in d due to $r(k) = T'_r(k)d$ and Lemma 2. The reference this time can be defined as $y_r = [\mathbf{I}_{n_{y_r}} \ \mathbf{0}_{n_{y_r} \times n_a}]d$, and the reference constraint as

$$f_{y_r}(d) = ||Q_{y_r}[\mathbf{I}_{n_{y_r}} \ \mathbf{0}_{n_{y_r} \times n_a}]d||_\infty \leq 1$$

with Q_{y_r} defined in the same way as in the proof of Proposition 1. Thus, the reference constraint is convex and symmetric in d, which concludes the proof. □

3.5 Illustrative Example

In this section, we illustrate how the attack models we proposed can be used to conduct risk assessment. We observe the part of a chemical process [86] shown on Figure 3.2. The control objective is to keep a constant liquid level and a constant temperature in Tank 2. This objective is achieved by injecting hot water from Tank 1, and cold water from Tank 3. The cyber-infrastructure is assumed to be as shown in Figure 3.3. The communication links that connect the routers with the controller are unprotected, and our task is to decide which one is more important to be protected.

3.5.1 Process Model

The states of the system are the volume in Tank 3 (x_1), the volume in Tank 2 (x_2), and the temperature in Tank 2 (x_3). All three states are measurable. The control signals are the flow rate of Pump 2 (u_1), the openness of the valve (u_2), the flow rate of Pump 1 (u_3), and the power of the heater (u_4). The system was discretized

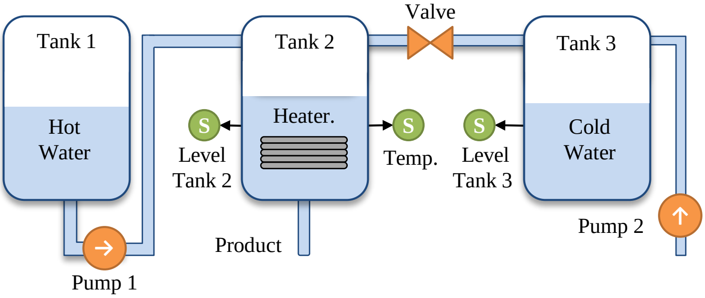


Figure 3.2: Chemical process with four actuators (two pumps, heater, and valve), and three sensors (two level sensors and one temperature sensor).

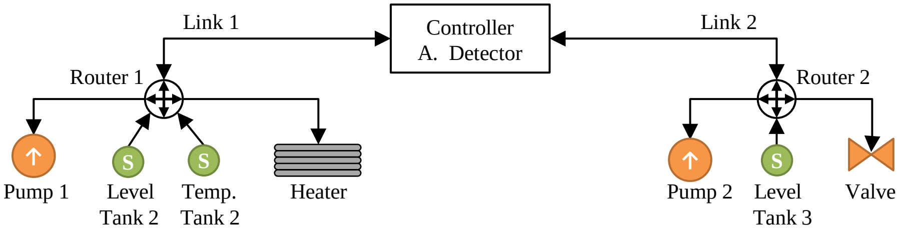


Figure 3.3: Cyber infrastructure of the process.

with a sampling time of $T_s = 200\,\mathrm{s}$ using zero order hold method, and the following state space model was obtained

$$A_p = \begin{bmatrix} 0.9121 & 0 & 0 \\ 0.0850 & 0.9364 & 0 \\ -0.0821 & 0.0014 & 0.8025 \end{bmatrix} \quad B_p = \begin{bmatrix} 17.20 & -4.3947 & 0 & 0 \\ 0.7857 & 4.2507 & 9.6771 & 0 \\ -0.7790 & -4.1062 & 3.5978 & 40.0348 \end{bmatrix}$$
$$C_p = \mathbf{I}_3 \qquad D_p = \mathbf{0}_{3\times 4}.$$

The matrices of a feedback controller we designed are

$$A_f = \begin{bmatrix} -0.3121 & -0.0850 & 0.0821 \\ 0 & -0.0861 & -0.0014 \\ 0 & 0 & -0.0525 \end{bmatrix} \quad B_f = \begin{bmatrix} 0.7121 & 0.0850 & -0.0821 \\ 0 & 0.6361 & 0.0014 \\ 0 & 0 & 0.5525 \end{bmatrix}$$

Table 3.1: Impact of different attack strategies for the stateless detector.

Attack Strategy	**Impact Link 1**	**Impact Link 2**
Denial of Service	1.2164	1.4343
Rerouting	1.8460	1.4909
Sign Alternation	1.8161	2.1535
Replay (Injecting Bias)	∞	2.7853
False Data Injection	∞	3.4136
Bias Injection	4.4002	2.7923

$$C_f = \begin{bmatrix} -0.0289 & -0.0037 & 0.0002 \\ 0.0034 & -0.0145 & 0.0009 \\ -0.0079 & -0.0332 & -0.0004 \\ 0.0025 & 0.0014 & -0.0074 \end{bmatrix} \qquad D_f = \mathbf{0}_{4\times 3}$$

$$K_f = \begin{bmatrix} 0.6000 & 0 & 0 \\ 0 & 0.4500 & 0 \\ 0 & 0 & 0.5000 \end{bmatrix} \qquad E_f = \begin{bmatrix} 0.0349 & 0.0000 & 0.0000 \\ 0 & 0 & 0 \\ -0.0028 & 0.0465 & -0.0000 \\ 0.0009 & -0.0042 & 0.0125 \end{bmatrix}.$$

Finally, the matrices of a detection filter are

$$A_d = \begin{bmatrix} 0.2000 & -0.0850 & 0.0821 \\ 0.0850 & 0.3000 & -0.0014 \\ -0.0821 & 0.0014 & 0.2500 \end{bmatrix} \qquad B_d = \begin{bmatrix} 17.20 & -4.39 & 0 & 0 \\ 0.79 & 4.2 & 9.68 & 0 \\ -0.78 & -4.1 & 3.60 & 40.04 \end{bmatrix}$$

$$C_d = -\mathbf{I}_3 \qquad D_d = \mathbf{0}_{3\times 4}$$

$$K_d = \begin{bmatrix} 0.7121 & 0.0850 & -0.0821 \\ 0 & 0.6361 & 0.0014 \\ 0 & 0 & 0.5525 \end{bmatrix} \qquad E_d = \mathbf{I}_3.$$

We considered stateless detector with $Q_r = \mathbf{I}_3$ and $\delta_{y_r} = 1$. The attack length was set to $N = 20$. We assume the critical states to be x_2 and x_3, so $Q_c = [\mathbf{0}_{2\times 1}\ \mathbf{I}_2\ \mathbf{0}_{2\times 6}]$. Note that for the attack on Link 1, we have $\mathcal{A}_a^1 = \{3, 4\}$ and $\mathcal{S}_s^1 = \{2, 3\}$, while $\mathcal{A}_a^2 = \{1, 2\}$ and $\mathcal{S}_s^2 = \{1\}$ for the attack on Link 2.

3.5.2 Simulation Results

For the given configuration, we derive the impact $I(\mathcal{S}_a, \mathcal{A}_a)$ of the presented cyber-attacks for different anomaly detectors. The results are shown in Table 3.1. We see from the table that the false data injection and replay attack can have devastating impacts on the system if the attacker has access to Link 1. This is according to expectation, since an attack on Link 1 can directly manipulate the measurements of the critical states x_2 and x_3, and these states are not measurable from sensors measurements transmitted over Link 2. In that case, $\ker(T_r) \not\subseteq \ker(T_x(l,:))$ for $l \in \{1,2\}$, so the attacker can make an arbitrary large impact. This also shows that in certain cases simpler attack such as a replay attack, might be equally as dangerous as false data injection attacks with full model knowledge.

We can also see that the attack impact on Link 1 is not always larger then the attack impact on Link 2. In particular, the impact of denial of service and sign alternation attacks on Link 2 is larger than the impact on Link 1. Nevertheless, given that the attack impact on Link 1 is higher for most of the attack strategies, we conclude that protecting this link is more important than protecting Link 2.

3.6 Summary

In this chapter, we treated a problem of estimating attack impact in control systems with various types of anomaly detectors. In particular, a modeling framework for estimating the impact of several cyber-attack strategies was proposed. The framework can be used to estimate the impact of denial of service, sign alternation, rerouting, replay, false data injection, and bias injection attack strategies. The anomaly detectors for which the framework is applicable are stateless, moving window, cumulative sum, and multivariate exponentially weighted moving average. The infinity norm of the state vector after a fixed number of time steps was adopted as impact metric. We showed that for this metric, the problem of estimating the impact of aforementioned attack strategies can be reduced to solving multiple convex minimization problems. Therefore, the exact value of the attack impact can be obtained using standardized convex optimization solvers. Finally, it was illustrated how the framework can be used for risk assessment on a numerical example.

Chapter 4

Estimating Attack Impact in Monitoring Systems

In this chapter, we consider an ICS that has a task to estimate the state of a physical process from noisy measurements. The system consists of a noisy plant, an estimator (Kalman filter), and an anomaly detector (chi-squared test). For the measure of the attack impact, we adopt the steady state mean square estimation error, and for the attack strategy, bias injection attacks. For this combination of impact measure and attack strategy, we characterize the worst case impact. In addition, a lower bound for the attack impact is derived. Finally, simulation study is included to illustrate the theoretical findings. As we shall see, the problem of estimating impact becomes more challenging once the model and measurement noises are taken into account.

4.1 Introduction

4.1.1 Literature Review

The problem of Kalman filtering in the presence of cyber-attacks gained a lot of popularity within control community. In general, the literature on this subject can be divided into two sets. The first set of literature addresses the problem of estimating bounds of the worst case performance degradation. One of the first works on this topic was [88], where authors considered a Kalman filter equipped with chi-squared anomaly detector, and proposed an algorithm that calculates upper and lower bounds of the performance degradation under stealthy cyber-attacks. Following this work, Murguia *et al.* [89] proposed easier to calculate ellipsoidal bounds using linear matrix inequalities. Another extension of this work was [90], where authors observed the performance degradation in the presence of an authentication mechanism. Bai *et al.* [91, 92] analyzed the performance degradation of Kalman filter in presence of cyber-attacks, but did not restrict themselves to any particular anomaly detector. The authors derived fundamental performance bounds and also introduced several attack strategies that achieve these bounds.

The second set of literature studies concrete attack strategies. Kwon *et al.* [93] analyzed an attacker with different sets of resources, and proposed several strategies the attacker needs to follow in order to stay undetected. Chen *et al.* [94] observed an attacker with the goal to move the system state to a certain point in the state space, and show that this attack strategy can be transformed to a convex optimization program. The optimal linear cyber-attack strategy against the Kalman filter was analyzed in [44, 95, 96].

Similar to other literature, we consider the configuration where the Kalman filter is used as an estimator, and the chi-squared test is used to detect anomalies. However, we focus our attention to bias injection attacks [38]. This attack strategy was introduced in [38], but was studied in a deterministic setting. Our aim in this chapter is to extended the analysis of these attacks to stochastic estimation problem, with the purpose of estimating the attack impact.

4.1.2 Contributions

The contributions of this chapter are threefold. Firstly, we extend the bias injection attack strategy to Kalman filtering problem. This generalizes [38], where these attacks were studied in a deterministic setting, and phenomena such as false alarms in stochastic systems were neglected. Secondly, we show that the impact of the worst-case bias injection attack in a stochastic setting can be analyzed by a quadratically constrained quadratic program, which has an analytical solution (Theorem 1). Thirdly, we derive a lower bound on the attack impact that is dependent of the number of sensors the attacker controls (Theorem 2). Finally, we illustrate the theoretical findings on a model of quadruple tank process.

4.1.3 Organization

The chapter is organized as follows. In the remainder of this section, we introduce some mathematical background. In Section 4.2, we introduce the estimation problem in the presence of cyber-attacks. In Section 4.3, we formulate the problem of finding the worst case bias injection attacks. In Section 4.4, we provide analysis of the attack impact and derive a lower bound of the attack impact in terms of number of sensors. In Section 4.5, we illustrate the obtained results on a numerical example. In Section 4.6, we briefly summarize the results presented in the chapter.

4.1.4 Mathematical Background

4.1.4.1 Non-Central Chi-Squared Distribution

Let X be an $n-$ dimensional Gaussian random vector with mean value μ_X and covariance matrix $\Sigma_X = \mathbf{I}_n$. The random variable $\chi^2_\mu = X^T X$, is then distributed according to the noncentral chi-squared distribution. This distribution is specified by two parameters. The first parameter is the number of degrees of freedom. This parameter is equal to n, the dimension of the random vector X. The second parameter is called noncentrality parameter, and we denote it by λ. This parameter is equal to the squared Euclidian norm of the mean value of X, that is, $\lambda = ||\mu_X||_2^2$.

Complementary culmulative distribution function of the random variable χ_μ^2 is given by

$$\mathbb{P}(\chi_\mu^2 > \tau) = Q_{\frac{n}{2}}(\sqrt{\lambda}, \sqrt{\tau})$$

where $\tau > 0$, and $Q_{\frac{n}{2}}(\sqrt{\lambda}, \sqrt{\tau})$ represents the generalized Marcum Q-function. It is known that $Q_\nu(a, b)$ is contiunous in a for $b, \nu > 0$ [97, Secton I]. Moreover, the following result holds.

Lemma 3. The generalized Marcum Q-function $Q_\nu(a, b)$ is strictly increasing in a for all $a \geq 0$ and $b, \nu > 0$.

Proof. We refer the reader to [98, Section III]. □

From Lemma 3, it follows directly that for fixed n and τ, $Q_{\frac{n}{2}}(\sqrt{\lambda}, \sqrt{\tau})$ is strictly increasing in λ.

4.1.4.2 Generalized Eigenvalues and Eigenvectors

In this section we revisit some results concerning generalized eigenvalues and eigenvectors.

Definition 1. Let A, B be matrices in $\mathbb{C}^{n\times n}$. The set of generalized eigenvalues of the matrix pencil(pair) (A, B) is defined as

$$\lambda(A, B) = \{\lambda \in \mathbb{C} : \det(A - \lambda B) = 0\}.$$

The generalized eigenvector x of (A, B) is a nontrivial solution of the equation

$$Ax = \lambda Bx$$

with $\lambda \in \lambda(A, B)$.

In the case of real pencils, with $A \succeq 0$ and $B \succ 0$, there exists exactly n real nonnegative generalized eigenvalues [99]. Moreover, the following result holds.

Lemma 4. Let $A \succeq 0$, $B \succ 0$, with $A, B \in \mathbb{R}^{n\times n}$, and let $0 \leq \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$ be the generalized eigenvalues of the pencil (A, B). Then for any $j \in \{1, ..., n\}$ we have

$$\lambda_j = \underset{\substack{U\subseteq\mathbb{R}^n \\ \dim(U)=j}}{\text{minimize}} \ \underset{\substack{x\in U \\ x\neq 0}}{\text{maximize}} \ \frac{x^T Ax}{x^T Bx},$$

where U represents a subspace of the vector space $\mathbb{R}^n$.

Proof. We refer the reader to [100, 101]. □

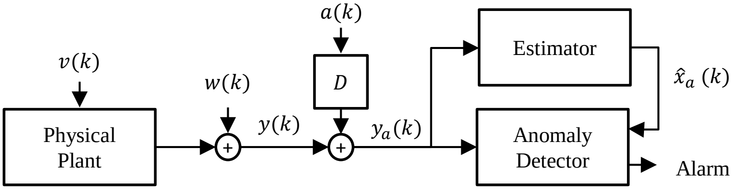


Figure 4.1: A schematic of an estimator in the presence of cyber-attacks. Physical plant is driven with unknown process noise $v(k)$, and the sensor measurements are corrupted with measurement noise $w(k)$. At certain point in time, the attack begins. The measurement signal $y(k)$ is intercepted by an an attacker, and signal $Da(k)$ is added to it, so the estimator and anomaly detector receive attacked version of the measurement signal $y_a(k)$.

4.2 Model Setup

The schematic of the system we consider is shown in Figure 4.1. The system represents the interconnection of a plant, an estimator, an anomaly detector, and an attacker. At some point, the attacker starts changing the sensor measurements that it controls. Our aim in this section is to characterize how the corrupted measurements influence the state estimate, and the signal from the anomaly detector.

4.2.1 Plant Model

The plant is modeled as a linear time-invariant system

$$\mathcal{P}\colon \begin{cases} x(k+1) = Ax(k) + v(k) \\ \quad\;\; y(k) = Cx(k) + w(k) \end{cases} \tag{4.1}$$

where $x(k) \in \mathbb{R}^{n_x}$ is the system state, $v(k) \in \mathbb{R}^{n_x}$ is the process noise, $y(k) \in \mathbb{R}^{n_y}$ is the vector of sensor measurements, and $w(k) \in \mathbb{R}^{n_y}$ is the measurement noise. The processes $\{v(k)\}$ and $\{w(k)\}$ are independent, zero mean, Gaussian white processes with covariance matrices $\Sigma_v \succeq 0$ and $\Sigma_w \succ 0$, respectively. The initial state of the system $x(0)$ is a Gaussian random variable with mean value $\mu_{x(0)}$ and covariance matrix $\Sigma_{x(0)} \succ 0$, independent of $\{v(k)\}$ and $\{w(k)\}$. We assume the pair (C, A) is detectable, and the pair $(\Sigma_v^{1/2}, A)$ is stabilizable. The set of all the sensors is denoted with $\mathcal{S} = \{1, 2, ..., n_y\}$.

4.2.2 Estimator

The vector $x(k)$ is estimated using the Kalman filter. The state estimate evolves according to the equation

$$\mathcal{E}\colon \begin{cases} \hat{x}(k+1|k) = (A - K(k)C)\hat{x}(k|k-1) + K(k)y(k) \\ \qquad e(k) = x(k) - \hat{x}(k|k-1) \end{cases} \tag{4.2}$$

where $\hat{x}(k+1|k) \in \mathbb{R}^{n_x}$ represents the one step ahead prediction, and the matrix $K(k)$ is the Kalman gain matrix. The filter is initialized with $\hat{x}(0|-1) = \mu_{x(0)}$. The estimation error $e(k)$ is a Gaussian random variable with zero mean and covariance matrix $\Sigma_e(k)$, which is characterized below.

Although the system (4.1) is time invariant, the matrices $K(k)$ and $\Sigma_e(k)$ are updated in every time step according to the equations

$$\begin{aligned} K(k) &= A\Sigma_e(k)C^T(C\Sigma_e(k)C^T + \Sigma_w)^{-1} \\ \Sigma_e(k+1) &= (A - K(k)C)\Sigma_e(k)A^T + \Sigma_v \end{aligned}$$

with $\Sigma_e(0) = \Sigma_{x(0)}$. Under the stabilizability and detectability assumptions we introduced, the matrices $K(k)$ and $\Sigma_e(k)$ converge to unique constant steady states. Additionally, the matrix $A - KC$ in steady state is asymptotically stable [102, Chapter 4]. Therefore, for sake of simplicity, we assume that the Kalman filter has reached a steady state before the attack starts. In the remainder of the chapter, the steady state values of $K(k)$ and $\Sigma_e(k)$ are denoted by K and Σ_e, respectively.

4.2.3 Detector

In order to detect possible anomalies, a chi-squared test is used. The first step of the anomaly detection procedure is to generate a residual signal

$$\mathcal{D}\colon \Big\{ r(k) = y(k) - C\hat{x}(k|k-1). \tag{4.3}$$

Note that $C\hat{x}(k|k-1)$ is the estimate of $y(k)$, thus $r(k)$ represents the difference between $y(k)$ and its modeled behavior. In the absence of anomalies, $\{r(k)\}$ is a white Gaussian process with zero mean and covariance matrix

$$\Sigma_r = C\Sigma_e C^T + \Sigma_w.$$

The statistical approach we use assumes that the presence of anomalies would change the distribution of $r(k)$. Thus, the second step of the anomaly detection procedure is to define a suitable test to judge if the residual $r(k)$ comes from the Gaussian distribution that we mentioned previously, or if an anomaly occurred and the distribution changed. A simple method that is used for this purpose is to test if the squared distance measure

$$\chi^2(k) = r^T(k)\Sigma_r^{-1}r(k) = \|\Sigma_r^{-1/2}r(k)\|_2^2$$

is greater than a sufficiently large threshold $\tau > 0$. The random variable $\chi^2(k)$ is distributed according to a chi-squared probability distribution, which is the reason why this test is called a chi-squared test.

In absence of anomalies, the random variable $\chi^2(k)$ takes relatively small values most of the time. The cases when this signal exceeds the threshold τ might be an indication that an anomaly occurred. However, it is important to realize that any threshold τ is occasionally breached even when anomalies are not present. These false alarms happen due to the random noise, with probability

$$\mathbb{P}(\chi^2(k) > \tau) =: \alpha.$$

Large values of τ would decrease the false alarm probability α, but also the sensitivity to anomalies. On the other hand, a small value of τ would result in high probability of false alarms, which is also undesirable. Hence, τ has to be chosen as a reasonable trade-off between these two phenomena.

