Security metrics and allocation of security resources for control systems

JEZDIMIR MILOŠEVIĆ

Doctoral Thesis
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology
Stockholm, Sweden 2020

KTH Royal Institute of Technology
School of Electrical Engineering and Computer Science
Division of Decision and Control Systems
SE-100 44 Stockholm
Sweden

TRITA-EECS-AVL-2020:17
ISBN: 978-91-7873-459-7

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av doktorsexamen i elektro- och systemteknik fredagen den 27 mars 2020 klockan 09.00 i sal Kollegiesalen, Kungliga Tekniska högskolan, Brinellvägen 8, Stockholm.

Tryck: Universitetsservice US AB

Abstract

Achieving a sufficient level of security of control systems is very important, yet challenging. Firstly, control systems operate critical infrastructures vital for our society. Hence, attacks against them can result in dire consequences. Secondly, large numbers of security vulnerabilities typically exist in these systems, which makes them attractive targets of attacks. In fact, several attacks have already occurred. Thirdly, due to their specific nature, securing control systems can be costly. For example, their real time availability requirements complicate the deployment of security measures, and control system equipment with limited computational power is unsuited for many security solutions. Motivated by the necessity of control systems security, we study two security-related applications.

The first application considers classifying and preventing security vulnerabilities. We aim to first characterize the most critical vulnerability combinations in a control system, and then prevent these combinations in a cost-effective manner. To characterize the critical vulnerability combinations, we develop an impact estimation framework. Particularly, we use a physical model of the control system to simulate the impact that attack strategies may have on the physical process. Our framework is compatible with a number of attack strategies proposed throughout the literature, and can be used to estimate the impact efficiently. To prevent critical vulnerability combinations in a cost-effective manner, we develop a security measure allocation framework. The framework includes an algorithm for systematically finding critical vulnerability combinations, and two approaches for allocating security measures that prevent these combinations cost-effectively.

The second application considers actuator security. Actuators are vital components of control systems to protect, since they directly interact with the physical process. To evaluate the vulnerability of every actuator in a control system, we develop actuator security indices. These indices characterize resources that the attacker needs to compromise to conduct a perfectly undetectable attack against each actuator. We propose methods to compute the actuator security indices, show that the defender can improve the indices by allocating additional sensors, and discuss the robustness of the indices. We also study a sensor allocation game based on actuator security indices. The goal of studying this game is to develop a monitoring strategy that improves the indices. We derive an approximate Nash Equilibrium of the game, and present the cases when this approximate Nash Equilibrium becomes exact. We also outline the intuition behind this equilibrium, and discuss the ways to further improve the monitoring strategy from the equilibrium.

Sammanfattning

Att säkerställa reglersystem mot manipulation utifrån är mycket viktigt, men samtidigt utmanande. För det första styr reglersystem kritiska infrastrukturer vars funktionalitet är avgörande för vårt samhälle. Således kan attacker mot dem ha allvarliga konsekvenser. För det andra kan ofta ett stort antal säkerhetsluckor hittas i dessa system, vilket gör dem sårbara för attacker. Faktum är att flera attacker mot reglersystem redan har inträffat. För det tredje, på grund av systemspecifika karaktäristika, kan säkerställandet av dessa reglersystem vara mycket kostsamt. Till exempel komplicerar realtidskrav implementeringen av säkerhetsåtgärder; styrsystemutrustning med begränsad beräkningskraft är inte väl lämpad för säkerhetsåtgärder. Givet de säkerhetskrav moderna reglersystem har överväger vi två säkerhetsapplikationer.

Den första applikationen består av klassifikation och förebyggande av säkerhetsproblem. Vi strävar efter att först karakterisera de mest kritiska sårbarhetskombinationerna i ett reglersystem och sedan förhindra dessa kombinationer på ett kostnadseffektivt sätt. För att karakterisera de kritiska sårbarhetskombinationerna utvecklar vi ett ramverk för påföljdsanalys. Vi använder en fysikalisk modell av reglersystemet för att simulera de effekter som attackstrategier kan ha på den fysiska processen. Vårt ramverk är förenligt med ett antal attackstrategier som övervägs i litteraturen och kan användas för att effektivt uppskatta dessas påföljder. För att förhindra kritiska sårbarhetskombinationer på ett kostnadseffektivt sätt utvecklar vi ett ramverk för allokering av säkerhetsåtgärder. Ramverket inkluderar en algoritm för att systematiskt upptäcka kritiska sårbarhetskombinationer och innefattar även två metoder för att fördela säkerhetsåtgärder på ett kostnadseffektivt sätt.

Den andra applikationen avser ställdonssäkerhet. Ställdon är viktiga reglerkomponenter att skydda eftersom de direkt interagerar med den fysikaliska processen. För att utvärdera varje ställdons sårbarhet utvecklar vi säkerhetsindex för dessa. Detta index karakteriserar de resurser som angriparen behöver för att utföra en oupptäckbar attack mot varje ställdon. Vi föreslår metoder för att beräkna dessa index, visar att försvararen kan förbättra indexen genom att placera ytterligare sensorer och diskuterar relaterade robusthetsfrågor. Vi studerar också ett sensorplaceringsspel baserat på detta säkerhetsindex. Målet med att studera detta spel är att utveckla en övervakningsstrategi som förbättrar säkerhetsindex för ställdon. Vi härleder en approximativ Nash-jämvikt i spelet och presenterar de fall när denna approximativa Nash-jämvikt blir exakt. Vi beskriver också intuitionen bakom denna jämvikt och diskuterar metoder för att ytterligare förbättra övervakningsstrategin baserad på denna jämvikt.

Acknowledgements

I would like to begin with expressing my sincere appreciation towards my main advisor Henrik Sandberg. Thank you for giving me a chance to do PhD under your supervision, guiding me through this challenging process, giving me freedom to tackle research problems I felt passionate about, and for being patient and supportive in difficult moments. I would also like to express appreciation towards my co-advisor Karl Henrik Johansson for inspiring discussions, collaborations, his eye for detail, and keeping his doors always open to me.

Next, I would like to thank to Saurabh Amin for his feedback on my licentiate thesis, and for giving me a chance to visit and collaborate with his group. I am also grateful that I had a chance to collaborate with Mathieu Dahan, Farhad Farokhi, Sebin Gracy, Matias Müller, Crisitian Rojas, Takashi Tanaka, André Teixeira, and David Umsonst. Thank you for sharing your knowledge with me. I learned a lot by working with you.