Remark 2. The presence of false alarms makes attack detection difficult. If an alarm occurs when the attack is present, and if distribution of $\chi^2(k)$ is not changed considerably, the alarm may be classified as a consequence of noise.

4.2.4 Attack Model

Suppose that at the time instant $k = k_0$ the attacker gains control over the set $\mathcal{S}_a \subseteq \mathcal{S}$ of the sensor measurements, and starts changing them. From that point onwards, the measurement equation becomes

$$y_a(k) = y(k) + Da(k) \tag{4.4}$$

where the vector $a(k) \in \mathbb{R}^{n_a}$ represents the signal the attacker injects, and $n_a = |\mathcal{S}_a|$ is the number of sensors the attacker controls. The matrix $D \in \mathbb{R}^{n_y \times n_a}$ is a matrix that maps $a(k)$ to corresponding measurements in the following way. Let $\mathcal{S}_a = \{j_1, j_2, ..., j_{n_a}\}$. Then the elements $(j_1, 1), (j_2, 2)..., (j_{n_a}, n_a)$ of the matrix D are one, and the rest are zero.

In case the attack is not detected, the attacked measurements $y_a(k)$ are used to construct the state estimate. Due to the linearity of the Kalman filter, the attacked estimate $\hat{x}_a(k)$ is the sum of the responses to $y(k)$ and $a(k)$, that is,

$$\hat{x}_a(k) = \hat{x}(k|k-1) + \Delta\hat{x}(k) \tag{4.5}$$

where $\Delta\hat{x}(k)$ is dependent just on the attack signal and propagates by

$$\Delta\hat{x}(k+1) = (A - KC)\Delta\hat{x}(k) + KDa(k). \tag{4.6}$$

The error between state $\hat{x}(k)$ and the corrupted estimate $\hat{x}_a(k)$ now becomes

$$x(k) - \hat{x}_a(k) = e(k) - \Delta\hat{x}(k). \tag{4.7}$$

The attack influences the residual signal $r(k)$ as well. From (4.3), the attacked residual signal changes to

$$y_a(k) - C\hat{x}_a(k) = r(k) + \Delta r(k)$$

where $\Delta r(k)$ can be obtained from (4.4) and (4.5) as

$$\Delta r(k) = Da(k) - C\Delta\hat{x}(k). \tag{4.8}$$

We assume the goal of the attacker to be to increase the mean square of estimation error (4.7), and at the same time remain undetected. In next section, we introduce bias injection attacks as one of the possible strategies to construct signal $a(k)$ that accomplishes this goal.

4.3 Worst Case Bias Injection Attack

In the bias injection attack scenario, the attack signal $a(k)$ slowly converges to a constant vector a. In this section, we formulate the problem of finding a vector a that maximizes mean square estimation error in steady state, and does not increase alarm probability more then a certain threshold.

By substituting $a(k)$ with a in (4.6), we get

$$\Delta\hat{x}(k+1) = (A - KC)\Delta\hat{x}(k) + KDa.$$

As we mentioned earlier, matrix $A - KC$ of the Kalman filter is asymptotically stable, hence both $\Delta\hat{x}(k)$ and $\Delta r(k)$ reach steady states. The steady state equations for $\Delta\hat{x}(k)$ and $\Delta r(k)$ are

$$\Delta\hat{x} = (A - KC)\Delta\hat{x} + KDa \tag{4.9}$$

$$\Delta r = Da - C\Delta\hat{x}. \tag{4.10}$$

Since $A - KC$ is stable, $\mathbf{I}_n - A + KC$ is invertible, thus the solution of (4.9) is unique and given by

$$\Delta\hat{x} = (\mathbf{I}_n - A + KC)^{-1}KDa =: G_{\hat{x}}Da. \tag{4.11}$$

Combining (4.11) with (4.10) gives

$$\Delta r = Da - CG_{\hat{x}}Da = (\mathbf{I}_m - CG_{\hat{x}})Da =: G_rDa. \tag{4.12}$$

Remark 3. Note that the previous discussion holds if the bias injection attack is not detected during the transient phase. We will assume that this is not the case since the attacker can make the transient smooth by increasing the attack slowly.

We define the attacker's objective as to increase the mean square of error (4.7) once $\Delta\hat{x}(k)$ reaches steady state $\Delta\hat{x}$. The attack impact can then be defined as

$$I(\mathcal{S}_a) := \mathbb{E}\{\|e(k) - \Delta\hat{x}\|_2^2\} - \mathbb{E}\{\|e(k)\|_2^2\}. \tag{4.13}$$

The term $\mathbb{E}\{\|e(k)\|_2^2\}$ represents the mean square of estimation error without attacks, and it is constant added for the sake of scaling.

The constraint for the attacker is that it wants to remain undetected. The bias injection attacks preserves the nature of residual distribution, since it remains Gaussian. However, since the attack changes the mean value of the distribution, the alarm probability increases. For this reason, we assume that the constraint for the attacker is to not considerably increase the alarm probability. This constraint can be modeled as

$$\mathbb{P}(\|\Sigma_r^{-1/2}(r(k)+\Delta r)\|_2^2 > \tau) \leq \alpha + \Delta\alpha$$

where $\Delta\alpha > 0, \alpha + \Delta\alpha \leq 1$ is a threshold that models the attacker's willingness to risk detection. For example, say that the false alarm probability is $\alpha = 5\%$. In case that the alarm probability in presence of attacks raises to $\alpha + \Delta\alpha = 6\%$, the alarms will probably be classified as a consequence of noise, and the attack will remain undetected. However, in case that the alarm probability in presence of attack changes to $\alpha + \Delta\alpha = 25\%$, the attack will most likely be detected.

Based on the previous discussion, the problem the attacker wants to solve can be formalized as the following optimization problem.

Problem 8. (Estimating the worst case impact of bias injection attacks)

$$\underset{a}{\text{maximize}} \quad I(\mathcal{S}_a) = \mathbb{E}\{\|e(k) - \Delta\hat{x}\|_2^2\} - \mathbb{E}\{\|e(k)\|_2^2\} \tag{4.14a}$$

$$\text{subject to} \quad \mathbb{P}(\|\Sigma_r^{-1/2}(r(k)+\Delta r)\|_2^2 > \tau) \leq \alpha + \Delta\alpha. \tag{4.14b}$$

Note that Problem 8 in general has many parameters that are not necessarily known to the attacker. In this chapter we are interested in studying the worst case scenario, thus we adopt the following assumption about the attacker.

Assumption 2. We assume the attacker knows the structure of Problem 8.

In order to find the solution of Problem 8, we prove that it can be transformed into the following problem.

Problem 9. (Estimating the worst case impact of bias injection attacks)

$$\underset{a}{\text{maximize}} \quad I(\mathcal{S}_a) = \|G_{\hat{x}} D a\|_2^2 \tag{4.15a}$$

$$\text{subject to} \quad \|\Sigma_r^{-1/2} G_r D a\|_2^2 \leq \delta^2. \tag{4.15b}$$

As we show in Section 4.4, Problem 9 is a quadratically constrained quadratic program that has an analytical solution. In what follows, we show that for a particular choice of δ, Problem 8 is equivalent to Problem 9.

In order to prove the claim, we introduce two lemmas. In the first lemma, we prove that the objective functions of the problems are equivalent.

Lemma 5. The following equality holds

$$\mathbb{E}\{\|e(k) - \Delta\hat{x}\|_2^2\} - \mathbb{E}\{\|e(k)\|_2^2\} = \|G_{\hat{x}}Da\|_2^2.$$

Proof. The objective function (4.14a) can be rewritten as follows

$$\mathbb{E}\{\|e(k) - \Delta\hat{x}\|_2^2\} = \mathbb{E}\{\|e(k)\|_2^2 - 2\Delta\hat{x}^T e(k) + \|\Delta\hat{x}\|_2^2\}.$$

The term $\mathbb{E}\{2e(k)^T\Delta\hat{x}\}$ is equal to zero because $\Delta\hat{x}$ is constant, and $e(k)$ is zero mean. The term $\mathbb{E}\{\|e(k)\|_2^2\} = \text{Tr}(\Sigma_e)$ is constant, since it represents the part of the error coming from the noise. Thus, it follows

$$\mathbb{E}\{\|e(k) - \Delta\hat{x}\|_2^2\} - \mathbb{E}\{\|e(k)\|_2^2\} = \|G_{\hat{x}}Da\|_2^2 + \text{Tr}(\Sigma_e) - \text{Tr}(\Sigma_e) = \|G_{\hat{x}}Da\|_2^2$$

which concludes the proof. □

The second lemma is used to prove the equivalence of the constraints. The proof is based on the discussion in Section 4.1.4.1 about non-central chi-squared distribution.

Lemma 6. The alarm probability

$$\mathbb{P}(\|\Sigma_r^{-1/2}(r(k) + \Delta r)\|_2^2 > \tau) \tag{4.16}$$

is *strictly increasing* in $\|\Sigma_r^{-1/2}\Delta r\|_2^2 = \|\Sigma_r^{-1/2}G_r Da\|_2^2$.

Proof. The random vector $\Sigma_r^{-1/2}(r(k)+\Delta r)$ is Gaussian random vector with mean value

$$\mathbb{E}\{\Sigma_r^{-1/2}(r(k) + \Delta r)\} = \Sigma_r^{-1/2}\Delta r = \Sigma_r^{-1/2}G_r Da$$

and covariance matrix

$$\mathbb{E}\{\Sigma_r^{-1/2}r(k)r^T(k)(\Sigma_r^{-1/2})^T\} = \mathbf{I}_m.$$

Thus, $\|\Sigma_r^{-1/2}(r(k)+\Delta r)\|_2^2$ represents the non-central chi-squared random variable.

As explained in Section 4.1.4.1, the non-central chi-squared distribution is defined by two parameters. The first one is the number of degrees of freedom, and it is equal to the dimension of the random variable m. The second parameter is called the non-centrality parameter, and it is equal to the magnitude of the mean value $\mu_{\chi_a^2} = \|\Sigma_r^{-1/2}G_r Da\|_2^2$.

The alarm probability represents the complementary cumulative distribution function of the non-central chi-squared random variable. From Section 4.1.4.1, we know that cumulative distribution function is the generalized Marcum Q-function. Then it follows from Lemma 3 that (4.16) is strictly increasing in the non-centrality parameter for $m, \tau > 0$. Given that the non-centrality parameter is equal to $\|\Sigma_r^{-1/2}G_r Da\|_2^2$, the claim of the lemma follows. □

We are now ready to prove the equivalence between Problem 8 and Problem 9.

Theorem 1. There exists $\delta \in \mathbb{R}$ such that Problem 8 is equivalent to Problem 9.

Proof. Let $\bar{r} \in \mathbb{R}^m$ be any unit vector, and let δ be such that

$$\mathbb{P}(\|\Sigma_r^{-1/2} r(k) + \delta\bar{r}\|_2^2 > \tau) = \alpha + \Delta\alpha \tag{4.17}$$

is satisfied. Such δ exists, since the alarm probability is the Marcum Q-function, and this function is continuous and strictly increasing in δ^2 for $m, \tau > 0$.

From Lemma 5, we know that the objective functions (4.14a) and (4.15a) of the problems are equal, thus it is sufficient to prove that the feasible domains specified by the constraints are equal. We will use a contradiction argument to prove this. Assume that there exists a that satisfies constraint (4.14b)

$$\mathbb{P}(\|\Sigma_r^{-1/2}(r(k) + G_r D a)\|_2^2 > \tau) \leq \alpha + \Delta\alpha \tag{4.18}$$

but violates constraint (4.15b)

$$||\Sigma_r^{-1/2} G_r D a||_2^2 > \delta^2.$$

In that case, we have that

$$||\Sigma_r^{-1/2} G_r D a||_2^2 > \delta^2 = ||\delta\bar{r}||_2^2$$

and from (4.17) and (4.18) it follows

$$\mathbb{P}\big(\|\Sigma_r^{-1/2}(r(k) + G_r D a)\|_2^2 > \tau\big) \leq \mathbb{P}\big(\|\Sigma_r^{-1/2} r(k) + \delta\bar{r}\|_2^2 > \tau\big).$$

This is in contradiction with Lemma 6 where it was proven that the alarm probability is strictly increasing function of $||\Sigma_r^{-1/2} G_r D a||_2^2$.

In a similar manner, we disprove the existence of a that satisfies the quadratic constraint, but violates the probability constraint. Assume this time that there exists a that satisfies constraint (4.15b)

$$||\Sigma_r^{-1/2} G_r D a||_2^2 \leq \delta^2 \tag{4.19}$$

but violates constraint (4.14b)

$$\mathbb{P}(\|\Sigma_r^{-1/2}(r(k) + G_r D a)\|_2^2 > \tau) > \alpha + \Delta\alpha. \tag{4.20}$$

This is again in contradiction with Lemma 6, since from (4.19) we have

$$||\delta\bar{r}||_2^2 \geq ||\Sigma_r^{-1/2} G_r D a||_2^2$$

and from (4.20) and (4.17)

$$\mathbb{P}\big(\|\Sigma_r^{-1/2} r(k) + \delta\bar{r}\|_2^2 > \tau\big) < \mathbb{P}\big(\|\Sigma_r^{-1/2}(r(k) + G_r D a)\|_2^2 > \tau\big).$$

This concludes the proof. □

Remark 4. It is interesting to note that the optimization Problem 9 is of the same form as the one in [38], which was obtained by analyzing bias injection attacks in a deterministic setting.

Remark 5. Since there is no simple closed form for the Marcum Q-function, finding the parameter δ that satisfies

$$\mathbb{P}(\|\Sigma_r^{-1/2}r(k) + \delta\bar{r}\|_2^2 > \tau) = \alpha + \Delta\alpha$$

needs to be done either numerically, or by using a Monte Carlo method.

4.4 Attack Impact Analysis

In this section, we analyze Problem 9. As we shall see, generalized eigenspectrum of the matrix pencil $(D^T G_{\hat{x}}^T G_{\hat{x}} D, D^T G_r^T \Sigma_r^{-1} G_r D)$ have a crucial role in analysis of the attack impact. In particular, the worst case attack impact is proportional to the largest eigenvalue of this pencil, while the remaining generalized eigenvalues for $D = \mathbf{I}_m$ can be used to construct a lower bound of the worst case attack impact. Background on generalized eigenvalues and eigenvectors that is necessary to follow this section is provided in Section 4.1.4.2.

4.4.1 The Worst Case Attack Impact

It is always of interest to first check if the attacker is able to inflict arbitrary large damage while staying undetected at the same time. The following result shows that for the problem we consider, this is possible only in a very restricted case.

Proposition 5. A necessary condition for the optimal value of Problem 9 to be unbounded is that the matrix A has an eigenvalue equal to 1.

Proof. Assume that the attacker is able to increase the error arbitrarily. A necessary condition for that is existence of $Da \neq 0$ such that

$$\Sigma_r^{-1/2} G_r Da = \Sigma_r^{-1/2} \Delta r = 0.$$

Since matrix Σ_r^{-1} is positive definite, it follows $\Delta r = G_r Da = 0$. In that case, from (4.10) we have $Da = C\Delta\hat{x}$. But then it follows from (4.9) that

$$\Delta\hat{x} = (A - KC)\Delta\hat{x} + KDa = A\Delta\hat{x} + K(Da - C\Delta\hat{x}) = A\Delta\hat{x}.$$

We see that the last equation has a nontrivial solution only in case that matrix A has an eigenvalue equal to 1. □

Since matrix A has eigenvalue equal to 1 only in exceptional cases, which can be treated independently, we introduce the following assumption without significant loss of generality.

Assumption 3. We assume A does not have eigenvalue equal to 1.

The following result then holds.

Proposition 6. If A does not have eigenvalue equal to 1, then $\mathrm{null}(G_r) = \emptyset$.

Proof. We prove the claim by using contradiction. Assume that A does not have eigenvalue equal to 1, but $\mathrm{null}(G_r) \neq \emptyset$. In that case, there exists vector $d \neq 0$ such that $G_r d = 0$. Since $G_r = \mathbf{I}_m - CG_{\hat{x}}$, we have $d = CG_{\hat{x}}d$. Let

$$d' = G_{\hat{x}}d. \tag{4.21}$$

By multiplying both sides of (4.21) with $(\mathbf{I}_n - A + KC)$ from the right and using (4.11), it follows

$$Kd = d' - Ad' + KCd' = G_{\hat{x}}d - AG_{\hat{x}}d + KCG_{\hat{x}}d.$$

Since $d = CG_{\hat{x}}d$, then $KCG_{\hat{x}}d = Kd$, so it follows $G_{\hat{x}}d - AG_{\hat{x}}d = 0$. Since A does not have eigenvalue equal to 1, we conclude $G_{\hat{x}}d = 0$. However, from $d = CG_{\hat{x}}d$, it follows that $d = 0$, which is in contradiction with the initial assumption that $d \neq 0$, which concludes the proof. □

Under Assumption 3 and from Proposition 6, the solution of Problem 9 can be found analytically.

Lemma 7. [38, Theorem 11] Suppose Assumption 3 holds. The solution of Problem 9 is then given by

$$a^* = \pm \frac{\delta}{\|\Sigma_r^{-1/2} G_r D v^*\|_2} v^*$$

where v^* is the unit length generalized eigenvector that corresponds to the maximal generalized eigenvalue λ^* of the matrix pencil

$$(D^T G_{\hat{x}}^T G_{\hat{x}} D, D^T G_r^T \Sigma_r^{-1} G_r D). \tag{4.22}$$

The maximal impact in terms of the mean square estimation error is

$$I(\mathcal{S}_a) = \|G_{\hat{x}} D a^*\|_2^2 = \lambda^* \delta^2. \tag{4.23}$$

Thus, it follows from the result that the attack impact is the product of two essentially different parts. The first part is δ^2, which models that the attack can increase the impact by increasing the risk of being detected. The second part is λ^*, and it is dependent on the properties of the matrix pencil (4.22). Since in general we do not know how much is the attacker willing to risk, λ^* can be used as an estimate of the attack impact $I(\mathcal{S}_a)$.

4.4.2 Lower Bound of the Attack Impact

Besides the largest generalized eigenvalue, the other generalized eigenvalues prove to be useful for the attack analysis. If we set $D = \mathbf{I}_m$, the matrix pencil (4.22) becomes

$$(G_{\hat{x}}^T G_{\hat{x}}, G_r^T \Sigma_r^{-1} G_r). \tag{4.24}$$

Under Assumption 3, $\text{null}(G_r) = \emptyset$, which implies that $G_r^T \Sigma_r^{-1} G_r$ is positive definite and the pencil (4.24) has exactly m real nonnegative generalized eigenvalues, as explained in Section 4.1.4.2. It turns out that the impact of the attack with *any* combination of p sensors is always larger than p-th generalized eigenvalue of the matrix pencil (4.24).

Theorem 2. Suppose Assumption 3 holds. Denote by

$$0 \leq \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_m$$

the generalized eigenvalues of $(G_{\hat{x}}^T G_{\hat{x}}, G_r^T \Sigma_r^{-1} G_r)$. Assume the attacker has control over $p \in \{1, ..., m\}$ sensors. Then the maximal impact (4.23) conducted with any combination of p sensors is larger or equal to $\lambda_p \delta^2$.

Proof. In case that the attacker controls p measurements, the attack signal Da is sparse with p non-zero components. Define by

$$\mathbb{S}^p = \{Da \in \mathbb{R}^m | \text{card}(Da) \leq p\}$$

the set of all possible bias injection attacks that the attacker is able to construct using p measurements. From Lemma 4, it follows

$$\underline{\lambda}_p^* = \min_{\substack{U \subseteq \mathbb{S}^p \\ \dim(U) = p}} \max_{Da \in U} \frac{a^T D^T G_{\hat{x}}^T G_{\hat{x}} Da}{a^T D^T G_r^T \Sigma_r^{-1} G_r Da} \geq \min_{\substack{U \subseteq \mathbb{R}^m \\ \dim(U) = p}} \max_{x \in U} \frac{x^T G_{\hat{x}}^T G_{\hat{x}} x}{x^T G_r^T \Sigma_r^{-1} G_r x} = \lambda_p \tag{4.25}$$

since $\mathbb{S}^p \subseteq \mathbb{R}^m$. Recall that the attack impact is given by (4.23). By multiplying (4.25) with δ^2, it follows that minimal impact conducted with p sensors is larger or equal to $\lambda_p \delta^2$, which concludes the proof. □

4.5 Illustrative Example

In this section, we illustrate how the worst case bias injection attack influences the estimation error and the chi-squared distance, and further clarify Theorem 2.

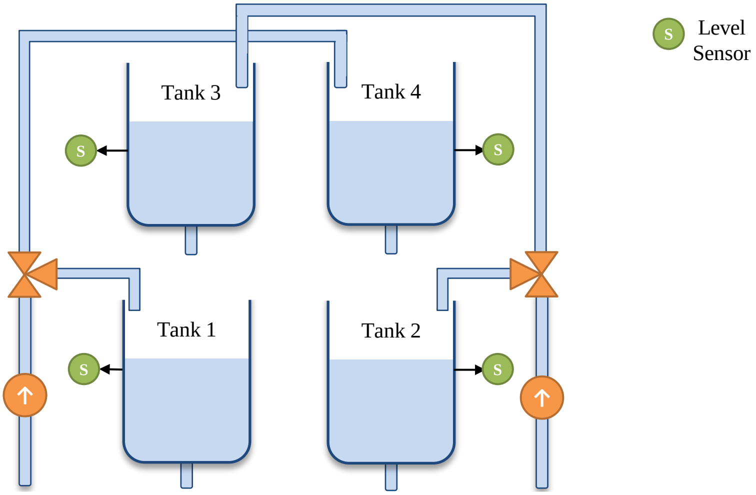


Figure 4.2: Schematic of the quadruple tank process. The level in each of the tanks is monitored by a level sensor.

4.5.1 Process Model

To illustrate theoretical results, we use the linearized discrete model of the quadruple-tank process shown on Figure 4.2. The matrices of the system are given by

$$A = \begin{bmatrix} 0.975 & 0 & 0.042 & 0 \\ 0 & 0.977 & 0 & 0.044 \\ 0 & 0 & 0.958 & 0 \\ 0 & 0 & 0 & 0.956 \end{bmatrix} \qquad \Sigma_v = 10^{-3} \begin{bmatrix} 0.8 & 0.2 & 2.7 & 0.7 \\ 0.2 & 0.8 & 0.7 & 2.7 \\ 2.7 & 0.7 & 9.0 & 2.3 \\ 0.7 & 2.7 & 2.3 & 9.0 \end{bmatrix}$$

$$C = 0.2\mathbf{I}_4 \qquad \Sigma_w = 10^{-3}\text{diag}(2.5, 2.5, 0.5, 0.5).$$

The states of the system are the levels in tanks, which are all measured by the level sensor.

4.5.2 Evolution of Estimation Error and Chi-Squared Distance

The threshold of the detector was set to $\tau = 9.488$, which corresponds to approximately 5% false alarm probability. We assumed that the attacker control sensors $\mathcal{S}_a = \{1, 3\}$, and does not want to increase the alarm rate more than $\Delta\alpha = 1\%$. From these values of τ and $\Delta\alpha$, we estimated $\delta = 0.4457$. The attack signal starts slowly increasing from 0 at time instant $k = 2500$, and converges to the final value of a^* at $k = 5000$.

Mean square estimation error under the normal operation (blue) and under the worst case bias injection attack (red) are shown in Figure 4.3. As we can see, the estimation error during the attack is considerably larger then during the normal

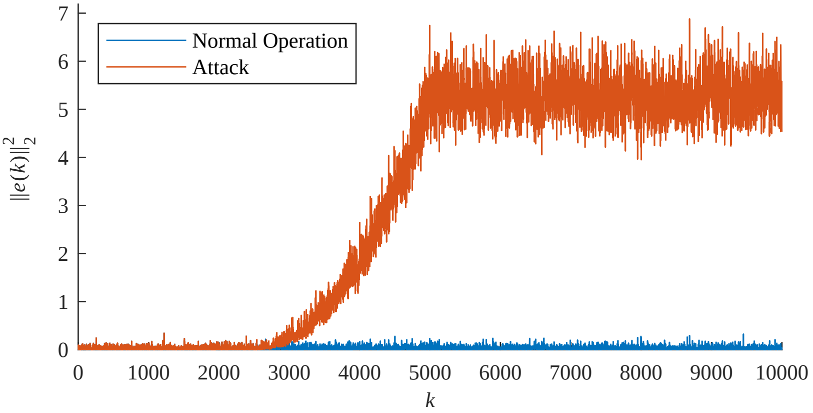


Figure 4.3: Estimation error under normal operation, and under the worst case bias injection attack.

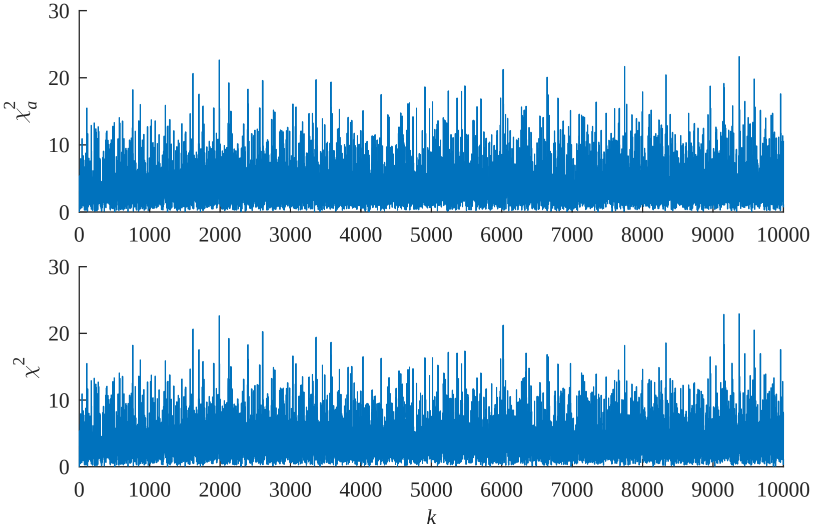


Figure 4.4: Chi-squared metric under the normal operation (lower plot), and under the worst case bias injection attack (upper plot).

operation. Evolution of chi-squared distance metrics under the normal operation and under the worst case bias injection attack are shown in Figure 4.4. In contrast to mean squared estimation error, these two two sequences are indistinguishable.

Table 4.1: Maximal generalized eigenvalue of the matrix pencil (4.22) for different combinations of attacked sensors.

Attacked sensors $\mathcal{S}_a$	**Impact** $I(\mathcal{S}_a) = \lambda^*$	**Attacked sensors** $\mathcal{S}_a$	**Impact** $I(\mathcal{S}_a) = \lambda^*$
$\{1,2,3,4\}$	38.00	$\{2,4\}$	30.07
$\{1,2,3\}$	26.38	$\{2,3\}$	0.08
$\{1,2,4\}$	30.07	$\{3,4\}$	0.08
$\{1,3,4\}$	26.38	$\{1\}$	0.01
$\{2,3,4\}$	30.07	$\{2\}$	0.02
$\{1,4\}$	0.07	$\{3\}$	0.08
$\{1,3\}$	26.38	$\{4\}$	0.07
$\{1,2\}$	0.02	$\emptyset$	0

4.5.3 Lower Bounds

In what follows, we clarify lower bound introduced in Theorem 2. We assume for simplicity that $\delta^2 = 1$, so the attack impact (4.23) is equal to the largest generalized eigenvalue. We first set the matrix D to $D = \mathbf{I}_m$, and calculate the eigenvalues of the pencil $(G_{\hat{x}}^T G_{\hat{x}}, G_r^T \Sigma_r^{-1} G_r)$. The eigenspectrum of this matrix pencil is given by $\lambda_1 = 6.6 \cdot 10^{-4}$, $\lambda_2 = 8.8 \cdot 10^{-4}$, $\lambda_3 = 22.26$, $\lambda_4 = 38.00$.

The maximal generalized eigenvalues for different combinations of attacked sensors are shown in Table 4.1. According to Theorem 2, λ_3 is lower bound for the impact of any worst case bias injection attack conducted with three sensors. From Table 4.1, we see that the ratio of the largest value of the attack and bound is $30.07/22.26 \approx 1.35$, so λ_3 represents relatively good approximation of impact with three sensors. However, the bound λ_2 is quite loose. The largest value of impact is 30.07, while the bound is equal to $8.8 \cdot 10^{-4}$, which represents approximately $3.4 \cdot 10^4$ times smaller value.

4.6 Summary

In this chapter, the problem of estimating the impact of bias injection attacks was considered. The system consisted of a noisy plant, a Kalman filter, and chi-squared anomaly detector. The mean square estimation error in steady state was adopted as impact metric. It was proven that the problem of finding the worst-case bias injection attack can be reduced to a quadratically constrained quadratic program, for which the optimal value can be obtained. Additionally, a lower bound of the attack impact was derived. The results of the chapter were illustrated in a simulation study.

Chapter 5

Allocating Security Measures in ICSs

In this chapter, we consider Problem 2. Recall from Chapter 1 that this problem is on selecting the least expensive set of security measures that prevents the high risk vulnerabilities. However, both the problem of finding the high-risk vulnerabilities (constructing security measure allocation problem), and then allocating security measures (solving security measure allocation problem) become challenging once the number of vulnerabilities and measures is large. These two challenges are tackled within the chapter. In particular, we propose an algorithm that can be used to systematically search for the high-risk vulnerabilities. Once these are located and security measure allocation problem is constructed, we exploit submodularity property to find a suboptimal solution of the problem with guaranteed performance. In particular, we show how to convert security measure allocation problem into a problem of minimizing a linear set function under a submodular constraint. In that case, a polynomial time greedy algorithm can be applied to obtain a suboptimal solution with guaranteed approximation bound. Finally, we demonstrate the applicability of our framework on a control system used for regulation of temperature within a building.

5.1 Introduction

5.1.1 Literature Review

A significant amount of work on allocating security measures was developed for power grid monitoring systems. The grid was modeled as a *static* linear system, and a particular combination of an estimator and an anomaly detector was used. The assumption was in most of the cases that all the sensors are vulnerable, and that all sensors can be protected by deploying some security measure. The security measure allocation problem was then formulated as securing some of the existing, and/or placing additional secured sensors, such as to make undetectable attacks introduced in [30] harder to achieve.

To solve this problem for large number of security vulnerabilities and security measures, which correspond to large number of sensors, many different approaches were taken. Bobba *et al.* proved that it is sufficient to protect the set of so-called basic measurements in order to prevent undetectable attacks, and then used LU decomposition to find these measurements [103]. Kim and Poor approximated the attacker's effort with a linear program, and then used greedy algorithms to select sensors such as to maximize the approximated effort [104]. Dán and Sandberg proposed greedy algorithms that allocate security measures based on the so-called security index [105]. A continuation of this work was presented in [22], where more detailed models of communication network and security measures were introduced. Deka and Vishwanath approached the problem by using graph theoretic methods [106], and considered security measure allocation against any attacker and an attacker constrained in resources. They showed that the first problem can be solved by using Dijkstra's algorithm, while the second one is NP hard and approximate greedy algorithm was proposed. Bi and Zhang transformed the security allocation problem into a variant of the Steiner tree problem, and proposed two algorithms that can solve the problem in exponential time, as well as a polynomial time tree pruning approximation [107]. Liu *et al.* formulated the security allocation problem as a bi-level mixed integer linear program, and proposed decomposition based method to solve the problem [108]. Deng *et al.* assumed that the level of security of a sensor is proportional to the amount of invested resources. The authors then formulated the security allocation problem in two ways – as linear programing and mixed integer linear programing problems [109].

Security measures allocation problem for *dynamical* control systems has attracted less attention. Cardenas *et al.* introduced several ways to estimate the attack impact, and then commented how these attack models can be used to distinguish which sensors/actuators are more urgent to be protected [82]. Teixeira *et al.* proposed a flexible risk model that can be used to allocate security measures [110]. Milošević *et al.* considered a Kalman filtering problem in presence of bias injection attacks, and proposed a way to secure sensors such as to mitigate the impact of these attacks [111].

It is also worth mentioning that other types of allocation problems gained attention within control community recently. Examples include selecting list of leader agents in multi agent systems [112], selecting locations to place actuators [113], and placing sensors in the water distribution network to detect and locate faults [114]. In particular, these works use the submodularity property of the set function in order to obtain suboptimal solution of the problem with known performance guarantees.

We now identify several directions in which the existing literature can be extended. Firstly, the framework for allocating security measures for dynamical control systems is lacking. The methods developed for purposes of power grid monitoring rely heavily on the static model of the power grid, estimator, bad data detector, and attack. Thus, these methods are not straight forward to extend to dynamical models of ICSs, and attack strategies developed for these models. Moreover, the publications on allocating security measures for dynamical control systems [82, 110, 111] do not propose any systematic way to do so once the number of security measures and vulnerabilities is large.

Secondly, the modeling framework proposed in previous publications can be improved. For instance, the connection between the cyber and physical part of the system is often missing. Thus, it is unclear in most of the publications in what way the attacker gains control over the sensors/actuators, and in what way the defender protects the sensors/actuators. The work that partially addressed this issue is [22], but the authors were mostly concerned with models of security vulnerabilities and security measures in the communication network part of the system.

Thirdly, most of the previous publications do not allocate security measures based on a risk model, which is the recommended practice [7–9]. In particular, the resources are allocated mostly relying on the attack impact, but not that much attention is given to how complex the attack is to conduct. We may therefore spend unnecessary amount of resources by preventing attacks that are not likely to happen. A flexible risk model was proposed in [110], but how this method can be used once the number of vulnerabilities and security measures is large was not explained.

Finally, once the criterion based on which we deploy security measures is determined, it is rarely discussed what the quality of the obtained solution is. The security measure allocation problem is a combinatorial problem in nature, and these problems are NP hard in general. Thus, the solution obtained in polynomial time can be arbitrarily far from the optimal value, unless in some special cases.

5.1.2 Contributions

The contributions of this chapter are outlined as follows. As the first contribution, we propose a flexible modeling framework for allocating security measures suitable for dynamical ICSs. We assume the attacker can gain control over sensors and actuators by exploiting a combination of *security vulnerabilities* in the ICS. These security vulnerabilities can model, for instance, unprotected communication links, a control network connected to the Internet without protection, or lack of physical protection of some components. The defender deploys *security measures*, which can model encryption of communication links, installing and maintaining anti-virus software, or improving physical protection.

Security measures are allocated based on a risk model. In particular, we adapt the risk model from [110] to our framework. In this model, each subset of exploited vulnerabilities is a possible attack *scenario*, and its risk is determined based on the *impact* and *complexity* set functions. We remark that the proposed framework is not restricted to any particular instance of these functions. The only requirement we impose is that impact and complexity set functions are non-decreasing with the number of vulnerabilities exploited. The impact set function reflects how dangerous it is if the attacker exploits a given subset of vulnerabilities, and it is estimated based on a model of physical dynamics. The complexity set function reflects how hard it is for the attacker to exploit a given subset of vulnerabilities, which can be determined based on a security expert knowledge [18–20]. The scenarios with low complexity and potentially large impact are defined as high-risk scenarios, and are the most critical to prevent.

Our framework targets the allocation of security measures once the number of vulnerabilities and measures is large. The first challenge that arises due to large number of security vulnerabilities is conducting risk assessment. Given that the number of possible attack scenarios is equal to the number of subsets of the vulnerability set, it is not feasible to go through all the subsets to find those that are critical. To reduce the search space, we use the non-decreasing property of the complexity function, and we prove that it suffices to find the so-called *sufficient representation of minimal cardinality* instead of the whole set of critical scenarios. Furthermore, given that the impact set function can possibly be expensive to evaluate, we exploit the non-decreasing property of this function to reduce the number of its evaluations. To model all the combinations of security vulnerabilities, we use the tree representation proposed in [115] for the purpose of systematic and efficient search. To search through the tree, the breadth first search algorithm that exploits the aforementioned properties is proposed (Algorithm 1), and it is proven that this algorithm returns the sufficient representation of minimal cardinality (Theorem 3), which we outline as the second major contribution.

Once the most critical attack scenarios are selected, the second challenge that arises is how to solve the security measure allocation problem. In our framework, this problem reduces to an integer linear program, which is NP hard to solve in general. However, if this problem can be reformulated as a minimization of linear function under a submodular constraint, it is proven that a polynomial-time greedy heuristic can then be used to provide a suboptimal solution with known performance bound [116]. As a third major contribution, we prove that the security measure allocation problem has submodular structure (Theorem 4), and prove that the set of critical scenarios returned by the breadth first search algorithm provides the best performance guarantees for the greedy algorithm (Theorem 5). Finally, we demonstrate the applicability of our framework on a model of an ICS that is used for regulating temperature within a building [117].

5.1.3 Organization

The remainder of the chapter is organized as follows. In Section 5.2, we describe the modeling framework and the problem formulation. In Section 5.3 we propose an algorithm that systematically searches for critical scenarios to construct security measure allocation problem. In Section 5.4, we connect the problem of security measure allocation with submodularity property. In Section 5.5, we illustrate on an example applicability of our approach. In Section 5.6, we briefly summarize the results presented in the chapter.

5.2 Model Setup and Problem Formulation

In this section, we introduce model setup we use and formulate the security measure allocation problem. In particular, we introduce model of security vulnerabilities within an ICS, model their connection with sensors and actuators, as well as their connection with security measures. In order to distinguish which vulnerabilities/combination of vulnerabilities are the most critical to be prevented, we introduce model of risk. The security measure allocation problem can then be formulated as selecting the least expensive subset of security measures that prevent the combinations of vulnerabilities with the high risk.

5.2.1 Security Vulnerabilities and Security Measures

We denote the security vulnerabilities identified within the system by

$$\mathcal{V} = \{v_1, \ldots, v_{n_v}\}.$$

A security vulnerability $v \in \mathcal{V}$ can model a communication link without protection, lack of anti-virus software on some of the computers in control center, lack of physical protection of some equipment, etc. Throughout the chapter, we refer to the case where the attacker exploits multiple vulnerabilities $\mathcal{V}_a \subseteq \mathcal{V}$ at the same time as a *scenario* $\mathcal{V}_a$.

Remark 6. The model of vulnerabilities in this chapter assumes that vulnerabilities are known. In other words, undiscovered (zero-day) vulnerabilities are not captured with the model. However, since every security strategy needs to be updated over time, newly discovered vulnerabilities can be taken into consideration once a new strategy is deployed.

As it was stated in the previous section, the attacker exploits security vulnerabilities to gain access to sensors and actuators, and in that way endanger the physical world. Let the set of all the sensors and the set of all the actuators be denoted by

$$\mathcal{S} = \{s_1, \ldots, s_{n_y}\} \qquad\qquad \mathcal{A} = \{a_1, \ldots, a_{n_u}\}$$

respectively. With each vulnerability $v \in \mathcal{V}$, we associate the subsets of compromised sensors $\mathcal{S}_v \subseteq \mathcal{S}$ and actuators $\mathcal{A}_v \subseteq \mathcal{A}$. These sets model the components the attacker gain control over once it exploits vulnerability v. In the scenario $\mathcal{V}_a \subseteq \mathcal{V}$ where the attacker exploits multiple security vulnerabilities, the sensors and actuators under its control are

$$\mathcal{S}(\mathcal{V}_a) = \bigcup_{v \in \mathcal{V}_a} \mathcal{S}_v \qquad\qquad \mathcal{A}(\mathcal{V}_a) = \bigcup_{v \in \mathcal{V}_a} \mathcal{A}_v.$$

We assume that the attacker can then freely change the measurements of the sensors $\mathcal{S}(\mathcal{V}_a)$ and the control actions sent to the actuators $\mathcal{A}(\mathcal{V}_a)$.

The security vulnerabilities can be prevented by investing in security measures. We denote the set of security measures by

$$\mathcal{M} = \{m_1, \ldots, m_{n_m}\}.$$

With each security measure $m \in \mathcal{M}$, we associate a number $c_m \in \mathbb{R}^+$ that models the cost of deploying this security measure, and a set of vulnerabilities $\mathcal{V}_m$ prevented by this security measure. In the case when we deploy the subset of security measures $\mathcal{M}_d \subseteq \mathcal{M}$, the set of prevented vulnerabilities $\mathcal{V}(\mathcal{M}_d)$ and the total cost $c(\mathcal{M}_d)$ are defined by

$$\mathcal{V}(\mathcal{M}_d) = \bigcup_{m \in \mathcal{M}_d} \mathcal{V}_m \qquad\qquad c(\mathcal{M}_d) = \sum_{m \in \mathcal{M}_d} c_m. \tag{5.1}$$

The assumption is that the security measures provide perfect prevention of the vulnerabilities $\mathcal{V}(\mathcal{M}_d)$.

Assumption 4. Let $\mathcal{V}(\mathcal{M}_d) \subseteq \mathcal{V}$ be the subset of vulnerabilities prevented by deploying corresponding security measures $\mathcal{M}_d$. The vulnerabilities from the set $\mathcal{V}(\mathcal{M}_d)$ cannot be exploited by an attacker.

Based on Assumption 4, we say that scenario $\mathcal{V}_a$ is *prevented* if it requires at least one of the prevented vulnerabilities from the set $\mathcal{V}(\mathcal{M}_d)$ to be conducted, that is, $\mathcal{V}_a \cap \mathcal{V}(\mathcal{M}_d) \neq \emptyset$. Similarly, we say that a set of attack scenarios $\mathcal{V}_A = \{\mathcal{V}_{a_1}, \ldots, \mathcal{V}_{a_n}\} \subseteq 2^{\mathcal{V}}$ is prevented, if for every $\mathcal{V}_a \in \mathcal{V}_A$ it holds $\mathcal{V}_a \cap \mathcal{V}(\mathcal{M}_d) \neq \emptyset$. We also assume that each security vulnerability can be prevented by deploying a suitable security measure from the set $\mathcal{M}$. This assumption is needed in order to be able to guarantee feasibility of security measure allocation problem.

Assumption 5. Let $v \in \mathcal{V}$ be an arbitrary selected vulnerability. Then there exists at least one security measure $m \in \mathcal{M}$ that prevents this vulnerability, that is, $\mathcal{V}_m \cap v \neq \emptyset$.