Special thanks to Mohamed Abdalmoaty, Rijad Alisic, Michelle Chong, Mladen Čičić, Takuya Iwaki, Inês Lourenço, Matias Müller, Rui Oliveira, Marina Oluić, Stefan Stanković, Ellis Stefansson, Emma Tegling, David Umsonst, Yu Wang, and Ingvar Ziemann for proofreading my thesis and helping me to improve it.

I am immensely grateful to my current and former colleagues at the Division of Decision and Control Systems for making PhD life enjoyable, the Serbian community at KTH for making me feel like home, and colleagues from CERCES project for fun meetings and interesting team building activities.

This work has been financially supported by the Swedish Civil Contingencies Agency through the CERCES project and KTH School of Electrical Engineering through the Scholarship of Excellence. Their support is greatly acknowledged.

Last, but not the least, I would like to express my appreciation to my family for their love and support. This thesis is dedicated to you.

Jezdimir Milošević

Contents

Notation

Sets

$\mathbb{R}$	The set of real numbers
$\mathbb{R}^+$	The set of positive real numbers
$\mathbb{R}_{\geq 0}$	The set of non-negative real numbers
$\mathbb{R}^n$	The set of n-dimensional vectors over $\mathbb{R}$
$\mathbb{R}^{n \times m}$	The set of $n \times m$-dimensional matrices over $\mathbb{R}$
$\mathbb{C}$	The set of complex numbers
$\mathbb{N}$	The set of natural numbers
$\mathbb{Z}$	The set of integers
$\mathbb{Z}^-$	The set of negative integers
$\mathbb{Z}_{\geq 0}$	The set of non-negative integers
$\lvert\mathcal{V}\rvert$	The cardinality of the set $\mathcal{V}$
$2^{\mathcal{V}}$	The power set of the set $\mathcal{V}$
$\emptyset$	The empty set

Vectors

1_n	The n-dimensional vector whose elements are equal to one
0_n	The n-dimensional vector whose elements are equal to zero
e_i	The i^{th} vector of the canonical basis of appropriate size
x_i	The i^{th} element of the vector x
$x^{(I)}$	The vector consisting of the elements of the vector x from the set I
x^T	The transpose of the vector x
$\text{supp}(x)$	$= \{i : x_i \neq 0\}$ (the support of the vector x)
$\lVert x \rVert_\infty$	$= \max_i \lvert x_i \rvert$ (the infinity norm of the vector x)
$\lVert x \rVert_2$	$= \sqrt{x^T x}$ (the euclidean ℓ_2-norm of the vector x)

Matrices

I_n	The n-dimensional identity matrix
$0_{n \times m}$	The $n \times m$-dimensional matrix whose elements are equal to zero

$\mathrm{Tr}(A)$	The trace of the matrix A
A^T	The transpose of the matrix A
A^{-1}	The inverse of the matrix A
$\mathrm{rank}[A]$	The rank of the matrix A
$\det(A)$	The determinant of the matrix A
$A \otimes B$	The Kronecker product of the matrices A and B
$A \succeq 0$	$\Leftrightarrow$ The matrix A is positive semi-definite
$A \succ 0$	$\Leftrightarrow$ The matrix A is positive definite
$\mathrm{null}(A)$	The null-space of the matrix A
$A(:, i)$	The i^{th} column of the matrix A
$A(i, :)$	The i^{th} row of the matrix A
$A(:, i:j)$	The matrix that contains the columns $i, i+1, \ldots, j$ of the matrix A
$A(i:j, :)$	The matrix that contains the rows $i, i+1, \ldots, j$ of the matrix A

Signals and systems

$s_i(k)$	The i^{th} element of the vector $s(k)$
$s \equiv 0$	$\Leftrightarrow s(k) = 0$ for all k
$s \not\equiv 0$	$\Leftrightarrow s(k) \neq 0$ for at least one k
$s_{N:M}$	$= [s(N)^T \ \ldots \ s(M)^T]^T$
$s^{(i)}_{N:M}$	The i^{th} element of the vector $s_{N:M}$
$\mathrm{supp}(s)$	$= \cup_{k \in \mathbb{Z}_{\geq 0}} \mathrm{supp}(s(k))$ (the support of the signal s)
$\|s\|_0$	$= \lvert\mathrm{supp}(s)\rvert$ (the ℓ_0 "norm" of the signal s)
$\mathrm{nrank}[G]$	$= \max_{z \in \mathbb{C}} \mathrm{rank}[G(z)]$ (the normal rank of the transfer matrix G)
$G^{(I)}$	The transfer matrix consisting of the columns of the transfer matrix G from the set I

Other notation

$\mathbb{P}(.)$	Probability of a random vector
$\mathbb{E}\{.\}$	Expectation of a random vector
$\mathbb{P}(.\,; d)$	Probability of a random vector parametrized by d
$\mathbb{E}\{.\,; d\}$	Expectation of a random vector with respect to $\mathbb{P}(.\,; d)$
$\mathcal{N}(\mu, \Sigma)$	Gaussian distribution with expectation μ and covariance matrix Σ
$\mathbb{1}_{[\delta]}$	$= 1$ (resp. $= 0$) if the statement δ is true (resp. false)
log	The natural logarithm

List of acronyms

IT	Information Technology
DoS	Denial of Service
FDI	False Data Injection
PLC	Programmable Logic Controller
NE	Nash Equilibrium
KL	Kullback–Leibler
CC	Control Center
CGP	Column Generation Procedure

Chapter 1

Introduction

This thesis is on control system security. In the following, we explain what makes this topic important and interesting to study, introduce four security-related problems addressed in the thesis, and outline our contributions.

1.1 Motivation

Control systems operate physical processes that are vital for our society. Electricity production, oil and gas distribution, water purification, manufacturing, and transportation are just some of the numerous examples of these processes.