5.2.2 Model of Risk

In order to protect an ICS from cyber-attacks, we want to select a subset of security measures $\mathcal{M}_d$ to deploy. As discussed earlier, in most cases the total budget would not be large enough to deploy all the security measures. Thus, we introduce a model of risk based on which we prioritize among attack scenarios. We use the risk model introduced in [118], where it was proposed to model risk as a triplet (Scenario, Impact, Complexity). This model can be applied in our context by defining triplets as

$$\big(\mathcal{V}_a, I(\mathcal{A}(\mathcal{V}_a), \mathcal{S}(\mathcal{V}_a)), \pi(\mathcal{V}_a)\big)$$

where scenarios are represented by the subset of vulnerabilities $\mathcal{V}_a \subseteq \mathcal{V}$ that the attacker exploits in order to conduct a certain attack, $I(\mathcal{A}(\mathcal{V}_a), \mathcal{S}(\mathcal{V}_a))$ represents the impact that occurs in that scenario, and $\pi(\mathcal{V}_a)$ represents the complexity of the scenario.

The impact function $I(\mathcal{A}(\mathcal{V}_a), \mathcal{S}(\mathcal{V}_a))$ should reflect how dangerous it is if the vulnerabilities $\mathcal{V}_a$ are exploited. As we saw in Chapter 3 and Chapter 4, the attack impact can be estimated based on a physical model of the system. We assume that the attacks conducted with more components are more severe then those conducted with less.

Assumption 6. Let $I : 2^{\mathcal{A}} \times 2^{\mathcal{S}} \to \mathbb{R}^+$ be the impact set function. For any $\mathcal{A}_a \subseteq \mathcal{A}_b \subseteq \mathcal{A}$ and $\mathcal{S}_a \subseteq \mathcal{S}_b \subseteq \mathcal{S}$, it holds $I(\mathcal{A}_a, \mathcal{S}_a) \leq I(\mathcal{A}_b, \mathcal{S}_b)$.

The complexity function $\pi(\mathcal{V}_a)$ models how hard it is for the attacker to exploit all the vulnerabilities from $\mathcal{V}_a$ simultaneously. The function is usually estimated based on a security expert knowledge [18–20]. We assume that more vulnerabilities the attacker exploits, the more complex the attack scenario becomes.

Assumption 7. The complexity function $\pi : 2^{\mathcal{V}} \to \mathbb{R}^+$ is a non-decreasing set function, that is, $\pi(\mathcal{V}_a) \leq \pi(\mathcal{V}_b)$ for any $\mathcal{V}_a \subseteq \mathcal{V}_b$.

In the next section, Assumption 6 and Assumption 7 are used to construct algorithm that systematically searches for the most critical attack scenarios. Naturally, the most critical attack scenarios are not complex to conduct, and can lead to large impact. We formally define these scenarios as follows.

Definition 2. A scenario $\mathcal{V}_a \subseteq \mathcal{V}$ is said to be *critical* if $I(\mathcal{V}_a) \geq \bar{I}$ and $\pi(\mathcal{V}_a) \leq \bar{\pi}$, where $\bar{I} \in \mathbb{R}^+$ and $\bar{\pi} \in \mathbb{R}^+$ are some predefined thresholds.

5.2.3 Problem Formulation

Let the set of all critical scenarios be $\mathcal{V}_C$. Our goal is then to find the least expensive set of security measures $\mathcal{M}_d \subseteq \mathcal{M}$ that prevents all the critical scenarios from $\mathcal{V}_C$. This problem can be formulated as an integer linear program, as explained next.

The set of deployed security measures $\mathcal{M}_d$ can be represented as an integer-valued decision vector $x_m \in \{0, 1\}^{n_m}$. In case that we chose to deploy security measure m_i, then $x_m(i)$ is set to 1. Otherwise, $x_m(i)$ equals 0. The objective function we want to minimize can then be modeled as $c^T x_m$, where $c = [c_1 \dots c_{n_m}]^T$ is a vector containing individual costs of security measures.

The constraints for the problem are that the deployed set of security measures x_m should prevent all of the critical scenarios. Thus, we introduce the matrix $R \in \mathbb{R}^{n_v \times n_m}$, which is the incidence matrix modeling which vulnerabilities are prevented by the deployed security measures x_m. The vector of prevented vulnerabilities is denoted with x_v, and equality $x_v = R x_m$ must hold. In case that $x_v(i) > 0$, we know that vulnerability v_i is prevented.

We then join a vector $f_{\mathcal{V}_a} \in \mathbb{R}^{n_v}$ with each of the critical scenarios $\mathcal{V}_a \in \mathcal{V}_C$, defined as

$$f_{\mathcal{V}_a}(i) = \begin{cases} 1 & \text{if } v_i \in \mathcal{V}_a \\ 0 & \text{otherwise} \end{cases} .$$

The condition $f_{\mathcal{V}_a}^T x_v \geq 1$ is then equivalent to perfect prevention of the scenario $\mathcal{V}_a$. The problem of selecting a subset of security measures to deploy can then be defined as the following integer linear program.

Problem 10. (The security measure allocation problem as an integer linear program)

$$\begin{aligned} \underset{x_m}{\text{minimize}} \quad & c^T x_m \\ \text{subject to} \quad & x_v = R x_m \\ & f_{\mathcal{V}_a}^T x_v \geq 1 \qquad \text{for every } \mathcal{V}_a \in \mathcal{V}_C \\ & x_m(i) \in \{0,1\} \qquad i = 1, \ldots, n_m \end{aligned}$$

The two main difficulties with solving this problem are the following. Firstly, constructing Problem 10 represents an issue. In order to form constraints for this problem, we need to find the set of critical scenarios $\mathcal{V}_C$. The number of possible attack scenarios is equal to the number of subsets of the set $\mathcal{V}$. Thus, simply going through all the subsets of $\mathcal{V}$ and deciding whether they are critical or not is not feasible when the cardinality of the set $\mathcal{V}$ is large. Secondly, even if the set $\mathcal{V}_C$ is found, Problem 10 is NP hard in general. Thus, polynomial time algorithms that return the optimal or suboptimal solution are in general unknown, unless some special structure of the problem is identified.

In the next two sections, we tackle these two issues. In Section 5.3, we present an algorithm that systematically generates so-called sufficient representation of minimal cardinality $\tilde{\mathcal{V}}_C$ of the set $\mathcal{V}_C$. In general, the set $\tilde{\mathcal{V}}_C$ is the subset of $\mathcal{V}_C$, with the property that if $\tilde{\mathcal{V}}_C$ is prevented, then we know that $\mathcal{V}_C$ is prevented as well. We show that by using the properties of $\tilde{\mathcal{V}}_C$ and complexity function $\pi(\mathcal{V}_a)$, we can potentially significantly reduce the search space. To resolve the second issue, we use the fact that Problem 10 can be reduced to the problem of minimizing a linear function under a submodular constraint. In that case, a polynomial-time greedy algorithm can be used to find a suboptimal solution of the problem with known performance, as shown in Section 5.4.

5.3 Constructing the Security Measure Allocation Problem

In this section, we address the issue of constructing a set of critical scenarios $\mathcal{V}_C$. As we mentioned, the number of possible attack scenarios in our model is equal to the number of subsets of the set $\mathcal{V}$, that is $2^{|\mathcal{V}|}$. Thus, simply going through all the scenarios and deciding whether it is critical or not is not feasible for large cardinalities of $\mathcal{V}$. In this section, we propose an algorithm that systematically searches for critical scenarios. Before we move to the design of the algorithm, we first explain the rules that this algorithm uses to reduce the execution time of the search.

The first way of reducing the execution time is by reducing the number of scenarios we explore. For this purpose, we use the property of complexity function π introduced in Assumption 7, and we also show that it suffices to find a suitable subset $\tilde{\mathcal{V}}_C$ of the set of critical scenarios, instead of the whole set $\mathcal{V}_C$. In order to further improve the total execution time, the algorithm avoids executing time consuming impact function I for every scenario. In particular, we show how the property of I introduced in Assumption 6 can be exploited to reuse the information from the previously explored scenarios. We then formulate the algorithm that systematically constructs $\tilde{\mathcal{V}}_C$.

5.3.1 Reducing Number of Explored Scenarios

The first way to reduce the number of scenarios to explore is by using the properties of the complexity function. Since the complexity function is non-decreasing, we conclude that if we find a scenario $\mathcal{V}_a$ that has complexity $\pi(\mathcal{V}_a)$ greater than the threshold $\bar{\pi}$, we do not need to investigate any other scenario $\mathcal{V}_b$ that contains this scenario. The reason is that these scenarios have complexity that is greater or equal to $\pi(\mathcal{V}_a)$, hence they do not belong to critical scenarios.

Lemma 8. Assume that a scenario $\mathcal{V}_a$ satisfies $\pi(\mathcal{V}_a) > \bar{\pi}$. Then any scenario $\mathcal{V}_b$, that satisfies $\mathcal{V}_a \subseteq \mathcal{V}_b$ is not a critical scenario.

Proof. From Assumption 7, $\pi(\mathcal{V}_b) \geq \pi(\mathcal{V}_a)$ for $\mathcal{V}_a \subseteq \mathcal{V}_b$. Thus, $\pi(\mathcal{V}_b) > \bar{\pi}$, which implies that scenario $\mathcal{V}_b$ does not satisfy Definition 2. □

The second way to reduce the number of explored scenarios is by observing that we do not need to find the whole set $\mathcal{V}_C$. Instead, it is sufficient to find a suitable subset of this set. We use an example to explain the idea.

Example 5. Assume that we have a set of vulnerabilities $\mathcal{V} = \{v_1, v_2, v_3\}$, and let the set of critical scenarios be given by

$$\mathcal{V}_C = \big\{\{v_1\}, \{v_1, v_3\}, \{v_1, v_2\}, \{v_2, v_3\}\big\}.$$

Consider now the subset $\tilde{\mathcal{V}}_C = \big\{\{v_1\}, \{v_2, v_3\}\big\}$. This subset is prevented if the set of prevented vulnerabilities is one of the following

$$\mathcal{V}(\mathcal{M}_d) = \{v_1, v_2\} \qquad \mathcal{V}(\mathcal{M}_d) = \{v_1, v_3\} \qquad \mathcal{V}(\mathcal{M}_d) = \{v_1, v_2, v_3\}.$$

What is important to realize is that every choice of $\mathcal{V}(\mathcal{M}_d)$ that prevents $\tilde{\mathcal{V}}_C$, prevents $\mathcal{V}_C$ as well. Thus, instead of constructing the complete set of critical scenario $\mathcal{V}_C$, it is suffices to find a subset of smaller cardinality $\tilde{\mathcal{V}}_C$. The motivation for this is that every time we prevent $\tilde{\mathcal{V}}_C$, we also know that $\mathcal{V}_C$ is prevented.

Motivated by the previous example, we introduce a notion of sufficient representation of the set of critical scenarios $\mathcal{V}_C$. In general, the sufficient representation $\tilde{\mathcal{V}}_C$ is a subset of $\mathcal{V}_C$ with the property that once we prevent all the scenarios in $\tilde{\mathcal{V}}_C$, all the critical scenarios $\mathcal{V}_C$ are prevented, as stated in the following definition.

Definition 3. A set $\tilde{\mathcal{V}}_C \subseteq 2^{\mathcal{V}}$ is a *sufficient representation* of a set $\mathcal{V}_C \subseteq 2^{\mathcal{V}}$ if the following two conditions are satisfied

1. $\tilde{\mathcal{V}}_C \subseteq \mathcal{V}_C$;
2. for every set of prevented vulnerabilities $\mathcal{V}(\mathcal{M}_d) \subseteq \mathcal{V}$ it holds that $\mathcal{V}(\mathcal{M}_d)$ prevents $\tilde{\mathcal{V}}_C$ if and only if $\mathcal{V}(\mathcal{M}_d)$ prevents $\mathcal{V}_C$.

Remark 7. Note that a sufficient representation is not unique in general. Furthermore, from the definition it follows that if the set of security measures prevents one sufficient representation of the set $\tilde{\mathcal{V}}_C$, then it also prevents any other sufficient representation $\tilde{\mathcal{V}}'_C$. We use this property in some of the proofs in this section.

Naturally, we are interested in finding a sufficient representation that has minimal cardinality. Besides helping us to reduce the number of scenarios to explore, in the next section we show that a sufficient representation of minimal cardinality is also beneficial for the problem of preventing the critical scenarios using the minimal budget. In the following lemma, we introduce the condition that sufficient representation of minimal cardinality needs to satisfy. In particular, we show that none of the scenarios in this representation should contain any other scenario from the representation. We then prove that the sufficient representation of minimal cardinality is unique.

Lemma 9. Assume that a set $\tilde{\mathcal{V}}_C \subseteq 2^{\mathcal{V}}$ is a sufficient representation of the set of critical scenarios $\mathcal{V}_C$. Then the set $\tilde{\mathcal{V}}_C$ is the sufficient representations of $\mathcal{V}_C$ of *minimal cardinality* if and only if for any two non empty sets $\mathcal{V}_a, \mathcal{V}_b \in \tilde{\mathcal{V}}_C$, $\mathcal{V}_a \neq \mathcal{V}_b$, it holds $\mathcal{V}_a \nsubseteq \mathcal{V}_b$.

Proof. ($\Rightarrow$) We prove that necessity holds by using a contradiction argument. Let $\tilde{\mathcal{V}}_C$ be a sufficient representation of the set $\mathcal{V}_C$ with minimal cardinality, and assume that there exists $\mathcal{V}_a \in \tilde{\mathcal{V}}_C$ and $\mathcal{V}_b \in \tilde{\mathcal{V}}_C$, $\mathcal{V}_a \neq \mathcal{V}_b$, such that $\mathcal{V}_a \subseteq \mathcal{V}_b$. In what follows, we prove that the set $\tilde{\mathcal{V}}'_C = \tilde{\mathcal{V}}_C \setminus \mathcal{V}_b$ is a sufficient representation of $\mathcal{V}_C$, with cardinality smaller than $\tilde{\mathcal{V}}_C$.

Let $\mathcal{V}_P$ be the set of prevented vulnerabilities, and assume that $\tilde{\mathcal{V}}'_C$ is prevented by this set. Given that $\tilde{\mathcal{V}}'_C = \tilde{\mathcal{V}}_C \setminus \mathcal{V}_b$, this implies that $\mathcal{V}_P$ prevents any scenario from $\tilde{\mathcal{V}}_C$, except perhaps $\mathcal{V}_b$. However, since $\mathcal{V}_a \in \tilde{\mathcal{V}}'_C$ we know that $\mathcal{V}_a \cap \mathcal{V}_P \neq \emptyset$, which implies $\mathcal{V}_b \cap \mathcal{V}_P \neq \emptyset$. Hence, we proved that any set $\mathcal{V}_P$ that prevents $\tilde{\mathcal{V}}'_C$ automatically prevents $\tilde{\mathcal{V}}_C$.

Given that $\tilde{\mathcal{V}}'_C \subset \tilde{\mathcal{V}}_C$, any set $\mathcal{V}_P$ that prevents $\tilde{\mathcal{V}}_C$ prevents $\tilde{\mathcal{V}}'_C$ as well. Thus, $\mathcal{V}_P$ prevents $\tilde{\mathcal{V}}_C$ if and only if $\mathcal{V}_P$ prevents $\tilde{\mathcal{V}}'_C$. We then conclude that $\tilde{\mathcal{V}}'_C$ represents a sufficient representation of $\mathcal{V}_C$. Since $|\tilde{\mathcal{V}}'_C| = |\tilde{\mathcal{V}}_C| - 1$, it follows that $\tilde{\mathcal{V}}_C$ is not a sufficient representation of $\mathcal{V}_C$ with minimal cardinality, which is in contradiction with the initial assumption.

($\Leftarrow$) Same as in the case of necessity, we use contradiction argument to prove sufficiency. Let the set $\tilde{\mathcal{V}}_C = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_l\}$ be a sufficient representation of the set $\mathcal{V}_C$, which satisfies $\mathcal{V}_a \not\subseteq \mathcal{V}_b$ for any $\mathcal{V}_a \in \tilde{\mathcal{V}}_C$ and $\mathcal{V}_b \in \tilde{\mathcal{V}}_C$ excepts when $\mathcal{V}_a = \mathcal{V}_b$. Assume now there exists a sufficient representation $\tilde{\mathcal{V}}'_C = \{\mathcal{V}'_1, \mathcal{V}'_2, \ldots, \mathcal{V}'_m\}$ of $\mathcal{V}_C$ with $|\tilde{\mathcal{V}}'_C| < |\tilde{\mathcal{V}}_C|$. In that case, there has to be at least one scenario $\mathcal{V}_n$ that satisfies $\mathcal{V}_n \in \tilde{\mathcal{V}}_C$ and $\mathcal{V}_n \notin \tilde{\mathcal{V}}'_C$. In what follows, we prove that $\mathcal{V}_n$ does not exists.

Assume first that for any $\mathcal{V}'_i \in \tilde{\mathcal{V}}'_C$ we have $\mathcal{V}'_i \setminus \mathcal{V}_n \neq \emptyset$. If we choose prevented vulnerabilities to be $\mathcal{V}_P = \{v_{p_1}, \ldots, v_{p_m}\}$, where $v_{p_i} \in \mathcal{V}'_i \setminus \mathcal{V}_n$, we see that $\mathcal{V}_P$ prevents $\tilde{\mathcal{V}}'_C$. However, this set does not prevent $\tilde{\mathcal{V}}_C$ since it does not prevent $\mathcal{V}_n$. This is inconsistent with the fact that both $\tilde{\mathcal{V}}_C$ and $\tilde{\mathcal{V}}'_C$ are sufficient representations of $\mathcal{V}_C$, since any $\mathcal{V}_P$ that prevents one sufficient representation, needs to prevent any other as well.

Therefore, there has to be at least one set $\mathcal{V}'_n \in \tilde{\mathcal{V}}'_C$ such that $\mathcal{V}'_n \subset \mathcal{V}_n$. Let the set of prevented vulnerabilities be now $\mathcal{V}_P = \{v_{p_1}, \ldots, v_{p_l}\}$, where $v_{p_i} \in \mathcal{V}_i \setminus \mathcal{V}_n$ for $\mathcal{V}_i \in \tilde{\mathcal{V}}_C, \mathcal{V}_i \neq \mathcal{V}_n$ and $v_{p_n} \in \mathcal{V}_n \setminus \mathcal{V}'_n$. Note that $\mathcal{V}_i \setminus \mathcal{V}_n$ is always non-empty, since $\mathcal{V}_i \not\subseteq \mathcal{V}_n$ by the assumption. This set prevents $\tilde{\mathcal{V}}_C$. However, this set does not prevent $\tilde{\mathcal{V}}'_C$ since it does not prevent $\mathcal{V}'_n$, which is again in contradiction with the fact that both $\tilde{\mathcal{V}}_C$ and $\tilde{\mathcal{V}}'_C$ are sufficient representations of $\mathcal{V}_C$. Thus, we conclude that the set $\mathcal{V}_n$ does not exist, which contradicts the assumption $|\tilde{\mathcal{V}}'_C| < |\tilde{\mathcal{V}}_C|$. □

Lemma 10. The sufficient representation of the minimal cardinality of the set $\mathcal{V}_C$ is unique.

Proof. Assume that the statement is not true, and that $\tilde{\mathcal{V}}_C = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_l\}$ and $\tilde{\mathcal{V}}'_C = \{\mathcal{V}'_1, \mathcal{V}'_2, \ldots, \mathcal{V}'_l\}$ are the two different sufficient representations of minimal cardinality. Then there exists $\mathcal{V}_j$ that satisfies $\mathcal{V}_j \in \tilde{\mathcal{V}}_C$ and $\mathcal{V}_j \notin \tilde{\mathcal{V}}'_C$. The first possibility is that for all $\mathcal{V}'_i \in \tilde{\mathcal{V}}'_C$ we have $\mathcal{V}'_i \setminus \mathcal{V}_j \neq \emptyset$. If we select $\mathcal{V}_p = \{v_{p_1}, \ldots, v_{p_l}\}$, where $v_{p_i} \in \mathcal{V}'_i \setminus \mathcal{V}_j$, then $\mathcal{V}_P$ prevents $\tilde{\mathcal{V}}'_C$. However, $\mathcal{V}_P$ does not prevent $\mathcal{V}_j$, so $\tilde{\mathcal{V}}_C$ is not prevented. This contradicts the fact that both $\tilde{\mathcal{V}}_C$ and $\tilde{\mathcal{V}}'_C$ are the sufficient representations. The remaining possibility is that there exists $\mathcal{V}'_t \in \tilde{\mathcal{V}}'_C$ such that $\mathcal{V}'_t \subset \mathcal{V}_j$. Let $v_{p_i} \in \mathcal{V}_i \setminus \mathcal{V}_j$ for $i \neq j$ and $v_{p_j} \in \mathcal{V}_j \setminus \mathcal{V}'_t$. The set $\mathcal{V}_P = \{v_{p_1}, \ldots, v_{p_l}\}$ prevents $\tilde{\mathcal{V}}_C$, but not $\tilde{\mathcal{V}}'_C$ since it does not prevent $\mathcal{V}'_t$. This is again in contradiction with the fact that both $\tilde{\mathcal{V}}_C$ and $\tilde{\mathcal{V}}'_C$ are sufficient representations of $\mathcal{V}_C$. Thus, $\mathcal{V}_j$ does not exist. □

5.3.2 Reducing Number of Executions of Impact Set Function

As we stated, impact set function is estimated based on a physical model of the system. Thus, this function can become expensive to calculate for models of large dimension. Therefore, it is desirable to reduce the number of executions of impact function as much as possible. One way to do this would be to store the combinations of sensors and actuators for which we have already evaluated the impact function.

These combinations can then be further divided into two lists, which we denote with $\mathcal{C}_+$ and $\mathcal{C}_-$. These lists could be updated on-line, but also in a preprocessing step based on some priori knowledge.

The list $\mathcal{C}_+$ contains combinations of sensors and actuators that lead to the impact greater or equal to $\bar{I}$. For instance, say that the list contains combination $(\mathcal{S}_a, \mathcal{A}_a)$, and we need to check if the impact $I(\mathcal{S}(\mathcal{V}_a), \mathcal{A}(\mathcal{V}_a))$ for some scenario $\mathcal{V}_a$ is greater than $\bar{I}$. In case that $\mathcal{S}_a \subseteq \mathcal{S}(\mathcal{V}_a)$ and $\mathcal{A}_a \subseteq \mathcal{A}(\mathcal{V}_a)$, then we know from Assumption 6 that $I(\mathcal{S}(\mathcal{V}_a), \mathcal{A}(\mathcal{V}_a)) \geq \bar{I}$. In other words, if the combination $(\mathcal{S}(\mathcal{V}_a), \mathcal{A}(\mathcal{V}_a))$ for which we need to calculate impact contains any of the scenarios from $\mathcal{C}_+$, we know that it leads to impact greater than $\bar{I}$.

The list $\mathcal{C}_-$ contains those combinations that lead to impact less than $\bar{I}$. Let the combination $(\mathcal{S}_a, \mathcal{A}_a)$ be the element of $\mathcal{C}_-$. In case that $\mathcal{S}(\mathcal{V}_a) \subseteq \mathcal{S}_a$ and $\mathcal{A}(\mathcal{V}_a) \subseteq \mathcal{A}_a$, then it follows from Assumption 6 that $I(\mathcal{S}(\mathcal{V}_a), \mathcal{A}(\mathcal{V}_a)) \leq \bar{I}$. Thus, if a combination for which we need to calculate impact is the subset of a combination from $\mathcal{C}_-$, then it follows that this combination has impact less than $\bar{I}$.

Remark 8. Since the number of combinations of sensors and actuators investigated rapidly grows, we can store in $\mathcal{C}_+$ only those combinations that contain relatively small number of sensors and actuators and lead to impact equal or greater than $\bar{I}$. Similarly, in $\mathcal{C}_-$, only those combinations with a large number of sensors and actuators that lead to the impact less than $\bar{I}$ can be stored.

5.3.3 Algorithm for Constructing the Sufficient Representation of Minimal Cardinality

In the following, we introduce a systematic way to find the sufficient representation of minimal cardinality $\tilde{\mathcal{V}}_C$. We begin with introducing a set enumeration tree of the power set $2^{\mathcal{V}}$ shown in Figure 5.1. This representation was introduced in [115] for the purposes of systematically searching through the power set.

In our case, each node of the tree represents one attack scenario. The tree has $|\mathcal{V}| + 1$ layers enumerated with $p = 0, 1, \ldots, |\mathcal{V}|$, where each layer p of the tree only contains subsets of $\mathcal{V}$ with the cardinality p. Another important property of the tree is that the connections are based on the following two principles. Firstly, the node $\mathcal{V}_a$ in layer p can be connected only to nodes $\mathcal{V}_a \cup v_j$, $v_j \notin \mathcal{V}_a$ in layer $p + 1$. Secondly, the node $\mathcal{V}_a$ is connected to node $\mathcal{V}_a \cup v_j$, only if there exists $v_i \in \mathcal{V}_a$ such that $j < i$. For instance, in the case of the graph shown Figure 5.1, the node $\{2\}$ is connected to $\{1, 2\}$, but not to $\{2, 3\}$.

In order to explore the tree for the set $\tilde{\mathcal{V}}_C$, we adapt a *Breadth first search* algorithm [119]. This algorithm explores the tree by layers, that is, it does not move to the next layer before all the scenarios from the current layer are explored. For each scenario in the current layer, the algorithm performs *classification* of that scenario. Once all the scenarios are classified, the algorithm *generates new scenarios* that will be searched in the next layer. Once there is no more scenarios

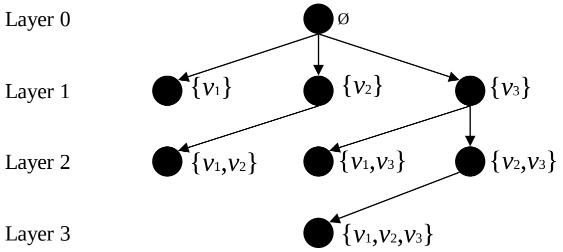


Figure 5.1: Tree representation of the power set of the set $\mathcal{V} = \{v_1, v_2, v_3\}$.

to be explored, the algorithm terminates. In the following, we explain how the classification and generation actions are performed.

Classification of Scenarios

The algorithm keeps the list of scenarios to be explored in the current layer, which is denoted with $\mathcal{V}_L$. For every scenario from the list $\mathcal{V}_L$, the algorithm first checks if the scenarios has complexity greater than $\bar{\pi}$. If it has, the algorithm moves to the next scenario. Otherwise, the algorithm checks if the impact of the scenario is greater than $\bar{I}$. For this purpose, the algorithm first checks the lists $\mathcal{C}_+$ and $\mathcal{C}_-$ that were introduced in Subsection 5.3.2. In case that it is not possible to determine whether or not impact of the scenario is greater than $\bar{I}$ based on the lists, we calculate the impact. We also indicate that the impact was calculated for this combination, in order to update the lists $\mathcal{C}_+$ and $\mathcal{C}_-$ later on.

Based on the result of impact calculation, the scenario is classified in one of the following two categories. The first category are critical scenarios that we want to find, and these scenarios are stored in the list $\tilde{\mathcal{V}}_C$. In the case that scenario is not critical, it is stored in the set that we denote with $\mathcal{V}_O$. By adding an additional vulnerability to the scenarios from $\mathcal{V}_O$ in the next layer, we can obtain the critical scenarios. Once the classification of scenarios is finished, the algorithm empties the list $\mathcal{V}_L$.

The important observation here is that classification of scenarios within a layer can be performed independently for each scenario. Thus if we have N cores available, classification step can be executed in parallel, and in that way reduce the execution time of this step N times. Since the classification step involves evaluating impact and complexity function large number of times, which is expected to be the most time consuming action in the algorithm, significant reduction in execution time can be made with parallelization.

Once the classification step is finished, the lists $\mathcal{C}_+$ and $\mathcal{C}_-$ are then updated with newly checked combination of sensors and actuators, and the algorithm moves to the generation step. The reason why we do not update the lists within classification step is in order to be able to execute this step in parallel.

Generating New Scenarios

Once the classification of the scenarios in the current layer is performed, the algorithm generates scenarios that are searched in the next layer. In order to reduce the number of scenarios to check, the algorithm relies on the property of the complexity function $\pi(\mathcal{V}_a)$ introduced in Lemma 8, and the property of the sufficient representation of the minimal cardinality introduced in Lemma 9. Mainly, if the algorithm classifies some scenario $\mathcal{V}_a$ as a critical scenario, it does not need to generate any other scenarios that contain $\mathcal{V}_a$, since the sufficient representation of minimal cardinality does not contain these. Similarly, when some scenario $\mathcal{V}_a$ has complexity greater than $\bar{\pi}$, the algorithm does not need to generate any other new scenarios that contain $\mathcal{V}_a$, since these scenarios have complexity greater than $\bar{\pi}$. In other words, every time we find a critical scenario or a scenario with complexity greater than $\bar{\pi}$, we eliminate the branch of the tree starting from this scenario. Therefore, we should only generate new scenarios from the set $\mathcal{V}_O$. In this way, we potentially achieve significant reduction in the number of scenarios we need to search through in the next layer, and the layers that follow.

The generation of scenarios is performed according to the following rules. Firstly, for each scenario $\mathcal{V}_a$ from the list $\mathcal{V}_O$, the algorithm adds a scenario $v_i \cup \mathcal{V}_a$ to the list $\mathcal{V}_L$, only if there does not exists $v_j \in \mathcal{V}_a$, such that $i < j$. This rule follows from the specific tree structure. However, from the tree structure, it also follows that a scenario in the layer $p + 1$ can contain a critical scenario from the previous layer, although the branch starting from that critical scenario was eliminated. Thus, in order to obtain a sufficient representation of minimal cardinality, the second rule is to add $v_i \cup \mathcal{V}_a$ to the list $\mathcal{V}_L$ only if there does not exist a critical scenario in the list $\tilde{\mathcal{V}}_C$ that is contained in $v_i \cup \mathcal{V}_a$. Once the new scenarios are generated for each of the elements in $\mathcal{V}_O$, $\mathcal{V}_O$ is reset to $\emptyset$.

Algorithm

Based on the previous discussion, we formulate Algorithm 1 for constructing the sufficient representation of minimal cardinality.

It is important to realize that in general, the algorithm may end up searching every subset of the set $\mathcal{V}$. Thus, based on the available time, the search can be restricted to only the first $\bar{p}$ layers. In that case,the algorithm searches for all the scenarios up to cardinality $\bar{p}$, which is $O(n_v^{\bar{p}})$. However, it is expected that this number would be significantly reduced, since we do not need to generate new scenarios for every scenario with complexity greater than $\bar{\pi}$ and every critical scenario.

We conclude this section by formally proving that the algorithm returns sufficient representation of the minimal cardinality. In case that the algorithm is restricted to search the first $\bar{p}$ layers, the algorithm returns the sufficient representation of the minimal cardinality for the set $\mathcal{V}_C^{\bar{p}}$, which contains all the critical scenarios with cardinality less or equal to $\bar{p}$.

Algorithm 1 Finding the Sufficient Representation of Minimal Cardinality

```
Input: $\mathcal{V}$, $\bar{\pi}$, $\bar{I}$, $\pi$, $I$
Output: $\tilde{\mathcal{V}}_C$
$\tilde{\mathcal{V}}_C \leftarrow \emptyset$;   % The sufficient representation of minimal cardinality
$\mathcal{V}_L \leftarrow \emptyset$;   % The list of scenarios to be searched in the current layer
$\mathcal{V}_O \leftarrow \emptyset$;   % The list of scenarios that are used for generating new scenarios
$\mathcal{C}_+, \mathcal{C}_- \leftarrow \emptyset$; % The lists in which we store explored combinations of sensors and actuators
% Initialize the list $\mathcal{V}_L$ with scenarios containing one vulnerability
for $i = 1:1:n_v$ do
    Add scenario $\{v_i\}$ to the bottom of the list $\mathcal{V}_L$;
end for
while $\mathcal{V}_L \neq \emptyset$ do
    % Classification of scenarios that is executed in parallel on $N$ cores
    for every scenario $\mathcal{V}_a$ in $\mathcal{V}_L$ do
        if $\pi(\mathcal{V}_a) \leq \bar{\pi}$ then
            determine if $I(\mathcal{S}(\mathcal{V}_a), \mathcal{A}(\mathcal{V}_a))$ is greater than $\bar{I}$, either using lists $\mathcal{C}_+$/ $\mathcal{C}_-$ or by evaluating $I$
            indicate if impact was calculated by evaluating impact function $I$
            if $I(\mathcal{S}(\mathcal{V}_a), \mathcal{A}(\mathcal{V}_a)) \geq \bar{I}$ then
                add the scenario to the bottom of the list $\tilde{\mathcal{V}}_C$;
            else
                add the scenario to the bottom of the list $\mathcal{V}_O$;
            end if
        end if
    end for
    empty $\mathcal{V}_L$
    update lists $\mathcal{C}_+$ and $\mathcal{C}_-$ with combinations $(\mathcal{S}(\mathcal{V}_a), \mathcal{A}(\mathcal{V}_a))$ for which the impact was evaluated
    % Generation of new scenarios
    for every scenario $\mathcal{V}_a$ in $\mathcal{V}_O$ do
        find a vulnerability with minimal index $j$ in $\mathcal{V}_a$
        for $i = 1:1:j$ do
            if there is no scenario from $\tilde{\mathcal{V}}_C$ contained in $\{\mathcal{V}_a \cup v_i\}$ then
                add scenario $\{\mathcal{V}_a \cup v_i\}$ to the bottom of the list $\mathcal{V}_L$;
            end if
        end for
    end for
    empty $\mathcal{V}_O$
end while
```

Theorem 3. The Algorithm 1 finds the sufficient representation of minimal cardinality $\tilde{\mathcal{V}}_C$.

Proof. We first show that the algorithm forms the list $\tilde{\mathcal{V}}_C$ that contains the sufficient representation of minimal cardinality using an induction argument. We start by proving that the claim holds for all the scenarios of cardinality $p = 1$. Since the list $\mathcal{V}_L$ is initialized with all the scenarios of the first layer, all of the critical scenarios among these are for sure found, and added to the list $\tilde{\mathcal{V}}_C$, so the claim holds for the first layer.

Suppose now that the algorithm reaches the layer p with a given list $\tilde{\mathcal{V}}_C$. Assume that the list contains all the scenarios with the cardinality less or equal then p that are contained in the sufficient representation. In the layer $p + 1$, all the critical scenarios among generated scenarios are added to the $\tilde{\mathcal{V}}_C$. The scenarios from the layer $p + 1$ that are not generated fall in one of the two categories. Either they are the scenarios that contain some of the critical scenarios that are added to the list $\tilde{\mathcal{V}}_C$, or they contain the scenarios with complexity greater than $\bar{\pi}$. Based on Lemma 8 and Lemma 9, we know these scenarios do not belong to the sufficient representation of minimal cardinality. Thus, the claim holds for the layer $p + 1$ as well, so the list $\tilde{\mathcal{V}}_C$ contains all the scenarios from the sufficient representation of minimal cardinality.

It remains to be proven that the list $\tilde{\mathcal{V}}_C$ that was obtained does not contain any two scenarios $\mathcal{V}_a \neq \mathcal{V}_b$ such that $\mathcal{V}_a \nsubseteq \mathcal{V}_b$. Firstly, for the two scenarios $\mathcal{V}_a \neq \mathcal{V}_b$ from the same layer, the condition $\mathcal{V}_a \nsubseteq \mathcal{V}_b$ cannot be satisfied. On the other hand, once we generate scenarios to be explored in the layer p, we always check if these contain any other scenarios from the lower layers previously added to $\tilde{\mathcal{V}}_C$ (Line 30). Thus, the overlaps are not possible either for scenarios from different layers, so we conclude that $\tilde{\mathcal{V}}_C$ represents a sufficient representation of minimal cardinality. □

Corollary 1. If we restrict Algorithm 1 to search the first $\bar{p}$ layers, the sufficient representation of the set $\mathcal{V}_C^{\bar{p}}$ is returned.

Proof. To prove the claim, we introduce the complexity function that is defined as

$$\pi'(\mathcal{V}_a) = \begin{cases} \pi(\mathcal{V}_a) & \text{if } |\mathcal{V}_a| \leq \bar{p} \\ \max\{\pi(\mathcal{V}_a), \bar{\pi}'\} & \text{if } |\mathcal{V}_a| > \bar{p} \end{cases}$$

where $\bar{\pi}' > \bar{\pi}$. This function satisfies Assumption 7 and represents a possible candidate for the complexity function. For a scenario with the cardinality less than $\bar{p}$, the function $\pi'(\mathcal{V}_a)$ equals to $\pi(\mathcal{V}_a)$. Thus, the sets of critical scenarios with the cardinalities less or equal to $\bar{p}$ are the same for both $\pi'(\mathcal{V}_a)$ and $\pi(\mathcal{V}_a)$. However, the scenarios with the cardinality greater than $\bar{p}$ are with complexity larger than $\bar{\pi}$ if the function $\pi'(\mathcal{V}_a)$ is used. In that case, scenarios with cardinality greater than $\bar{p}$ cannot be critical based on Lemma 8, which implies that the set of critical scenarios is equal to $\mathcal{V}_C^{\bar{p}}$. Therefore, if we apply Algorithm 1 with $\pi'(\mathcal{V}_a)$ as a complexity function, then it follows from Theorem 3 that the sufficient representation of minimal cardinality for the set $\mathcal{V}_C^{\bar{p}}$ is returned. □

5.4 Solving the Security Measure Allocation Problem

In this section, we address the problem of finding the subset of security measures $\mathcal{M}_d \subseteq \mathcal{M}$ of minimal cost that prevents all the critical scenarios from $\tilde{\mathcal{V}}_C$. The first important result of this section is to show that this problem can be casted as a minimization of linear function under a submodular constraint. In that case, a polynomial time greedy algorithm can be used to find a suboptimal solution with a guaranteed performance bounds. The second important result of this section is to show that the greedy algorithm gives the best performance guarantees on the solution once the sufficient representation of minimal cardinality $\tilde{\mathcal{V}}_C$ is used to represent the set of critical scenarios $\mathcal{V}_C$. In case that the algorithm was stopped due to time constraints after $\bar{p}$ layers, the same results hold, but for $\mathcal{V}_C^{\bar{p}}$ and $\tilde{\mathcal{V}}_C^{\bar{p}}$. We first introduce necessary theoretical background.

5.4.1 Submodular Optimization Problems

In this section, we revisit some results concerning the submodularity property. Submodularity is often refereed to as a diminishing returns property of a set function. In other words, if some set function is submodular, adding an element to a smaller set will result in a larger gain than adding one to a larger set containing that set.

Definition 4. Let $\mathcal{V} = \{v_1, ..., v_n\}$ be a finite non-empty set. A set function

$$F : 2^{\mathcal{V}} \to \mathbb{R}$$

is said to be submodular if for all $\mathcal{V}_a \subseteq \mathcal{V}_b \subseteq \mathcal{V}$ and $v \notin \mathcal{V}_b$, we have

$$F(\mathcal{V}_a \cup v) - F(\mathcal{V}_a) \geq F(\mathcal{V}_b \cup v) - F(\mathcal{V}_b).$$

We also recall the definition of non-decreasing set function.

Definition 5. Let $\mathcal{V} = \{v_1, ..., v_n\}$ be a finite non-empty set. A set function

$$F : 2^{\mathcal{V}} \to \mathbb{R}$$

is said to be nondecreasing if for all $\mathcal{V}_a \subseteq \mathcal{V}_b \subseteq \mathcal{V}$, we have $F(\mathcal{V}_a) \leq F(\mathcal{V}_b)$.

The submodularity property is preserved under many operations. In particular, we will use the following one.

Lemma 11. Let $F_1, F_2, ..., F_N$ be submodular set functions defined on a finite set $\mathcal{V}$ and let $c_1, ..., c_N \in \mathbb{R}^+$. The function

$$F(\mathcal{V}_a) = \sum_{i=1}^{N} c_i F_i(\mathcal{V}_a)$$

is submodular.

Proof. We refer the reader to [120, Proposition 2.7]. □

Similar to convexity in continuous optimization problems, the submodularity property plays an important role in combinatorial optimization. Mainly, certain classes of combinatorial optimization problems that have a submodular structure can be approximately solved with off-line performance guarantees using polynomial time greedy algorithms. The problem relevant for our investigation is minimizations of a linear function subject to a submodular constraint, that can be defined as follows.

Problem 11. (Minimization of linear function under a submodular constraint)

$$\begin{aligned} \underset{\mathcal{V}_a \subseteq \mathcal{V}}{\text{minimize}} \quad & b(\mathcal{V}_a) = \sum_{v \in \mathcal{V}_a} b_v \\ \text{subject to} \quad & F(\mathcal{V}_a) \geq \bar{F} \end{aligned}$$

where $b_v \in \mathbb{R}^+$ are costs, function $F : 2^{\mathcal{V}} \to \mathbb{Z}$ is submodular nondecreasing set function that takes only integer values, and $\bar{F} \in \mathbb{R}$.

It is known that this problem can be approximately solved by Algorithm 2, with the performance guarantees given in Lemma 12.