As depicted in Figure 1.1, control systems can roughly be divided into three layers: the enterprise layer, the supervisory layer, and the field layer [1, 2]. The enterprise layer is responsible for planning and optimizing the operation of the control system [2]. This layer consists of equipment that can be found in regular Information Technology (IT) systems, and is often connected to other external networks, the Internet, and to the supervisory layer [1]. The supervisory layer is responsible for monitoring and high-level control of the physical process. For example, operators in a control center can monitor, collect, and analyze process information, or take manual control over some of the equipment in the field layer.

This thesis mostly focuses on the field layer, which is responsible for the direct interaction with the physical process. This interaction can be captured by the equations

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + v_x(k), \\ y(k) &= Cx(k) + v_y(k), \end{aligned} \tag{1.1}$$

which describe how the physical process evolves over time. Here, $x(k) \in \mathbb{R}^{n_x}$ are the physical states (e.g., pressures, temperatures, or flows). The measurements $y(k) \in \mathbb{R}^{n_y}$ of these states are collected by the sensors, and sent to the control devices (e.g., programmable logic controllers (PLCs), intelligent electronic devices,

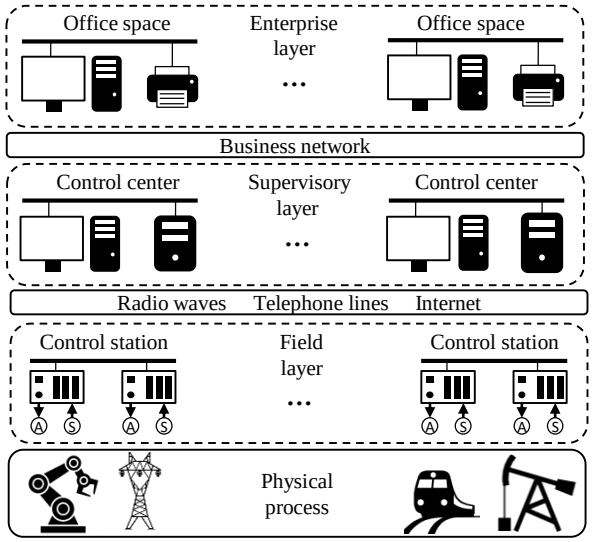

Figure 1.1: A typical three-layered architecture of a control system. Functions of the enterprise layer include scheduling and planning, the supervisory layer is responsible for monitoring and high-level control of the physical process, and the field layer directly interacts with and controls the physical process.

or remote terminal units [3]). Based on these measurements, the control devices compute appropriate control actions $u(k) \in \mathbb{R}^{n_u}$, and send them to actuators (e.g., motors and valves) for execution. Finally, $v_x(k) \in \mathbb{R}^{n_x}$ and $v_y(k) \in \mathbb{R}^{n_y}$ are random processes, which can model noise, disturbances, or faults.

From the description in the previous two paragraphs, one can see that control systems are cyber-physical systems that utilize cyber components to control physical processes. This cyber-physical coupling is precisely the reason why ensuring security of these systems is of utmost importance. Namely, by exploiting cyber-vulnerabilities, an attacker may gain an opportunity to manipulate some of the system components that directly interact with the physical process. He/she can then utilize these components to conduct a malicious attack against the process. How dangerous this could be is best illustrated by attacks that have occurred. We now briefly recall some of the well known attacks.

Example 1.1. *The Maroochy water services breach occurred in the year 2000, in Australia [4, 5]. The series of attacks targeted a sewerage control system and lasted for two months. During this time, one million liters of untreated sewage was released into a stormwater drain. The contaminated water flooded waterways and parks, resulted in the death of marine life, and an unbearable smell spread over the area. The attacks were conducted by an engineer who had previously worked*

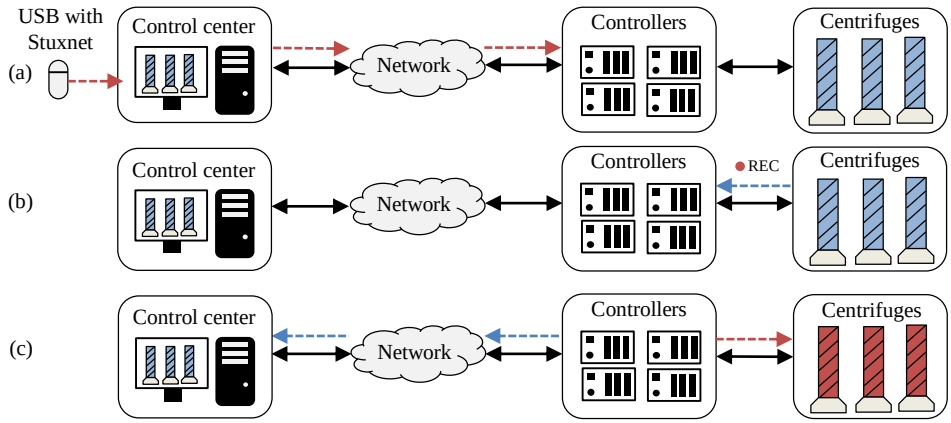

Figure 1.2: Illustration of the Stuxnet attack. Stuxnet: (a) infiltrated a nuclear facility through a USB and compromised controllers of uranium enriching centrifuges; (b) recorded measurements of normal operation; and (c) damaged the centrifuges by issuing malicious control actions while covering these actions by sending previously recorded measurements of normal operation to operators.

on the system. The attacker was familiar with the system's architecture, knew its vulnerabilities, and possessed specialized radio equipment. Using these resources, the attacker managed to alter configuration of control stations, issue malicious radio commands to controllers, and disable alarms.

Example 1.2. *Stuxnet was a computer worm specially designed to sabotage the Iranian nuclear program [6–8]. Since its discovery in 2010, it has attracted considerable attention in the media, industry, and research community. As shown in Figure 1.2, the Stuxnet attack consisted of three stages. In the first stage, Stuxnet infiltrated a uranium enrichment plant through a USB drive, localized controllers of uranium enriching centrifuges, and compromised them (Figure 1.2 (a)). Stuxnet then started recording sensor measurements of normal operation (Figure 1.2 (b)). In the final stage, Stuxnet started issuing malicious control actions while sending the previously recorded measurements to the control center (Figure 1.2 (c)). Thus, the operators falsely believed that the centrifuges were operating normally, while the harmful control signals were damaging the centrifuges.*

Example 1.3. *In 2015, three Ukrainian electricity distribution companies became targets of an organized attack [9]. In this attack, e-mails containing the malware BlackEnergy were sent to employees of the targeted companies. The employees were tricked into installing BlackEnergy, after which the malware enabled the attacker to infiltrate the companies' networks. Next, the attacker localized control centers, gained access to them, and familiarized himself/herself with the control centers'*

environments. The attacker then disabled operators from interfering with the attack while issuing malicious commands to the field layer equipment. These actions caused the blackout of the system, leaving 225,000 customers without electricity.