Algorithm 2 A greedy heuristics for Problem 11 [116]

```
Input: Set $\mathcal{V}$, set function $F$, costs $b_i$;
Output: Subset $\mathcal{V}_G \subseteq \mathcal{V}$.
$\mathcal{V}_G \leftarrow \emptyset$;
while $F(\mathcal{V}_G) < \bar{F}$ do
    $v^* \leftarrow \text{argmin}\{\frac{b_v}{F(\mathcal{V}_G \cup \{v\}) - F(\mathcal{V}_G)} : v \in \mathcal{V} \setminus \mathcal{V}_G\}$;
    $\mathcal{V}_G \leftarrow \mathcal{V}_G \cup v^*$;
end while
```

Lemma 12. Let F be a nondecreasing, submodular, and integer valued set function with $F(\emptyset) = 0$. Consider Problem 11, and denote by $b(\mathcal{V}_O)$ the optimal value, and by $b(\mathcal{V}_G)$ the value found by Algorithm 2. Then

$$b(\mathcal{V}_G) \leq b(\mathcal{V}_O) H\Big(\max_{v \in \mathcal{V}} F(\{v\}) - F(\emptyset)\Big) \tag{5.2}$$

where $H(d) = \sum_{i=1}^{d} \frac{1}{i}$.

Proof. We refer the reader to [116, Theorem 1]. □

Remark 9. Note that the bound (5.2) is dependent on the function G. In particular, the bound grows logarithmically in the value of $\max_{m \in \mathcal{M}} G(m)$, so the performance guarantees remain relatively good even for large values of $G(m)$. Furthermore, this bound represents the worst case performance guarantees. Algorithm 2 can perform much better in practice.

5.4.2 Submodular Structure of the Security Measure Allocation Problem

In what follows, we prove that the security measure allocation problem is an instance of Problem 11. In that case, we can use polynomial time Algorithm 2 for finding an approximate solution with guaranteed performance.

We begin with introducing mathematical formulation for the process of scenario prevention. Recall that $\mathcal{M}_d$ represents the set of security measures we want to choose, and that $\mathcal{V}(\mathcal{M}_d)$ is the subset of vulnerabilities prevented by the security measures $\mathcal{M}_d$. From Assumption 4, it follows that all the attack scenarios that exploit vulnerabilities from the set $\mathcal{V}(\mathcal{M}_d)$ becomes prevented. In order to model this relation, with each scenario $\mathcal{V}_a \in \tilde{\mathcal{V}}_C$, we define a gain function

$$g_a(\mathcal{M}_d) = \min\{|\mathcal{V}(\mathcal{M}_d) \cap \mathcal{V}_a|, 1\}.$$

In other words, in case that scenario $\mathcal{V}_a$ requires any of the vulnerabilities from the set of prevented vulnerabilities $\mathcal{V}(\mathcal{M}_d)$ to be conducted, it is prevented, and $g_a(\mathcal{M}_d)$ returns 1. Otherwise, we do not prevent $\mathcal{V}_a$, so $g_a(\mathcal{M}_d)$ returns 0. The global gain function can then be defined as

$$G(\mathcal{M}_d) = \sum_{\mathcal{V}_a \in \tilde{\mathcal{V}}_C} g_a(\mathcal{M}_d) = \sum_{\mathcal{V}_a \in \tilde{\mathcal{V}}_C} \min\{|\mathcal{V}(\mathcal{M}_d) \cap \mathcal{V}_a|, 1\}.$$

The problem of finding the least expensive $\mathcal{M}_d$ such as to prevent all the critical scenarios can then be formulated as follows.

Problem 12. (Security Measure Allocation)

$$\begin{aligned} \underset{\mathcal{M}_d \subseteq \mathcal{M}}{\text{minimize}} \quad & c(\mathcal{M}_d) = \sum_{m \in \mathcal{M}_d} c_m \\ \text{subject to} \quad & G(\mathcal{M}_d) = |\tilde{\mathcal{V}}_C|. \end{aligned}$$

The objective function $c(\mathcal{M}_d)$ we are trying to minimize represents the total cost of deployed security measures $\mathcal{M}_d$. The constraint $G(\mathcal{M}_d) = |\tilde{\mathcal{V}}_C|$ comes from the fact that $G(\mathcal{M}_d)$ equals to $|\tilde{\mathcal{V}}_C|$ once all of the scenarios from $\tilde{\mathcal{V}}_C$ are prevented. Recall that this automatically implies that all the critical scenarios $\mathcal{V}_C$ are prevented, since $\tilde{\mathcal{V}}_C$ represents the sufficient representation of $\mathcal{V}_C$. We now prove the first important result of this section, which is that Problem 10 has the same submodular structure as Problem 11. That implies that we can use Algorithm 2 to find the solution with the performance guarantees given in Lemma 12.

Theorem 4. Problem 12 is an instance of Problem 11.

Proof. In order to show that this claim holds, we prove that the function $G(\mathcal{M}_d)$ satisfies the conditions stated in Lemma 12.

Submodularity. It suffices to show that $g_a(\mathcal{M}_d)$ is submodular, since submodularity is preserved under a nonnegative sum (Lemma 11). We prove that $g_a(\mathcal{M}_d)$ is submodular by using the definition of submodularity, and contradiction argument. Let

$$\Delta_m(\mathcal{M}_d) = g_a(\mathcal{M}_d \cup m) - g_a(\mathcal{M}_d)$$

be the gain we achieve by adding the security measure $m \in \mathcal{M}$ to the set of already deployed security measures $\mathcal{M}_d$. We first show that this gain could be either 0 or 1. The gain equals zero in two situations. The first situation is when the scenario $\mathcal{V}_a$ is already prevented by the deployed security measures $\mathcal{M}_d$. We then have

$$g_a(\mathcal{M}_d) = g_a(\mathcal{M}_d \cup m) = 1.$$

Thus $\Delta_m(\mathcal{M}_d) = 0$. The second situation occurs when $\mathcal{M}_d \cup m$ do not prevent the scenario $\mathcal{V}_a$ from happening. We then have $g_a(\mathcal{M}_d) = g_a(\mathcal{M}_d \cup m) = 0$, hence $\Delta_m(\mathcal{M}_d) = 0$.

The second value $\Delta_m(\mathcal{M}_d) = 1$ is achieved when scenario $\mathcal{V}_a$ is prevented by security measure m, but not the security measures $\mathcal{M}_d$. In that case, $g_a(\mathcal{M}_d) = 0$ and $g_a(\mathcal{M}_d \cup m) = 1$, so $\Delta_m(\mathcal{M}_d) = 1$. Based on the previous discussion, it follows

$$\Delta_m(\mathcal{M}_d) = \begin{cases} 1 & \mathcal{V}_m \cap \mathcal{V}_a \neq \emptyset \text{ and } \mathcal{V}(\mathcal{M}_d) \cap \mathcal{V}_a = \emptyset \\ 0 & \text{otherwise} \end{cases} . \tag{5.3}$$

Assume now that $g_a(\mathcal{M}_d)$ is not submodular. In that case, there exists $\mathcal{M}_a \subseteq \mathcal{M}_b$ and $m \notin \mathcal{M}_b$ such that $\Delta_m(\mathcal{M}_a) = 0$ and $\Delta_m(\mathcal{M}_b) = 1$. From (5.3), it follows that $\mathcal{V}(\mathcal{M}_b) \cap \mathcal{V}_a = \emptyset$ and $\mathcal{V}_m \cap \mathcal{V}_a \neq \emptyset$. However, since $\mathcal{V}(\mathcal{M}_a) \subseteq \mathcal{V}(\mathcal{M}_b)$, we have $\mathcal{V}(\mathcal{M}_a) \cap \mathcal{V}_a = \emptyset$. But then it follows from (5.3) that $\Delta_m(\mathcal{M}_a)$ has to be equal to 1. This contradicts the initial assumption, so it follows that $g_a(\mathcal{M}_d)$ is submodular.

Nondecreasing property. Let $\mathcal{M}_a \subseteq \mathcal{M}_b \subseteq \mathcal{M}$. From (5.1), we have $|\mathcal{V}(\mathcal{M}_a) \cap \mathcal{V}_a| \leq |\mathcal{V}(\mathcal{M}_b) \cap \mathcal{V}_a|$ for any $\mathcal{V}_a$, which implies that $g_a(\mathcal{M}_d)$ is non-decreasing. Thus, $G(\mathcal{M}_d)$ is non-decreasing as well, as a non-negative sum of non-decreasing set functions.

Integer valued. Since each of the functions $g_a(\mathcal{M}_d)$ can take only values 0 or 1, $G(\mathcal{M}_d)$ is integer valued set function. □

Remark 10. In the case when each security measure prevents exactly one vulnerability, Problem 12 becomes a hitting set problem. In case that only the vulnerabilities from the first layer are included in the sufficient representation of minimal cardinality, the problem becomes a set cover problem. Both of the aforementioned problems are known to be NP complete [121].

We now introduce the second important result of this section, which is to show the benefit of using $\tilde{\mathcal{V}}_C$ compared with other sufficient representations of the set $\mathcal{V}_C$. Note that the function G is dependent on the set $\tilde{\mathcal{V}}_C$, and note that guarantees on performance stated in Lemma 12 are dependent on the set function used in the constraint. We now prove that sufficient representation of the minimal cardinality $\tilde{\mathcal{V}}_C$ provides the tightest guarantees on performance among all the sufficient representations.

Theorem 5. Let $\tilde{\mathcal{V}}_C$ be the sufficient representation of minimal cardinality of the set of critical scenarios $\mathcal{V}_C$, and let $\tilde{\mathcal{V}}'_C$ be any other sufficient representation. Then the set $\tilde{\mathcal{V}}_C$ guarantees the best performance in terms of the bound (5.2), that is,

$$H\big(\max_{m\in\mathcal{M}} G(m)\big) \leq H\big(\max_{m\in\mathcal{M}} G'(m)\big)$$

where

$$G(\mathcal{M}_d) = \sum_{\mathcal{V}_a\in\tilde{\mathcal{V}}_C} \min\{|\mathcal{V}(\mathcal{M}_d)\cap\mathcal{V}_a|, 1\} \qquad G'(\mathcal{M}_d) = \sum_{\mathcal{V}_a\in\tilde{\mathcal{V}}'_C} \min\{|\mathcal{V}(\mathcal{M}_d)\cap\mathcal{V}_a|, 1\}.$$

Proof. In order to prove the claim of the theorem, we first prove that any sufficient representation $\tilde{\mathcal{V}}'_C = \{\mathcal{V}'_1, \mathcal{V}'_2, \ldots, \mathcal{V}'_l\}$ contains the sufficient representation of minimal cardinality $\tilde{\mathcal{V}}_C$. Assume that $\tilde{\mathcal{V}}_C = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_m\}$, $\tilde{\mathcal{V}}'_C = \{\mathcal{V}'_1, \mathcal{V}'_2, \ldots, \mathcal{V}'_l\}$, and let $\mathcal{V}_j$ be an arbitrary selected scenario from $\tilde{\mathcal{V}}_C$. We prove this scenario belongs to $\tilde{\mathcal{V}}'_C$ as well.

Assume first that this is not the case. The first option is that for any $\mathcal{V}'_i \in \tilde{\mathcal{V}}'_C$ we have $\mathcal{V}'_i \setminus \mathcal{V}_j \neq \emptyset$. If we choose a set of prevented scenarios to be $\mathcal{V}_P = \{v_{p_1}, \ldots, v_{p_l}\}$, where $v_{p_i} \in \mathcal{V}'_i \setminus \mathcal{V}_j$, we see that $\mathcal{V}_P$ prevents $\tilde{\mathcal{V}}'_C$. However, $\tilde{\mathcal{V}}_C$ is not prevented by $\mathcal{V}_P$, since $\mathcal{V}_j$ is not prevented. This is inconsistent with the fact that both $\tilde{\mathcal{V}}_C$ and $\tilde{\mathcal{V}}'_C$ are sufficient representations of the set $\mathcal{V}_C$.

Therefore, there has to be at least one scenario $\mathcal{V}'_t \in \tilde{\mathcal{V}}'_C$ such that $\mathcal{V}'_t \subseteq \mathcal{V}_j$. Assume that $\mathcal{V}_t \subset \mathcal{V}_j$, and define the set of prevented scenarios as $\mathcal{V}_P = \{v_{p_1}, \ldots, v_{p_m}\}$, where $v_{p_i} \in \mathcal{V}_i \setminus \mathcal{V}_j$ for $i \neq j$ and $v_{p_j} \in \mathcal{V}_j \setminus \mathcal{V}'_t$. This set prevents $\tilde{\mathcal{V}}_C$, but does not prevent $\tilde{\mathcal{V}}'_C$ since it does not prevent $\mathcal{V}'_t$. This is again in contradiction with the fact that both $\tilde{\mathcal{V}}_C$ and $\tilde{\mathcal{V}}'_C$ are sufficient representations of $\mathcal{V}_C$. Thus the only option that remains is that $\mathcal{V}_j = \mathcal{V}'_t$. Since $\mathcal{V}_j$ was arbitrarily selected, we conclude that any scenario contained in $\tilde{\mathcal{V}}_C$ has to be contained in $\tilde{\mathcal{V}}'_C$ as well.

Let $m \in \mathcal{M}$ be an arbitrary security measure. We then have

$$\begin{aligned} G'(m) &= \sum_{\mathcal{V}_a\in\tilde{\mathcal{V}}'_C} \min\{|\mathcal{V}_m\cap\mathcal{V}_a|, 1\} \\ &= \sum_{\mathcal{V}_a\in\tilde{\mathcal{V}}_C} \min\{|\mathcal{V}_m\cap\mathcal{V}_a|, 1\} + \sum_{\mathcal{V}_a\in\tilde{\mathcal{V}}'_C\setminus\mathcal{V}_C} \min\{|\mathcal{V}_m\cap\mathcal{V}_a|, 1\} \\ &= G(m) + \sum_{\mathcal{V}_a\in\tilde{\mathcal{V}}'_C\setminus\mathcal{V}_C} \min\{|\mathcal{V}_m\cap\mathcal{V}_a|, 1\} \geq G(m) \end{aligned}$$

Thus, for any m, we have $G(m) \leq G'(m)$. This implies

$$H\big(\max_{m \in \mathcal{M}} G(m)\big) \leq H\big(\max_{m \in \mathcal{M}} G'(m)\big)$$

which concludes the proof. □

5.5 Illustrative Example

In this section, we illustrate the applicability of our framework. We consider an ICS that regulates temperature within a building. The assumption is that multiple security vulnerabilities are present in the system. The goal is to identify the most critical scenarios that have to be prevented, and then to select the security measures such as to prevent these scenarios at the minimal cost.

5.5.1 System Model

The variable-refrigerant-flow system consists of a compressor, a condenser, N evaporators, and N electronic expansion valves (EEV), as shown in Figure 5.2. Each EEV and evaporator pair corresponds to an area of the building. The objective of the system is to maintain desirable temperature within the areas. We now briefly introduce the state space model of the system, and we refer the reader to [117] for more detailed treatment of the model.

Each area of the building is modeled with three state space variables $x_i = \begin{bmatrix} T_{ai} & T_{wi} & P_i \end{bmatrix}^T$, where T_{ai} is the temperature of the corresponding area, T_{wi} is the temperature of the evaporator's lumped coil wall, and P_i is the refrigerant pressure after leaving the evaporator. These variables are controlled using the control signal $u_i = \begin{bmatrix} \omega_{fi} & a_{vi} \end{bmatrix}^T$, where ω_{fi} is the speed of evaporator's fan that is used to cool down the evaporator coil, and a_{vi} is the control signal that is used to change the fluid resistance of EEV.

Other dynamical states of the system are $x_c = \begin{bmatrix} P_C & P_q & T_{wc} \end{bmatrix}^T$, where P_C represents the refrigerant pressure after leaving the compressor, T_{wc} is the temperature of compressor's lumped coil wall, and P_q is the junction pressure. These variables are controlled using the control inputs $u_c = \begin{bmatrix} \omega_k & \omega_{fc} \end{bmatrix}^T$, where ω_k represents the compressor speed, and ω_{fc} represents the speed of the compressor's fan. All of the states in the system are assumed to be measurable. Thus, the state, the control, and the measurement vectors are given by $x = \begin{bmatrix} x_1^T & \dots & x_N^T & x_c^T \end{bmatrix}^T$, $u = \begin{bmatrix} u_1^T & \dots & u_N^T & u_c^T \end{bmatrix}^T$, and $y = \begin{bmatrix} x_1^T & \dots & x_N^T & x_c^T \end{bmatrix}^T$.

For illustration purposes, we assume the cyber-part of the system to be as shown in Figure 5.3. In this configuration, the equipment in each of the N areas is controlled using the two PLCs – the master PLC, and the slave PLC. The master PLC is used to control the evaporator. It collects the temperature of the evaporator's lumped coil wall T_{wi}, and controls the speed of the evaporator's fan ω_{fi}. It is also assumed that master PLCs receive and execute commands from the control

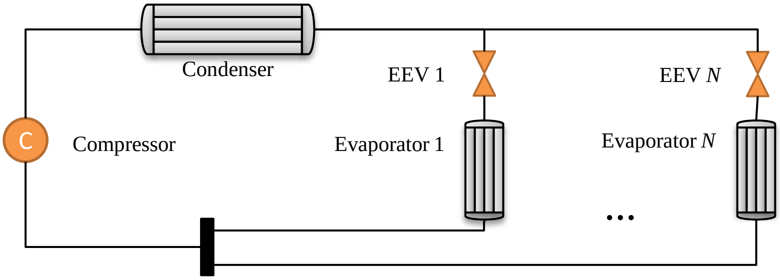


Figure 5.2: Schematic of variable-refrigerant-flow system used for air conditioning in the buildings.

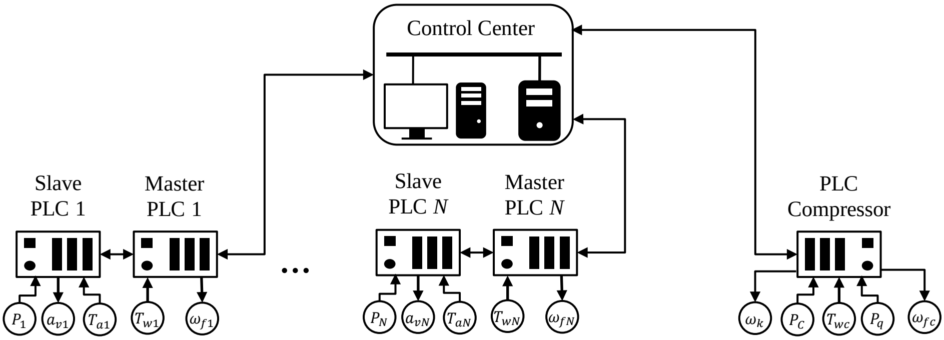


Figure 5.3: Cyber infrastructure of the variable refrigerant flow control system.

center. The tasks of slave PLCs include collecting the measurements P_i and T_{ai}, and controlling the actuator a_{vi}. It is assumed that slave PLCs communicate with the control center through their corresponding master PLCs. The compressor is controlled using a single PLC. This device collects the measurements P_C, P_q and T_{wc} and controls the actuators ω_k and ω_{fc}. It also exchanges the measurements and executes the commands from the control center.

5.5.2 Security Vulnerabilities and Security Measures

To model the sets of security vulnerabilities $\mathcal{V}$ and security measures $\mathcal{M}$, we used the list of common security vulnerabilities provided in [7]. In particular, we assumed the vulnerabilities listed in Table 5.1 are present in the system. Table 5.1 also contains the list of sensors and actuators that the attacker gains control over if the vulnerability v is exploited. The security measures are listed in Table 5.2. The vulnerabilities $\mathcal{V}_m$ prevented by the security measure m, and price c_m of implementing m are also provided in Table 5.2. The way of selecting c_m is explained in further sections.

Table 5.1: Vulnerabilities identified within the system, sensors and actuators the attacker gains control over if the vulnerability is exploited.

Vulnerability v	**Sensors** S_v **and actuators** A_v **controlled by attacker**
The control center computers connected to other networks without protection	All sensors and actuators
Unsecured physical ports in the control center	All sensors and actuators
Insecure connection between sensor or actuator i and a PLC	Sensor or actuator i
Insecure connection between slave i and master i	Sensors and actuators attached to slave i
Insecure connection between master i and the control center	Sensors and actuators attached to master i and slave i
Insecure connection between the compressor PLC and the control center	Sensors and actuators attached to the compressor PLC
Lack of physical protection of slave i	Sensors and actuators attached to slave i
Lack of physical protection of master i	Sensors and actuators attached to master i and slave i
Lack of physical protection of the compressor PLC	Sensors and actuators attached to the compressor PLC
Lack of physical protection of sensor or actuator i	Sensor or actuator i

Firstly, computers within the control center are connected to other IT networks without adequate protection, and physical ports on the computers are not secured. These vulnerabilities open the space for the attacker to spread malware within the system, either through the other networks or by USB sticks. In both of the cases, the attacker gains absolute control over all the sensors and actuators within the system. The first vulnerability can be prevented by deploying and properly adjusting firewalls, and the second one by locking or removing the ports.

Secondly, it was identified that the communication links between master and slave PLC controllers, master PLC controllers and the control center, and sensors/actuators and all the PLCs are unsecured. This opens a space for the attacker to intercept the communication and conduct man-in-the-middle attacks. The sensors and actuators that the attacker gains control over are dependent on a commu-

Table 5.2: Security measures, vulnerabilities prevented by these measures, and cost of deployment.

Security measure m	**Vulnerabilities** $\mathcal{V}_m$ **prevented by** m	**Cost** c_m
Installing and adjusting firewalls	The attacker cannot gain access to the computers in the control center from other networks	3–5
Locking and removing the physical ports	The attacker cannot inject malware through the physical ports within the control center	1–2
Protecting an unsecured communication link	The attacker cannot intercept and modify messages going through the corresponding link	1–4
Physical protection of an individual component (a sensor, an actuator, or a PLC controller)	The attacker cannot gain physical access to the component	1–2
Physical protection of a group of components (PLC controller and sensors and actuators attached to it)	The attacker cannot physically access the group of components, and cannot exploit unprotected communication between PLC and sensors/actuators	5–10

nication link attacked. If a link between a sensor/actuator and PLC is attacked, the attacker gains control over that sensor/actuator. If the attacker intercepts the communication between master and slave PLC, it gains control over all the sensors and actuators attached to the slave PLC. If the attacker compromises a link between a master PLC and the control center, it gains control over the sensors and actuators attached to both the master PLC and to the slave PLC. The vulnerabilities of this type can be prevented by implementing encryption and authentication schemes.

Finally, lack of physical protection is identified within the system. If the attacker has physical access to a sensor/actuator, we assume that the attacker can gain control over that sensor/actuator. If the attacker gains unauthorized access to a slave PLC, it gains control over all the sensors and actuators attached to that PLC. In case of unauthorized access to a master PLC, the attacker gains control over all the sensors and actuators attached to both the master and the slave PLC.

The unauthorized access can be prevented by introducing additional physical protection. The first alternative is to protect the group of components at the same time. The assumption is that physical protection prevents the attacker from gaining unauthorized access to the PLC and all the sensors and the actuators attached to that PLC. In addition, it also prevents the attacker from exploiting unsecured connections between the corresponding PLC and sensors/actuators, since these components are usually connected with wires, and lie in proximity of each other. The second alternative is to protect components individually.

By counting the total number of vulnerabilities, it can be verified that the total number of vulnerabilities is equal to $14N + 14$, where N represents the number of areas. The number of possible security measures is equal to $16N + 15$.

5.5.3 Attack Impact

One model of attacks introduced in control literature are so called zero dynamics attacks [34, 38]. These attacks are serious, since from the sensor data, the attacks are indistinguishable from the normal system operation. In that way, the attacker is able to potentially make some of the system states arbitrarily large, while staying undetected by the system operator at the same time.

To determine if it is possible for the attacker to conduct a zero dynamic attack, we first introduce the physical model of the system

$$\mathcal{P}\colon \begin{cases} x(k+1) = Ax(k) + B\tilde{u}(k) \\ \quad\;\; y(k) = Cx(k) \end{cases} \tag{5.4}$$

where $x(k) \in \mathbb{R}^{n_x}$ is the state of the system, $\tilde{u}(k) \in \mathbb{R}^{n_u}$ is the control signal applied to the process, and $y(k) \in \mathbb{R}^{n_y}$ is the vector of sensor measurements collected from the process. Due to attacks, the signal $\tilde{u}(k)$ is different from the control signals calculated by the controllers, which we denote with $u(k)$. Similarly, due to attacks against sensors, the operators receive false measurements $\tilde{y}(k)$ instead of the original ones $y(k)$.

Let $\mathcal{V}_a$ be an attack scenario, and $\mathcal{S}(\mathcal{V}_a)$ and $\mathcal{A}(\mathcal{V}_a)$ be the sensors and actuators controlled by the attacker in that scenario. The signals $\tilde{y}(k)$ and $\tilde{u}(k)$ can then be modeled as

$$\tilde{y}(k) = y(k) + D_y(\mathcal{V}_a)a_y(k) \qquad\qquad \tilde{u}(k) = u(k) + D_u(\mathcal{V}_a)a_u(k)$$

where $a_y(k) \in \mathbb{R}^{|\mathcal{S}(\mathcal{V}_a)|}$ is the additive attack signal added to the measurements from sensors $\mathcal{S}(\mathcal{V}_a)$, and $a_u(k) \in \mathbb{R}^{|\mathcal{A}(\mathcal{V}_a)|}$ be the additive attack signal sent to the actuators $\mathcal{A}(\mathcal{V}_a)$. It is important to note that the matrices $D_u(\mathcal{V}_a) \in \mathbb{R}^{n_u \times |\mathcal{A}(\mathcal{V}_a)|}$ and $D_y(\mathcal{V}_a) \in \mathbb{R}^{n_y \times |\mathcal{S}(\mathcal{V}_a)|}$ are dependent on the attack scenario $\mathcal{V}_a$. If the attacker is able to manipulate the sensors $\mathcal{S}(\mathcal{V}_a) = \{s_{j_1}, s_{j_2}, \ldots, s_{j_p}\}$ by exploiting vulnerability $\mathcal{V}_a$, then the elements $(j_1, 1), (j_2, 2), \ldots, (j_p, p)$ of the matrix $D_y(\mathcal{V}_a)$ are equal to one, and the remaining elements are equal to zero. The matrix $D_u(\mathcal{V}_a)$ is defined in an analogous way, but this time based on the set of exploited actuators $\mathcal{A}(\mathcal{V}_a)$. We also assume that the matrix $BD_u(\mathcal{V}_a)$ has a full column rank, which is an assumption adopted to exclude the attack signals that cancel each other and do not lead to any impact. Undetectable attacks can then be defined as follows.

Definition 6. The attack signals $a_u(k)$ and $a_y(k)$ are *undetectable*, if there exists an initial state $x(0)$ such that $\tilde{y}(k) = 0$ for $k \geq 0$.

In order to check if the attacker can conduct undetectable attack using the components $\mathcal{S}(\mathcal{V}_a)$ and $\mathcal{A}(\mathcal{V}_a)$, the Rosenbrock matrix [29] of the system

$$P(z) = \begin{bmatrix} A - zI & BD_u(\mathcal{V}_a) & \mathbf{0}_{n_x \times |\mathcal{S}(\mathcal{V}_a)|} \\ C & \mathbf{0}_{n_y \times |\mathcal{A}(\mathcal{V}_a)|} & D_y(\mathcal{V}_a) \end{bmatrix}$$

can be used [34, 38]. In particular, the attack is undetectable if and only if there exists $z_0 \in \mathbb{C}$, $x_0 \in \mathbb{C}^{n_x}$, $a_u \in \mathbb{C}^{|\mathcal{A}(\mathcal{V}_a)|}$, and $a_y \in \mathbb{C}^{|\mathcal{S}(\mathcal{V}_a)|}$ such that

$$P(z_0) \begin{bmatrix} x_0 \\ a_u \\ a_y \end{bmatrix} = 0.$$

The attack signals are then of the form $a_u(k) = a_u z_0^k$ and $a_y(k) = a_y z_0^k$. Two special cases of undetectable attacks are particularly important. If the condition

$$P(z_0) \begin{bmatrix} x_0 \\ a_u \\ a_y \end{bmatrix} = 0 \qquad\qquad \text{for some} \quad |z_0| > 1 \tag{5.5}$$

is satisfied, the attacker can make some of the system states arbitrarily large while staying undetected at the same time. An even more dangerous scenario occurs when

$$\text{normalrank}\big(P(z)\big) = \max_z \, \text{rank}\big(P(z)\big) < n_x + |\mathcal{S}(\mathcal{V}_a)| + |\mathcal{A}(\mathcal{V}_a)|. \tag{5.6}$$

In that case, the attacker can drive some of the system states to infinity *for any complex frequency* $|z_0| > 1$ while remaining undetected. Due to the importance of the Rosenbrock matrix in control theory, both the condition (5.5) and (5.6) can be checked efficiently for a given scenario $\mathcal{V}_a$ using well established algorithms.

Based on the previously introduced attack strategy, one way to define the attack impact function would be as shown in Table 5.3. Note that this impact metric satisfies Assumption 6. Mainly, if the attacker is able to conduct the zero dynamic attack using the components $\mathcal{S}_a$ and $\mathcal{A}_a$, then it can always conduct this same attack with the larger set of components $\mathcal{S}_b \supseteq \mathcal{S}_a$ and $\mathcal{A}_b \supseteq \mathcal{A}_a$. It just needs to send the same signals to sensors $\mathcal{S}_a$ and actuators $\mathcal{A}_a$, while keeping the attack signals that corresponds to sensors $\mathcal{S}_b \setminus \mathcal{S}_a$ and actuators $\mathcal{A}_b \setminus \mathcal{A}_a$ equal to zero.

We are interested in finding the scenarios that can lead to impact greater than $\bar{I} = 2$. To check if the attack impact is above or below the threshold, we use the Matlab function *tzero* which returns both the normal rank of the system and the list of transmission zeros.

Table 5.3: Impact metric $I(\mathcal{A}(\mathcal{V}_a), \mathcal{S}(\mathcal{V}_a))$.

Impact I	**Description**
0	The attacker cannot conduct undetectable attack
1	The attacker can conduct undetectable attack for some $\lvert z_0 \rvert < 1$
2	The attacker can conduct undetectable attack for some $\lvert z_0 \rvert \geq 1$
3	The attacker can conduct undetectable attack for any z_0

Table 5.4: Assigning complexity π_v of exploiting individual vulnerabilities $v \in \mathcal{V}$.

Complexity	**Very Low**	**Low**	**Medium**	**Difficult**	**Very Difficult**
π_v	1	2	3	4	5

5.5.4 Attack Complexity

The attack complexity is modeled as follows. The first step is to assign complexity $\pi_v > 0$ to each of the vulnerabilities $v \in \mathcal{V}$ according to Table 5.4. These costs can be obtained based on a security expert knowledge, taking into consideration several factors [19]. For simplicity, we assumed that the security vulnerabilities related to the equipment in areas:

- 1 to $\frac{N}{5}$ are of very low complexity;
- $\frac{N}{5} + 1$ to $\frac{2N}{5}$ are of low complexity;
- $\frac{2N}{5} + 1$ to $\frac{3N}{5}$ are of medium complexity;
- $\frac{3N}{5} + 1$ to $\frac{4N}{5}$ are difficult to exploit;
- $\frac{4N}{5} + 1$ to N are very difficult to exploit.

The vulnerabilities related to the control center and the equipment controlling compressor were assumed to be of very low complexity. In this way, we achieved approximately the equal number of vulnerabilities belonging to each of the five groups from the Table 5.4.

In the second step, we define complexity of scenarios consisting of multiple vulnerabilities $\mathcal{V}_a \subseteq \mathcal{V}$ as

$$\pi(\mathcal{V}_a) = \sum_{v \in \mathcal{V}_a} \pi_v. \tag{5.7}$$

Although simple, this function captures the essence of the problem. Scenarios containing vulnerabilities with higher values of π_v have a larger total complexity then those with equal number of exploited vulnerabilities, but with lower values of π_v. This complexity function is also non-decreasing since it represents a non-negative sum. Thus, it satisfies Assumption 7.

Table 5.5: Execution times of Algorithm 1.

No. of areas N	**No. of vulnerabilities** n_v	**Execution time Algorithm 1** [s]
5	84	4.2301
10	154	13.954
15	224	80.285
20	294	366.030
25	364	1292.560

We set the threshold to be $\bar{\pi} = 5$. This choice of the threshold $\bar{\pi}$ implies that all the scenarios with 6 or more vulnerabilities are with the complexity greater than $\bar{\pi}$. Thus, we need to explore only the first five layers of the power set $2^{\mathcal{V}}$ in order to find the critical scenarios.

5.5.5 Searching for Critical Scenarios

To find the sufficient representation of minimal cardinality, we used a computer cluster consisting of 4 Intel(R) Core(TM) i7-4470s computers with 16 cores in total. In order to increase or decrease the number of vulnerabilities, we varied the number of areas in the range N=5 to N=25.

For the aforementioned specifications, we measured the execution time of Algorithm 1. The results are shown in Table 5.5. The execution time was highest for the case of 364 vulnerabilities, where it reached 21.54 minutes. For the sake of comparison, the brute force search through the first five layers of the power set when vulnerability set consisted of 84 measures already took more then 8 minutes, while the estimated time of the brute force search for the vulnerability set consisting of 364 elements is more then 20 days. This demonstrates that systematic search enables exploring large sets of vulnerabilities in reasonable time.

The number of scenarios in the sufficient representation of minimal cardinality is reported in Table 5.6, together with the number of scenarios in each of the layers. As it can be seen, the attacker can conduct undetectable attack by exploiting only a single vulnerability. Naturally, if the attacker exploits vulnerabilities within the control center, it is able to conduct an undetectable attack, since it controls all of the sensors and actuators. It also turned out that attacking any of the master PLCs leads to undetectable attacks as well. For instance, if the attacker exploits lack of physical protection of a master PLC i, it can increase or decrease the temperature of the room T_{ai} and the temperature of the evaporator's coil T_{wi} by an arbitrary value. This is possible since changes in temperatures T_{ai} and T_{wi} influence neither pressures nor temperatures in other areas. The number of scenarios in the second layer was equal to four for all the values of N. The reason is that these scenarios involved only the vulnerabilities of equipment related to the compressor, which does not change by increasing or decreasing the number of areas. We also observe

Table 5.6: The number of critical scenarios within the sufficient representation of minimal cardinality $\tilde{\mathcal{V}}_C$ in total, and in each of the layers.

No. of areas N	**Total** $\lvert\tilde{\mathcal{V}}_C\rvert$	**Layer 1**	**Layer 2**	**Layer 3**	**Layer 4**	**Layer 5**
5	226	14	4	16	0	192
10	508	24	4	32	64	384
15	854	34	4	48	192	576
20	1264	44	4	64	384	768
25	1738	54	4	80	640	960

that scenarios from the layer 5 involved exploiting vulnerabilities of the equipment related to the compressor. For example, in order to increase or decrease pressure of the refrigerant P_i, the attacker needs to control the measurements of pressures P_c and P_q. This is to be expected, since the pressures P_c and P_q are coupled with all the pressures $P_1, \dots, P_N$.

5.5.6 Allocating Security Measures

Once the sufficient representation of minimal cardinality was found, we moved to solving the security measure allocation problem. We used both the greedy algorithm introduced in the previous section, but also specialized integer linear program solver included in the Gurobi package. For each case of N, we performed simulations 500 times for different values of costs c_m of security measures. The values of c_m were randomly selected from the intervals given in Table 5.2.

We first compare the algorithms in terms of execution time. The plot of the worst case execution times of the algorithms is shown in Figure 5.4. The maximal execution time was reached for 415 security measures, and it was approximately 8.4075 seconds for the Gurobi solver, and 0.0873 seconds for Algorithm 2. The explanation for short execution times lies in the fact that sufficient representation of minimal cardinality contained relatively small number of scenarios, and all of the scenarios were with cardinality not greater than 5. However, it can also be observed from Figure 5.4 that the worst case execution time increased much faster for the Gurobi solver than for Algorithm 2. This may indicate the restriction of using Gurobi solver once the number of scenarios to be prevented is very large. In that case, we can rely on Algorithm 2 to solve the problem with known performance guarantees. However, in this particular case, we conclude that the execution time did not represent an issue for any of the approaches.

Next, we compare the algorithms in terms of the solution obtained. We first remark that the Gurobi solver managed to find the optimal objective value of the problem $b(\mathcal{M}_O)$ in all the five cases, for all the realizations of costs. Again, we find the reason for this to be relatively small number of scenarios contained in the sufficient representation of minimal cardinality, and sparsity of the scenarios. Thus,

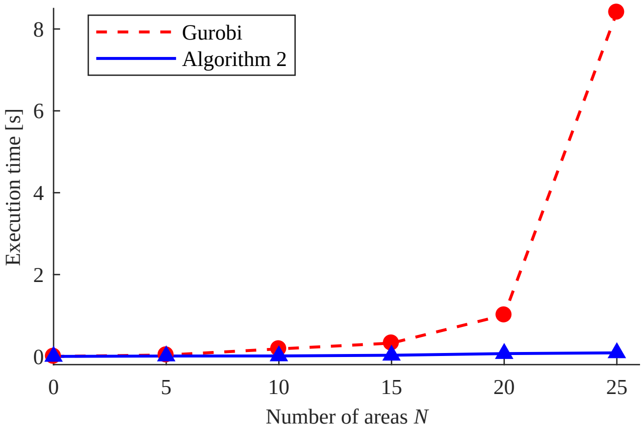


Figure 5.4: Comparison of Algorithm 2 and the Gurobi solver in terms of execution time.

Table 5.7: The comparison of the solutions $b(\mathcal{M}_G)$ obtained by Algorithm 2 and $b(\mathcal{M}_0)$ obtained by the Gurobi solver. The values of the bound introduced in Lemma 12 are also provided.

No. of areas N	**No. of security measures** n_m	**Max. quotient** $b(\mathcal{M}_G)/b(\mathcal{M}_O)$	**Bound Lemma 12**
5	95	1.22	5.86
10	175	1.19	6.54
15	255	1.16	6.94
20	335	1.11	7.23
25	415	1.10	7.45

although integer linear programs are NP hard in general, in this case we managed to find the optimal solution for the problem in a matter of seconds. To compare the solution $b(\mathcal{M}_G)$ returned by Algorithm 2 with the solution $b(\mathcal{M}_O)$ returned by the Gurobi solver, we recorded the worst case values of the quotient $b(\mathcal{M}_G)/b(\mathcal{M}_O)$ in Table 5.7. We also calculated bound from Lemma 12. From Table 5.7, we see that the solution $b(\mathcal{M}_G)$ returned by Algorithm 2 was close to the optimal. In particular, it was at most 1.22 times greater than the one returned by the Gurobi solver, which demonstrates that the bound stated in Lemma 12 may be quite conservative.

In Table 5.8, we recorded the maximal percentages of security measures deployed (number of deployed security measures divided by the total number of security measures). The number of deployed security measures varied from 18.31% to 21.05% for the Gurobi solver. For Algorithm 2, percentage of deployed security measures varied from 20.00% to 25.26%. This illustrates that by preselecting the most dan-

Table 5.8: The largest percentages of deployed security measures obtained in simulations.

No. of areas N	**Max. percentage of deployed measures Gurobi [%]**	**Max. percentage of deployed measures Algorithm 2 [%]**
5	21.05	25.26
10	19.43	22.29
15	18.82	21.20
20	18.51	20.60
25	18.31	20.00

gerous vulnerabilities and then allocating security measures in a systematic way, we can protect the system with relatively low number of deployed security measures. The security measures that were selected by the algorithm were mostly protecting the major components in the system. For instance, implementing protection within the control center, communication links between master PLCs and the control center, and physical protection of master PLCs. As expected, security measures that have not been implemented were mostly those preventing vulnerabilities of slave PLCs and individual sensors/actuators with the high values of complexity π_v.

5.6 Summary

In this chapter, we proposed a modeling framework for allocating security measures in a cost efficient way. The security measure allocation problem reduces to an integer linear program, which is difficult to construct, and NP hard to solve in general. To construct the security measure allocation problem, high risk combinations of vulnerabilities have to be found. For this purpose, an algorithm that systematically searches for these combinations was proposed. To approximately solve the problem, it was shown that the problem has submodular structure. Thus, a polynomial time greedy algorithm can be used to obtain a suboptimal solution with guaranteed performance bounds. The applicability of the framework was illustrated in a simulation study.

Chapter 6

Conclusions and Future Work

In this chapter, we summarize the results of this thesis, and outline possible directions for future work.

6.1 Conclusions

In Chapter 3, we proposed a modeling framework that can be used for estimating the impact of cyber-attacks against ICSs with control tasks. The framework can be used to estimate the impact of both simple and more complex attack strategies. In particular, denial of service, sign alternation, rerouting, replay, false data injection, and bias injection attack strategies were considered. The framework is applicable for stateless, moving window, cumulative sum, and multivariate exponentially weighted moving average detectors. The problem of estimating the impact was reduced to solving multiple convex minimization problems. Thus, the exact value of the attack impact can be obtained using standardized convex optimization solvers. It was also demonstrated how our modeling framework can be used for risk assessment on a model of chemical process.

In Chapter 4, we considered the problem of estimating the impact of bias injection attacks against an ICS used for monitoring task. The system used a Kalman filter to estimate the state of the physical process, and it was equipped with the chi-squared anomaly detector. We proved that the problem of finding the worst-case bias injection attack can be reduced to a quadratically constrained quadratic program, for which the optimal value can be obtained using well known algorithms. A lower bound of the attack impact was also derived. The theoretical results of Chapter 4 were illustrated on a model of quadruple-tank system.

In Chapter 5, we proposed a modeling framework for allocating security measures in dynamical control systems. Prior to solving the security measure allocation problem, high risk combinations of vulnerabilities have to be found. For this purpose, we proposed an algorithm that systematically searches for these combinations. Once the high risk combinations of vulnerabilities are found, security measures should be selected to prevent them. This problem reduces to solving an integer linear program. Given that these programs are NP hard in general, the submodular structure of the problem was outlined, and a polynomial time greedy algorithm was suggested to solve the problem with guaranteed performance bounds. The applicability of the framework was illustrated on an ICS used for air conditioning in buildings.

6.2 Future work

In this section, we discuss possible directions for future work.