Having the previous examples in mind, it is perhaps surprising that control system security was neglected in the past [10]. One reason for this is that control systems predominantly used to be isolated from other IT systems. Additionally, hardware and software for control systems were specially designed [11]. Thus, it was security through obscurity that provided a reasonably high level of protection.

However, this is no longer the case. Control systems are now connected to other networks, and the technologies used in control systems are becoming standardized and similar to those used in ordinary IT systems [12]. Moreover, these changes were not accompanied by application of appropriate security solutions. This has resulted in a large number of security vulnerabilities, and made attacks against control systems easier to design. For example, communication protocols commonly used in control systems are often lacking basic security features [13], a control center may be connected directly or indirectly to the Internet without adequate protection [14], and some devices within the system may be easily physically accessible [15].

Additionally, protecting control systems proves to be difficult. In contrast to ordinary IT systems that have a typical life span of two to five years, control systems are designed to last for decades. Thus, support for some of their equipment may not exist anymore [15]. Control systems also have tight real time requirements, which significantly complicates the deployment of security measures [15]. Furthermore, the equipment used in these systems is in many cases resource constrained. Hence, security measures such as encryption that require additional memory and computational resources may cause delays in the system, and thus, result in reduced performance or even instability [16]. Finally, control systems can be highly complex large-scale systems. Therefore, ensuring that each part of such a large-scale system is sufficiently well protected may be prohibitively expensive.

Given the potentially large number of vulnerabilities, difficulties in implementing security measures, and complexity of control systems, aiming to achieve perfect protection of these systems is not realistic. Therefore, it is highly recommended to deploy a security strategy according to a risk management program [12, 14, 15, 18]. As shown in Figure 1.3, this program consists of the risk framing, the risk assessment, the risk response, and the risk monitoring [17].

The risk framing defines a strategy for the other steps of the risk management program [17]. The risk assessment identifies attack scenarios of interest, estimates how likely these scenarios are to occur, and estimates the possible impact if they occur. Once the dangerous attack scenarios are identified, we move to the risk response step, where a cost-effective defense strategy against these scenarios is developed. This strategy may consists of: (i) attack prevention (e.g., by encrypting communication links or improving physical protection of devices [19,20]); (ii) attack

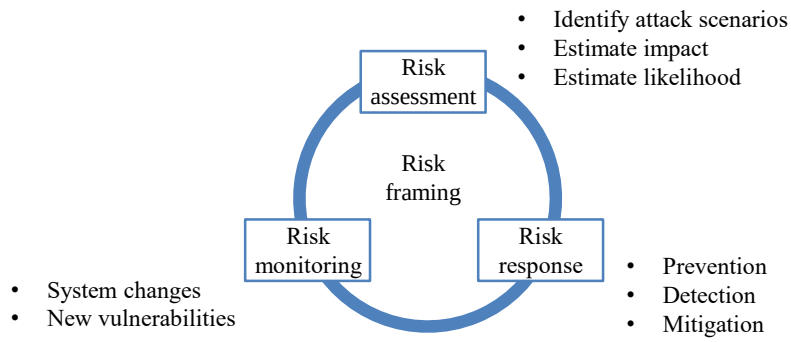

Figure 1.3: The risk management cycle [17]: (i) the risk framing defines a strategy for conducting the remaining three steps of the risk management; (ii) the risk assessment determines the most critical attack scenarios; (iii) the risk response develops a cost-effective defense strategy; and (iv) the risk monitoring evaluates how the risk changes over time.

detection (e.g., by detecting anomalies in the physical behavior of the system or network traffic [21, 22]); and (iii) attack mitigation (e.g., by reconfiguring the system in such a way that the non-attacked components control the process [23, 24]). Finally, the risk monitoring evaluates the effectiveness of the implemented strategy over time, and determines how system changes affect the risk [17].

Motivated by the importance of the risk management program, in this thesis we focus on developing mathematical models and tools that can be used for risk assessment and risk response purposes. Our focus is on two practical applications related to control systems, which are presented in the following.

1.2 Problem formulation

This section introduces two motivating applications that are considered in the thesis, together with the corresponding security problems.

Application 1: Classifying and preventing security vulnerabilities

We are given a set of security vulnerabilities $\mathcal{V}$ within a control system. Elements of $\mathcal{V}$ can model an unprotected communication link, lack of antivirus software on a computer in a control center, or insufficient physical protection of some devices [15]. By exploiting some of these vulnerabilities, the attacker can gain access to sensors and actuators, and then use these components to endanger the physical world. To prevent this, we seek to deploy some of the security measures from a set $\mathcal{M}$. Exam-

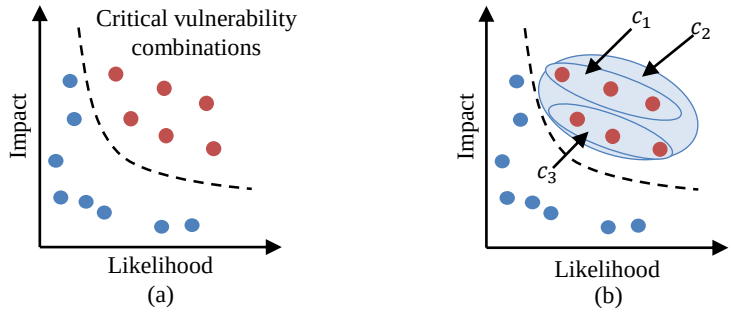

Figure 1.4: The problem of classifying and preventing critical vulnerability combinations can be divided into two sub-problems: (a) classifying critical vulnerability combinations based on their impact and likelihood (each circle represents one vulnerability combination); and (b) selecting the least expensive subset of security measures to prevent the critical vulnerability combinations.

ples of these measures are the encryption of a communication link, the installation and maintenance of anti-virus software, or the deployment of additional physical protection. However, our budget is insufficient to deploy all the security measures. Thus, the problem becomes how to deploy security measures in a cost-effective way. To resolve this problem, we need to develop tools for conducting the risk assessment and the risk response.

In this case, the risk assessment reduces to determining the critical vulnerability combinations that we want to prevent. As shown in Figure 1.4, an important factor in determining these combinations is the impact that can occur when a combination of vulnerabilities is exploited [17]. The impact can be estimated by modeling a control system, and then simulating possible attack strategies [25]. Attack strategies that attract special attention are those that can result in a large impact while staying stealthy from the system operator. Examples of these strategies are optimal False Data Injection (FDI) [26], bias injection [27], and replay [28] attack strategies. However, simpler and easier-to-conduct strategies such as Denial of Service (DoS) [29], rerouting [30], and sign alternation [31] strategies have also been considered. Since attacks against control systems may endanger the physical world, it is natural to use a physical model of the system to estimate the attack impact. Thus, the first problem we tackle in the thesis can be summarized as follows:

P1: How can we utilize a physical model of a control system to estimate the impact of attack strategies in a unified framework?

Remark 1.1. *Besides the impact, one should also consider the likelihood when determining critical vulnerability combinations [17]. The likelihood is typically a score representing the belief of an attack scenario occurring relative to other sce-*

narios [17]. This score can be formed based on expert knowledge [17, 32, 33], or by using tools developed for this purpose [34, 35]. Since estimating the likelihood that a combination of security vulnerabilities is exploited requires significantly different models from those that we use in this thesis, we do not address this problem.

Next, assume that we have a way to determine critical vulnerability combinations. The second step is to select the least expensive subset $M \subseteq \mathcal{M}$ of security measures that prevents all the critical vulnerability combinations (risk response). We name this problem the security measure allocation problem.

The security measure allocation problem is challenging for two reasons. Firstly, to construct this problem, we need to find the critical vulnerability combinations. This is difficult, since the number of vulnerability combinations equals to $2^{|\mathcal{V}|}$. Hence, simply searching through all the combinations is not feasible when the cardinality of $\mathcal{V}$ is large. Secondly, the security measure allocation problem is a combinatorial optimization problem. Thus, it is unclear if we can solve it efficiently. This leads us to the second problem:

P2: Can we develop tools for constructing and solving the security measure allocation problem in a scalable manner?

Application 2: Characterizing and improving the security level of actuators in large-scale control systems

The second application considers actuator security (Figure 1.5). Actuators are very important control system components, since they directly interact with the physical process. Unfortunately, documented attacks have shown that actuators can be compromised by an attacker [4,6,36]. These important components can then be sabotaged, or used to endanger the physical world. Therefore, it is essential to ensure that the actuators are well protected. However, if the control system is large, then it is expected that we are unable to protect all the actuators. Thus, it is crucial to develop tools for characterizing the most vulnerable actuators (risk assessment), and improving their security level in a cost-effective manner (risk response).

For the risk assessment purposes, we introduce an actuator security index δ. The security index $\delta(u_i)$ of an actuator u_i characterizes the minimum resources that the attacker needs to compromise to conduct a perfectly undetectable attack against u_i. Perfectly undetectable attacks are very dangerous, since they do not leave any trace in the sensor measurements [37]. Therefore, an actuator is more (resp. less) vulnerable, if it has a small (resp. large) actuator security index.

However, as shown in this thesis, the index δ is not practical to be used in large-scale control systems. Particularly, δ is difficult to compute and sensitive to system variations that are expected in large-scale systems. Additionally, δ is based on the assumption that the attacker possesses full model knowledge, which may be

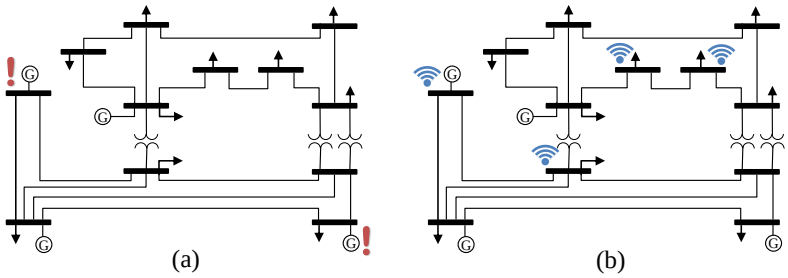

Figure 1.5: Application 2 consists of characterizing and improving the security level of actuators in large-scale control systems. The tools that we develop can for example be used to: (a) characterize vulnerable generators in a power grid; and (b) strategically allocate protected sensors to detect attacks against the generators.

conservative to assume in the case of large-scale control systems. Hence, the third problem we address is as follows:

P3: How to define actuator security indices for large-scale control systems?

Next, assume that we have actuator security indices that are suitable for large-scale systems. Additionally, assume that we determine that some of the actuators have low security indices. The question is then how to increase security indices of these actuators (risk response). We show that one way to achieve this is by allocating protected sensors to detect possible actuator attacks. However, since we focus on large-scale control systems, it is reasonable to assume that a number of protected sensors is insufficient to monitor every state in the system. Thus, the final problem that we address is as follows:

P4: How to strategically allocate a limited number of protected sensors in a large-scale control system such as to improve actuator security indices the most?

1.3 Structure and contributions of the thesis

This section explains the way we tackle the previously introduced problems, describes the structure of the thesis, and outlines our contributions.

Chapter 2: Literature review

Chapter 2 introduces the related literature. We also discuss how our work differs from and extends the existing literature.

Chapter 3: Mathematical preliminaries

Chapter 3 provides the mathematical background.

Chapter 4: Impact estimation

Chapter 4 tackles **P1**. Particularly, we propose and study a novel type of impact estimation problem. We consider two impact metrics: The probability that some of the critical physical states leave a safety region (I_P) and the expected value of the infinity norm of the critical states (I_E). A stealthiness constraint is defined using the Kullback–Leibler divergence (KL–divergence) between the attacked and non-attacked residual sequences. We also introduce constraints on attack signals through which we impose different types of attack strategies.