6.2.1 System Models

In Chapter 3, the attack strategies were developed for noiseless, linear time invariant systems. As we saw in Chapter 4, the analysis becomes more complex once the system and measurement noises are present in the system. Thus, a possible extension of the work will be to include process and measurement noises into the framework considered in Chapter 3. System models different than linear time invariant, such as nonlinear and hybrid, may also be considered.

6.2.2 Anomaly Detectors

In the thesis, we considered the problem of estimating the attack impact against anomaly detectors that are developed to detect faults. As discussed in Section 2.5, novel types of anomaly detectors have been developed to detect attacks. Estimating the attack impact against these novel types of anomaly detectors is still unexplored. This would be interesting for both risk assessment purposes and for checking the limitations of novel detectors.

Another improvement of the framework from Chapter 3 would be to conduct an impact analysis independent of the model of the anomaly detector. One way to do this would be to use a stealthiness constraint that is independent of the anomaly detector used. For instance, the Kullback–Leibler divergence was proposed to measure the stealthiness of attacks [92].

6.2.3 Experimental Verification

The issue with the model based impact analysis is that it is dependent on the system model. If the model is not a good approximation of the real system, the attack impact estimate may be imprecise. Therefore, experiments to check the accuracy of the estimated attack impact are important to consider.

6.2.4 Security Measure Allocation Framework

Improvements of the framework for allocating security measures are also planned. One future research direction is to make Algorithm 1, used for finding the critical attack scenarios, faster. In particular, the goal is to investigate how to exploit prior information about the structure and symmetry of the physical model for this purpose.

6.2.5 Security Index

As we mentioned in the Section 2.5, another way of analyzing the security of a control system is by using a security index. Moreover, this index also proved to be useful for deploying security measures. Unfortunately, finding the security index is an NP hard combinatorial optimization problem in general. However, in the case of static models of power grids, it was shown that the security index can in some cases be efficiently found [47]. No such extension is known for dynamical models of control systems, which is an interesting topic to explore.

Bibliography

[1] J. Slay and M. Miller, "Lessons learned from the Maroochy water breach," *in Proceedings of the International Conference on Critical Infrastructure Protection*, 2007.

[2] M. Abrams and J. Weiss, *Malicious control system cyber security attack case study–Maroochy Water Services, Australia.* The MITRE corporation, 2008.

[3] D. Kushner, "The real story of Stuxnet," *IEEE Spectrum*, vol. 50, no. 3, pp. 48–53, 2013.

[4] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[5] S. Karnouskos, "Stuxnet worm impact on industrial cyber-physical system security," *in Proceedings of the 37th Annual Conference on IEEE Industrial Electronics Society (IECON)*, 2011.

[6] *Analysis of the Cyber Attack on the Ukrainian Power Grid.* Electricity Information Sharing and Analysis Center, 2016.

[7] K. Stouffer, J. Falco, and K. Scarfone, *Guide to industrial control systems (ICS) security.* United States. National Institute for Standards and Technology, 2015.

[8] *Recommended Practice: Improving Industrial Control Systems Cybersecurity with Defense-in-Depth Strategies.* United States. Department of Homeland Security, 2016.

[9] *Guide to Increased Security in Industrial Information and Control Systems.* Sweden. Swedish Civil Contingencies Agency, 2014.

[10] T. Samad, P. McLaughlin, and J. Lu, "System architecture for process automation: Review and trends," *Journal of Process Control*, vol. 17, no. 3, pp. 191–201, 2007.

[11] R. Krutz, *Securing SCADA systems.* John Wiley & Sons, 2005.

[12] E. Knapp and J. Langill, *Industrial Network Security: Securing critical infrastructure networks for smart grid, SCADA, and other Industrial Control Systems.* Syngress, 2014.

[13] *Security for industrial automation and control systems:Models and Concepts.* International Society of Automaton, 2016.

[14] B. Zheng, P. Deng, R. Anguluri, Q. Zhu, and F. Pasqualetti, "Cross-layer codesign for secure cyber-physical systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 5, pp. 699–711, 2016.

[15] A. Cárdenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems," *in Proceedings of the 3rd USENIX Workshop on Hot Topics in Security (HotSec)*, 2008.

[16] D. Sullivan, E. Luiijf, and E. Colbert, *Components of Industrial Control Systems.* Springer International Publishing, 2016, pp. 15–28.

[17] M. A. Bishop, *The art and science of computer security.* Addison-Wesley Longman Publishing Co., Inc., 2002.

[18] *Guide for Conducting Risk Assessment.* United States. National Institute of Standards and Technology, 2012.

[19] E. J. Byres, M. Franz, and D. Miller, "The use of attack trees in assessing vulnerabilities in SCADA systems," *in Proceedings of the International Infrastructure Survivability Workshop*, 2004.

[20] M. R. Permann and K. Rohde, "Cyber assessment methods for SCADA security," *in Proceedings of the 15th Annual Joint ISA POWID/EPRI Controls and Instrumentation Conference*, 2005.

[21] S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber physical system security for the electric power grid," *Proceedings of the IEEE*, vol. 100, no. 1, 2012.

[22] O. Vuković, K. Sou, G. Dan, and H. Sandberg, "Network-aware mitigation of data integrity attacks on power system state estimation," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 6, pp. 1108–1118, 2012.

[23] J. Milošević, T. Tanaka, H. Sandberg, and K. H. Johansson, "Exploiting submodularity in security measure allocation for industrial control systems," *in Proceedings of the 1st ACM Workshop on the Internet of Safe Things (SafeThings'17)*, 2017.

[24] S. Ding, *Model-based fault diagnosis techniques: design schemes, algorithms, and tools.* Springer Science & Business Media, 2008.

[25] A. M. H. Teixeira, J. Araujo, H. Sandberg, and K. H. Johansson, "Distributed sensor and actuator reconfiguration for fault-tolerant networked control systems," *IEEE Transactions on Control of Network Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[26] K. Paridari, N. O'Mahony, A. E.-D. Mady, R. Chabukswar, M. Boubekeur, and H. Sandberg, "A framework for attack-resilient industrial control systems: Attack detection and controller reconfiguration," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 113–128, Jan 2018.

[27] K. J. Åström, *Introduction to stochastic control theory.* Courier Corporation, 2012.

[28] A. Bemporad, M. Heemels, and M. Johansson, *Networked control systems.* Springer, 2010.

[29] K. Zhou, J. C. Doyle, K. Glover *et al.*, *Robust and optimal control.* Prentice hall New Jersey, 1996.

[30] Y. Liu, P. Ning, and M. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information Systems Security*, vol. 14, no. 1, pp. 13:1–13:33, 2011.

[31] Y. Lun, A. D'Innocenzo, I. Malavolta, and M. Di Benedetto, "Cyber-physical systems security: a systematic mapping study," *arXiv preprint arXiv:1605.09641*, 2016.

[32] J. Giraldo, E. Sarkar, A. Cárdenas, M. Maniatakos, and M. Kantarcioglu, "Security and privacy in cyber-physical systems: A survey of surveys," *IEEE Design Test*, vol. 34, no. 4, pp. 7–17, Aug 2017.

[33] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1454–1467, 2014.

[34] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2715–2729, Nov 2013.

[35] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterations in the presence of malicious agents Part I: $\mathcal{C}_{-}$," *in Proceedings of the American Control Conference (ACC)*, 2008.

[36] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen, "Cyber security of water SCADA systems – Part I: Analysis and experimentation of stealthy deception attacks," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1963–1970, 2013.

[37] A. Teixeira, G. Dan, H. Sandberg, and K. H. Johansson, “A cyber security study of a scada energy management system: Stealthy deception attacks on the state estimator,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 11 271 – 11 277, 2011.

[38] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, “A secure control framework for resource-limited adversaries,” *Automatica*, vol. 51, pp. 135–148, 2015.

[39] R. S. Smith, “Covert misappropriation of networked control systems: Presenting a feedback structure,” *IEEE Control Systems*, vol. 35, no. 1, pp. 82–92, 2015.

[40] Y. Mo, R. Chabukswar, and B. Sinopoli, “Detecting integrity attacks on SCADA systems,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1396–1407, 2014.

[41] S. Amin, A. A. Cárdenas, and S. Sastry, “Safe and secure networked control systems under denial-of-service attacks,” *in Proceedings of the International Workshop on Hybrid Systems: Computation and Control*, 2009.

[42] A. Teixeira, K. Paridari, H. Sandberg, and K. H. Johansson, “Voltage control for interconnected microgrids under adversarial actions,” *in Proceedings of the 20th IEEE Conference on Emerging Technologies & Factory Automation (ETFA)*, 2015.

[43] R. M. Ferrari and A. M. Teixeira, “Detection and isolation of routing attacks through sensor watermarking,” *in Proceedings of the American Control Conference (ACC)*, 2017.

[44] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, “Optimal linear cyber-attack on remote state estimation,” *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 4–13, 2017.

[45] C.-Z. Bai and V. Gupta, “On Kalman filtering in the presence of a compromised sensor: Fundamental performance bounds,” *in Proceedings of the American Control Conference (ACC)*, 2014.

[46] H. Sandberg, A. Teixeira, and K. Johansson, “On security indices for state estimators in power networks,” *in Proceedings of the First Workshop on Secure Control Systems (SCS)*, 2010.

[47] J. M. Hendrickx, K. H. Johansson, R. M. Jungers, H. Sandberg, and K. C. Sou, “Efficient computations of a security index for false data attacks in power networks,” *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3194–3208, 2014.

[48] K. C. Sou, H. Sandberg, and K. H. Johansson, "Computing critical k-tuples in power networks," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1511–1520, 2012.

[49] O. Kosut, "Max-flow min-cut for power system security index computation," *in Proceedings of the 8th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM),*, 2014.

[50] A. Teixeira, K. C. Sou, H. Sandberg, and K. H. Johansson, *Quantifying Cyber-Security for Networked Control Systems*. Heidelberg: Springer International Publishing, 2013, pp. 123–142.

[51] H. Sandberg and A. M. H. Teixeira, "From control system security indices to attack identifiability," *in Proceedings of the 2016 Science of Security for Cyber-Physical Systems Workshop (SOSCYPS)*, 2016.

[52] M. S. Chong and M. Kuijper, "Characterising the vulnerability of linear control systems under sensor attacks using a system's security index," *in Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, 2016.

[53] Y. Mo, S. Weerakkody, and B. Sinopoli, "Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs," *IEEE Control Systems*, vol. 35, no. 1, pp. 93–109, 2015.

[54] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "Revealing stealthy attacks in control systems," *in Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton 2012)*, 2012.

[55] A. Hoehn and P. Zhang, "Detection of covert attacks and zero dynamics attacks in cyber-physical systems," *in Proceedings of the American Control Conference (ACC)*, 2016.

[56] A. Jones, Z. Kong, and C. Belta, "Anomaly detection in cyber-physical systems: A formal methods approach," *in Proceedings of the 53rd IEEE Conference on Decision and Control (CDC)*, 2014.

[57] E. Eyisi and X. Koutsoukos, "Energy-based attack detection in networked control systems," *in Proceedings of the 3rd ACM International Conference on High Confidence Networked Systems*, 2014.

[58] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas, "Coding schemes for securing cyber-physical systems against stealthy data injection attacks," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 106–117, 2017.

[59] Q. Zhu and T. Basar, "Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: games-in-games principle for optimal cross-layer resilient control systems," *IEEE control systems*, vol. 35, no. 1, pp. 46–65, 2015.

[60] S. Amin, G. A. Schwartz, A. A. Cárdenas, and S. S. Sastry, "Game-theoretic models of electricity theft detection in smart utility networks: Providing new capabilities with advanced metering infrastructure," *IEEE Control Systems*, vol. 35, no. 1, pp. 66–81, 2015.

[61] D. Shelar and S. Amin, "Security assessment of electricity distribution networks under DER node compromises," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 23–36, March 2017.

[62] V. Ugrinovskii and C. Langbort, "Controller–jammer game models of denial of service in control systems operating over packet-dropping links," *Automatica*, vol. 84, pp. 128 – 141, 2017.

[63] M. Pajic, I. Lee, and G. J. Pappas, "Attack-resilient state estimation for noisy dynamical systems," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 82–92, 2017.

[64] Y. Mo and E. Garone, "Secure dynamic state estimation via local estimators," *in Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, 2016.

[65] Y. Mo and B. Sinopoli, "Secure estimation in the presence of integrity attacks," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 1145–1151, 2015.

[66] Y. Nakahira and Y. Mo, "Dynamic state estimation in the presence of compromised sensory data," *in Proceedings of the 54th IEEE Conference on Decision and Control (CDC)*, 2015.

[67] Y. Shoukry and P. Tabuada, "Event-triggered state observers for sparse sensor noise/attacks," *IEEE Transactions on Automatic Control*, vol. 61, no. 8, pp. 2079–2091, 2016.

[68] Y. Shoukry, P. Nuzzo, A. Puggelli, A. L. Sangiovanni-Vincentelli, S. A. Seshia, and P. Tabuada, "Secure state estimation for cyber-physical systems under sensor attacks: A satisfiability modulo theory approach," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 4917–4932, 2017.

[69] S. Mishra, Y. Shoukry, N. Karamchandani, S. N. Diggavi, and P. Tabuada, "Secure state estimation against sensor attacks in the presence of noise," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 49–59, 2017.

[70] S. D. Bopardikar, A. Speranzon, and J. P. Hespanha, "An H-infinity approach to stealth-resilient control design," *in Proceedings of the 2016 Resilience Week (RWS)*, 2016.

[71] G. K. Befekadu, V. Gupta, and P. J. Antsaklis, "Risk-sensitive control under markov modulated denial-of-service (DOS) attack strategies," *IEEE Transactions on Automatic Control*, vol. 60, no. 12, pp. 3299–3304, 2015.

[72] Y. Yan, P. Antsaklis, and V. Gupta, "A resilient design for cyber physical systems under attack," *in Proceedings of the American Control Conference (ACC)*, 2017.

[73] M. Zhu and S. Martínez, "On the performance analysis of resilient networked control systems under replay attacks," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 804–808, 2014.

[74] Q. Zhu, L. Bushnell, and T. Başar, "Resilient distributed control of multi-agent cyber-physical systems," in *Control of Cyber-Physical Systems.* Springer, 2013, pp. 301–316.

[75] C. Kwon and I. Hwang, "Hybrid robust controller design: Cyber attack attenuation for cyber-physical systems," *in Proceedings of the 52nd IEEE Conference on Decision and Control (CDC)*, 2013.

[76] A. Gupta, C. Langbort, and T. Başar, "Optimal control in the presence of an intelligent jammer with limited actions," *in Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, 2010.

[77] J. Wei and D. Kundur, *Biologically Inspired Hierarchical Cyber-Physical Multi-agent Distributed Control Framework for Sustainable Smart Grids.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 219–259.

[78] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterations in the presence of malicious agents Part II: Overcoming malicious behavior," *in Proceedings of the American Control Conference (ACC)*, 2008.

[79] S. M. Dibaji and H. Ishii, "Resilient consensus of second-order agent networks: Asynchronous update rules with delays," *Automatica*, vol. 81, pp. 123 – 132, 2017.

[80] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 1, pp. 90–104, 2012.

[81] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 766–781, 2013.

[82] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: Risk assessment, detection, and response," *in Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, 2011.

[83] C. M. Ahmed, C. Murguia, and J. Ruths, “Model-based attack detection scheme for smart water distribution networks,” *in Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 2017.

[84] D. I. Urbina, J. A. Giraldo, A. A. Cárdenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg, “Limiting the impact of stealthy attacks on industrial control systems,” *in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[85] D. Umsonst, H. Sandberg, and A. Cárdenas, “Security analysis of control system anomaly detectors,” *in Proceedings of the American Control Conference (ACC)*, 2017.

[86] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, and J. Schröder, *Diagnosis and fault-tolerant control.* Springer, 2006.

[87] S. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[88] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli, “False data injection attacks against state estimation in wireless sensor networks,” *in Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, 2010.

[89] C. Murguia, N. van de Wouw, and J. Ruths, “Reachable sets of hidden CPS sensor attacks: Analysis and synthesis tools,” *IFAC Proceedings Volumes*, vol. 50, no. 1, pp. 2088 – 2094, 2017.

[90] I. Jovanov and M. Pajic, “Sporadic data integrity for secure state estimation,” *in Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, 2017.

[91] C.-Z. Bai, F. Pasqualetti, and V. Gupta, “Data-injection attacks in stochastic control systems: Detectability and performance tradeoffs,” *Automatica*, vol. 82, pp. 251 – 260, 2017.

[92] C. Z. Bai, V. Gupta, and F. Pasqualetti, “On Kalman filtering with compromised sensors: Attack stealthiness and performance bounds,” *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, 2017.

[93] C. Kwon, W. Liu, and I. Hwang, “Security analysis for cyber-physical systems against stealthy deception attacks,” *in Proceedings of the American Control Conference (ACC)*, 2013.

[94] Y. Chen, S. Kar, and J. M. F. Moura, “Optimal attack strategies subject to detection constraints against cyber-physical systems,” *IEEE Transactions on Control of Network Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[95] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, “Worst-case innovation-based integrity attacks with side information on remote state estimation,” *IEEE Transactions on Control of Network Systems*, vol. PP, no. 99, pp. 1–1, 2018.

[96] ——, “Worst-case stealthy innovation-based linear attack on remote state estimation,” *Automatica*, vol. 89, pp. 117 – 124, 2018.

[97] Y. Sun and Á. Baricz, “Inequalities for the generalized Marcum Q-function,” *Applied Mathematics and Computation*, vol. 203, no. 1, pp. 134 – 141, 2008.

[98] Y. Sun, Á. Baricz, and S. Zhou, “On the monotonicity, log-concavity, and tight bounds of the generalized Marcum and Nuttall Q - functions,” *IEEE Transactions on Information Theory*, vol. 56, no. 3, pp. 1166–1186, 2010.

[99] G. H. Golub and C. F. Van Loan, *Matrix computations.* JHU Press, 2012.

[100] H. Avron, E. Ng, and S. Toledo, “Using perturbed QR factorizations to solve linear least-squares problems,” *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 2, pp. 674–693, 2009.

[101] H. Avron and E. Ng, *A Generalized Courant-Fischer Minimax Theorem.* Technical report. Tel-Aviv University, Tel-Aviv, 2008.

[102] B. D. Anderson and J. B. Moore, *Optimal filtering.* Courier Corporation, 2012.

[103] R. Bobba, K. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. Overbye, “Detecting false data injection attacks on DC state estimation,” *in Proceedings of the First Workshop on Secure Control Systems (CPSWEEK)*, 2010.

[104] T. Kim and H. Poor, “Strategic protection against data injection attacks on power grids,” *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 326–333, 2011.

[105] G. Dan and H. Sandberg, “Stealth attacks and protection schemes for state estimators in power systems,” *in Proceedings of the First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2010.

[106] D. Deka, R. Baldick, and S. Vishwanath, “Data attack on strategic buses in the power grid: Design and protection,” *in Proceedings of the 2014 IEEE PES General Meeting and Conference Exposition*, 2014.

[107] S. Bi and Y. Zhang, “Graphical methods for defense against false-data injection attacks on power system state estimation,” *IEEE Transactions on Smart Grid*, vol. 5, no. 3, pp. 1216–1227, 2014.

[108] X. Liu, Z. Li, and Z. Li, “Optimal protection strategy against false data injection attacks in power systems,” *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–9, 2016.

[109] R. Deng, G. Xiao, and R. Lu, "Defending against false data injection attacks on power system state estimation," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 198–207, 2017.

[110] A. Teixeira, K. Sou, H. Sandberg, and K. Johansson, "Secure control systems: A quantitative risk management approach," *IEEE Control Systems*, vol. 35, no. 1, pp. 24–45, 2015.

[111] J. Milošević, T. Tanaka, H. Sandberg, and K. H. Johansson, "Analysis and mitigation of bias injection attacks against a kalman filter," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 6484–6489, 2017.

[112] T. Summers, "Actuator placement in networks using optimal control performance metrics," *in Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, 2016.

[113] A. Clark, L. Bushnell, and R. Poovendran, "A supermodular optimization framework for leader selection under link noise in linear multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 283–296, 2014.

[114] L. Sela Perelman, W. Abbas, X. Koutsoukos, and S. Amin, "Sensor placement for fault location identification in water networks," *Automatica*, vol. 72, no. C, pp. 166–176, 2016.

[115] R. Rymon, "Search through systematic set enumeration," *in Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, 1992.

[116] L. Wolsey, "An analysis of the greedy algorithm for the submodular set covering problem," *Combinatorica*, vol. 2, no. 4, pp. 385–393, 1982.

[117] N. Jain, J. Koeln, S. Sundaram, and A. G. Alleyne, "Partially decentralized control of large-scale variable-refrigerant-flow systems in buildings," *Journal of Process Control*, vol. 24, no. 6, pp. 798–819, 2014.

[118] S. Kaplan and B. Garrick, "On the quantitative definition of risk," *Risk analysis*, vol. 1, no. 1, pp. 11–27, 1981.

[119] T. Cormen, *Introduction to algorithms.* MIT press, 2009.

[120] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions–I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[121] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of computer computations.* Springer, 1972, pp. 85–103.