The main results are as follows. We characterize conditions under which the impact estimation problem is infeasible or its optimal value equals to the maximum impact. When these conditions are not satisfied, we prove that the optimal value of the impact I_P can be computed by solving a set of convex problems. We also derive lower and upper bounds for the metric I_E, which can be computed efficiently. Next, we show that our framework allows us to analyze the impact of the optimal FDI, bias injection, DoS, replay, rerouting, sign alternation, and combined DoS and FDI attack strategies. We also discuss how to use properties of these strategies to more efficiently estimate the impact. Finally, we consider a control system of a chemical plant, and illustrate how our framework can be used to compare security vulnerabilities. We also clarify some of the technical results through examples.

The chapter is based on the publication:

- J. Milošević, H. Sandberg, and K. H. Johansson, "Estimating the impact of cyber-attack strategies for stochastic control systems," IEEE Transactions on Networked Control Systems. Accepted in August 2019.

Related publications are:

- 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, 2018.

- J. Milošević, T. Tanaka, H. Sandberg, and K. H. Johansson, "Analysis and mitigation of bias injection attacks against a Kalman filter," in Proceedings of the 20th IFAC World Congress, 2017.

Chapter 5: Security measure allocation

Chapter 5 addresses **P2**. We propose a security measure allocation framework that is suitable for dynamical models of control systems. Our framework also captures the cyber-physical interaction, which allows us to study the problem of classifying and preventing security vulnerabilities.

The main results are as follows. First, we propose an algorithm that systematically searches for the critical vulnerability combinations and provably returns the combinations necessary to construct the security measure allocation problem. We then establish that the security measure allocation problem is NP-hard, and propose two suboptimal approaches for addressing it. In the first approach, we show how the problem can be simplified and tackled using integer linear program solvers. In the second, we show that the problem possesses a suitable submodular structure. This allows us to apply a polynomial-time greedy heuristic to find a suboptimal solution of the problem with guaranteed performance. We additionally investigate how to optimize these performance. Finally, we demonstrate the applicability of our framework on a control system for temperature regulation.

The chapter is based on the publication:

- 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," International Journal of Robust and Nonlinear Control. Accepted in September 2018.

A related publication is:

- 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, 2017.

Chapter 6: Actuator security indices

Chapter 6 focuses on **P3**. We first introduce a novel type of actuator security index δ. We propose a way to compute δ in small-scale systems, show that δ can be potentially increased by placing additional sensors, and that placement of additional actuators may decrease δ. We then discuss issues that arise in large-scale systems: The index δ is NP-hard to compute, sensitive to system variations that are expected in large-scale systems, and based on the assumption that the attacker knows the entire model of the system.

Next, we introduce the robust security index δ_r, which is based on a structural model of the system [38]. We show that δ_r can be efficiently computed and related

to both the full and limited model knowledge attackers. Since the results we derive imply that actuators with a small value of δ_r are very vulnerable in any system realization, we propose a sensor allocation strategy to increase δ_r. We first show that δ_r is guaranteed to increase if sensors are placed at suitable locations, and then discuss how to systematically allocate unprotected sensors.

Finally, we show how the indices we propose can be used to characterize vulnerable generators in power grids. We also clarify the technical results through examples.

The chapter is based on the publication:

- J. Milošević, A. Teixeira, H. Sandberg, and K. H. Johansson, "Actuator security indices based on perfect undetectability: Computation, robustness, and sensor placement," IEEE Transactions on Automatic Control. Provisionally accepted in October 2019.

Related publications are:

- J. Milošević , H. Sandberg, and K. H. Johansson, "A security index for actuators based on perfect undetectability: Properties and approximation," in Proceedings of the 56th Allerton Conference on Communication, Control, and Computing, 2018.
- J. Milošević, S. Gracy, and H. Sandberg, "On actuator security indices," in Proceedings of the 14th International Conference on Critical Information Infrastructures Security, 2019.

Chapter 7: Allocation of protected sensors

Chapter 7 considers **P4**. We model the sensor allocation problem as a game between a system operator and an attacker. The operator seeks to allocate a limited number of protected sensors to improve actuator security indices, while the attacker seeks to select an actuator with a low value of the security index to attack. We focus on the case where the attacker uses an extended replay strategy, which is inspired by the Stuxnet attack (see Example 1.2).

The main results are as follows. Firstly, we introduce an approximate Nash Equilibrium (NE) of the game, present cases when this NE becomes exact, and outline some game-theoretic interpretations behind this equilibrium. Secondly, we discuss how to further improve the monitoring strategy from the aforementioned equilibrium by deploying additional sensors, focusing on the most vulnerable actuators, and using the so-called Column Generation Procedure (CGP). Finally, we conduct experiments on a benchmark of a large-scale power grid, and show that the tools we propose allow us to construct NE monitoring strategies in a scalable manner.

The chapter is based on the publication:

- J. Milošević, M. Dahan, S. Amin, and H. Sandberg, "A monitoring game based on actuator security indices," under preparation for journal submission.

A related publication is:

- J. Milošević, M. Dahan, S. Amin, and H. Sandberg, "A network monitoring game with heterogeneous component criticality levels," in Proceedings of the 58th IEEE Conference on Decision and Control, 2019.

Chapter 8: Concluding remarks

This chapter summarizes the thesis and outlines possible directions for future work.

The author's contributions and other publications

In the aforementioned articles, the author of the thesis had the most significant role in formulating the problems, solving them, as well as writing the articles. The coauthors have assisted through discussions, suggestions, and text polishing.

We also remark that parts of Chapters 1, 2, and 5 appeared in the licentiate thesis:

- J. Milošević. Model based impact analysis and security measure allocation for control systems. KTH Royal Institute of Technology, 2018.

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, 2016.
- M. I. Müller, J. Milošević, H. Sandberg, and C. R. Rojas, "A risk-theoretic approach to $\mathcal{H}_2$-optimal control under covert attacks," in Proceedings of the 57th IEEE Conference on Decision and Control, 2018.
- S. Gracy, J. Milošević, and H. Sandberg, "Actuator security indices for structural systems," in Proceedings of the American Control Conference, 2020. To appear.

Chapter 2

Literature review

This chapter surveys literature related to the thesis. We first briefly review relevant results from control theory. We then provide a short comparison between IT security and control system security, after which we review work in control system security. Finally, we focus on the literature related to the security problems P1–P4 we tackle in the thesis, and explain how we differ from and extend this literature.

2.1 Related work in control theory

Researchers in the control community study how a feedback system performs in the presence of different types of disturbances (Figure 2.1). So far, a number of approaches for handling these disturbances have been proposed. For example:

(i) In fault tolerant control [39–41], the goal is to detect if a fault has occurred, isolate the fault, and then respond to it. The detection can be achieved by generating the so-called residual signal based on a model of the control system and sensor measurements [40]. A norm of this signal is then compared with some predefined threshold to determine if the fault has occurred or not. The isolation can be achieved by generating a bank of residuals, and then checking which combination of the residuals is active [39]. When the faulty components are isolated, one can respond to the fault by re-configuring the system such that the non-faulty components are used to control the process [23].

(ii) In stochastic control theory [42–44], the objective is to optimally control the system or estimate its trajectory in the presence of stochastic noise. For linear systems corrupted by Gaussian noise, the optimal estimator with respect to a quadratic criteria is a Kalman filter, and the optimal controller with respect to a quadratic criteria is a linear feedback from the state estimate to the control action [42].

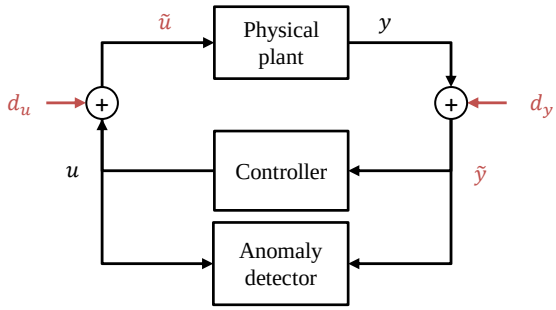

Figure 2.1: A schematic of a control systems in the presence of disturbances. Due to the disturbances, the measurements $\tilde{y}$ received by the controller and the anomaly detector differ from the measurements y collected from the plant, and the corrupted control actions $\tilde{u}$ are applied to the plant instead of the control actions u computed by the controller. While the tools for handling random disturbances such as noise, faults, and packet drops are well studied, novel tools for preventing, detecting, and mitigating malicious strategic attacks are required.

(iii) In robust control [45–47], one tackles the problem of designing controllers that are robust with respect to model uncertainties. One popular approach is $\mathcal{H}_\infty$ controller design. This approach aims at finding a stabilizing controller that minimizes the $\mathcal{H}_\infty$ norm of the transfer function from the bounded disturbances to the outputs of interest [46].

(iv) In the networked control systems literature [48–52], the goal is to investigate how communication network imperfections such as packet drops and delays influence the estimation and control performance. For example, [51] studies the performances of the Kalman filter in the presence of Bernoulli packet drops, and shows that there exists a critical value for the packet arrival rate beyond which the estimation error covariance becomes unbounded. In [52], the packet drops are also assumed to follow a Bernoulli distribution, and the optimal communication and control policies are derived.

Unfortunately, attacks pose a potentially far greater threat to control systems then the above-mentioned disturbances. Namely, disturbances such as faults, noise, or packet drops are random in nature, sometimes assumed bounded or with a known probability distribution, and without a malicious objective to fulfill. In contrast, attacks may be conducted using several components in a coordinated manner, can be designed based on system knowledge, have the intention to harm the system, and can take arbitrary unknown values [53, 54]. Therefore, novel tools need to be developed to protect control systems against malicious attacks.

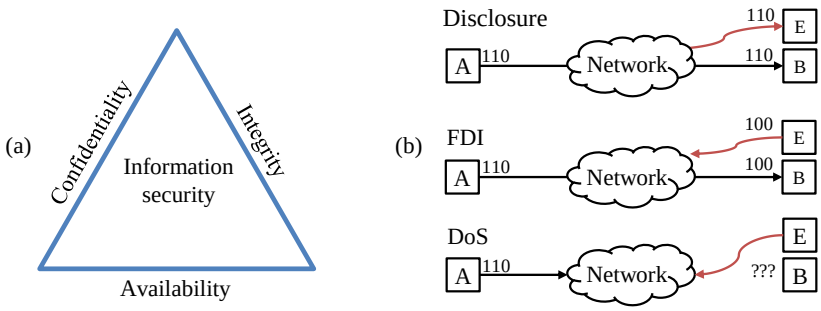

Figure 2.2: (a) The key properties of IT security are confidentiality (data and services can be accessed only by authorized users), integrity (unauthorized change of the information is not possible), and availability (data and services are available upon a user's request). (b) Attacks against aforementioned properties: Disclosure, FDI, and DoS attacks.

2.2 IT security and control system security

As depicted in Figure 2.2 (a), the key properties of IT security are confidentiality, integrity, and availability of the information and services [55]. 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 upon a user's request.

Attacks against these properties are illustrated in Figure 2.2 (b). Attacks against confidentiality are called disclosure attacks. Through these attacks, the attacker gains unauthorized access to the data or service. Attacks against integrity are called FDI attacks. Due to FDI attacks, users may end up using the false information thinking it is true. Finally, attacks against availability are called DoS attacks. In DoS attacks, the attacker disables users from gaining access to data or services.

To defend IT systems against the above-mentioned attacks, a number of security measures have been proposed. Examples of these measures include the encryption of communication links [56], network segmentation using firewalls [57], access control using passwords or smart cards [15], and the deployment of anti-virus software [14].

These security measures can and should be used in control systems as well. However, implementing these measures in control systems can be complicated and expensive, and can lead to undesirable consequences. Additionally, these measures may be insufficient to protect control systems. Some of the reasons are as follows:

(i) Control systems have a significantly longer life span than IT systems [15]. Thus, security solutions for some control system equipment may not exist anymore. This may force us to develop novel security solutions for such

equipment, or to replace it with new equipment [2].

(ii) Some control system equipment may have insufficient computational power. Hence, deploying security measures on this equipment may degrade system performance or destabilize the system [16].

(iii) Control systems have real time availability requirements. Stopping these systems to place security measures can be costly, and needs to be carefully planned well in advance [15].

(iv) Since control systems operate physical processes, attacks against them may have more severe consequences than attacks targeting IT systems. Therefore, additional layers of protection should be placed.

In summary, IT security solutions alone are not enough to protect control systems. These solutions need to be made more compatible with control systems. Additionally, since the deployment of security measures in control systems can be costly, tools for allocating these measures in a cost-effective manner need to be developed. Furthermore, novel layers of defense that protect control systems when IT security measures are breached are required.

2.3 Control system security

Due to the inability of traditional control theoretic and IT solutions to protect control systems against malicious attacks, a novel area of control system security emerged. The pioneering works in this area focused on showing that anomaly detectors developed to detect faults can be ineffective against strategic attacks. One of the first studies that revealed this was [58]. This study considered a power grid monitoring problem, and showed that an attacker can degrade the state estimate while staying undetected by the bad data anomaly detector. This result is illustrated in the following example.

Example 2.1. *A power grid model $y = Cx$ is used for estimating power flows. Here, $y \in \mathbb{R}^{n_y}$ are the sensor measurements and $x \in \mathbb{R}^{n_x}$ are the grid states. Based on the measurements, the state of the grid can be estimated as $\hat{x} = (C^T C)^{-1} C^T y$. Since some of the measurements can be faulty, the operator also generates the residual $r = y - C\hat{x}$. A large (resp. small) magnitude of r indicates that faults are present (resp. not present). Assume now that the measurements are subject to an attack, so the operator receives the corrupted measurements $\tilde{y} = Cx + a$ instead of y. Additionally, let the attack be given by $a = C\tilde{x}$, where $\tilde{x} \in \mathbb{R}^{n_x}$. We then have*

$$\hat{x} = (C^T C)^{-1} C^T (C\tilde{x} + Cx) = x + \tilde{x},$$
$$r = C\tilde{x} + Cx - C(C^T C)^{-1} C^T (C\tilde{x} + Cx) = C\tilde{x} + Cx - C\tilde{x} - Cx = 0_{n_y}.$$

Hence, the attack simultaneously degrades the quality of the estimate without leaving any trace in the residual signal.

Motivated by the previous result, a number of other limitations that attacks impose have been derived. For instance, Fawzi *et al.* considered an estimation problem for a noiseless linear dynamical system, and characterized the maximum number of attacked sensors for which the correct state can still be estimated [53]. Pasqualetti *et al.* considered an attack detection problem in absence of noise, and proved that attacks that excite the zero-dynamics of the system cannot be detected by a wide range of monitors [59]. Limitations for the detection of attacks in stochastic systems have also been investigated [60–62]. Attacks also pose restrictions on reaching a consensus among agents in networked control systems [63–65]. For example, Sundaram and Hadjicostis studied the resilience of linear iterative strategies [64]. They showed that if a number of vertex-disjoint paths from an agent x_j to an agent x_i is less than or equal to $2n$, then n malicious agents may conduct an attack such that x_i cannot recover x_j's value.

The threat of attacks was also verified through experiments [66–70]. For example, Teixeira *et al.* considered the above-mentioned power grid monitoring problem [66]. The experiments conducted on a realistic energy management software showed that it is indeed possible to design undetectable attacks that degrade the estimation quality. Another interesting experiment was reported in [67]. There, the authors considered an attacker with the full model knowledge and the ability to manipulate some of the measurements and control actions. Based on this attacker model, several attacks against a control system operating a water canal network were designed. The experiment demonstrated that these attacks can cause water pilfering from the canal system without being detected.

For the above-mentioned reasons, it is not surprising that security-related problems have attracted considerable attention within the control community [71–75]. Some of the classical problems previously considered in an attack-free setting have been extended to account for the presence of attacks. Examples include the design of attack resilient controllers [76–83], anomaly detectors [84–90], estimators [53,91–94], and consensus protocols [63–65,95–97]. Many other problems have also been studied within the area. Examples include the four problems we address in this thesis. In what follows, we focus on the literature treating these problems.

2.4 Impact estimation

To better understand the consequences of attacks and to better protect against them, one needs to develop suitable models of attack strategies. Attack strategies in which the attacker avoids being detected by anomaly detection mechanisms have attracted most of the attention so far [26–28, 98–103]. Examples of these strategies include optimal FDI [26, 98], bias injection [27, 103], and replay [28] attack strate-

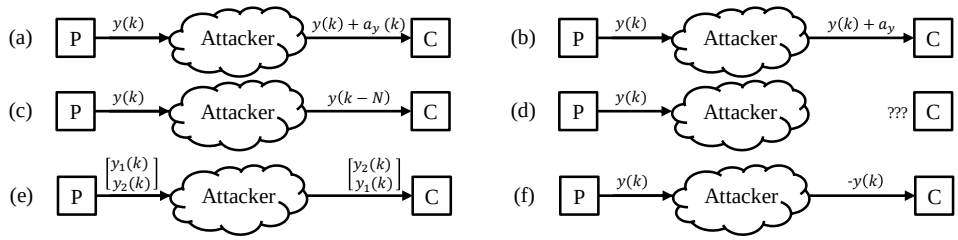

Figure 2.3: Attack strategies that the attacker can implement upon compromising the communication link between the plant (P) and the controller (C). (a) Optimal FDI strategy: The attacker injects a carefully designed attack sequence into the measurements. (b) Bias injection strategy: The attacker injects a carefully designed constant bias into the measurements. (c) Replay strategy: The attacker sends previously recorded measurements of normal operation to C. (d) DoS strategy: The attacker blocks the communication between P and C. (e) Rerouting strategy: The attacker reroutes the measurements coming from P. (f) Sign alternation strategy: The attacker changes the sign of the measurements.

gies (see Figure 2.3 (a)–(c)). However, less complex strategies such as DoS [2, 29], rerouting [30, 104], and sign alternation [31] attack strategies have also been considered (see Figure 2.3 (d)–(f)). Although not necessarily stealthy, these strategies are easier to conduct, which makes them important to study.

We are interested in estimating the impact of above-mentioned attack strategies. Initial studies on the impact estimation problem considered estimating the impact of attacks that remain undetected by a chi-square detector [101, 103, 105–108]. The focus of these studies was on the optimal FDI and bias injection attack strategies. For example, [105] considers the Kalman filter equipped with the chi-squared detector. The impact in this work was defined through the performance degradation of the Kalman filter, and an algorithm that computes the upper and lower bounds of the impact was proposed. Following [105], Murguia *et al.* proposed ellipsoidal bounds that are easier to compute [106]. Another extension of [105] was presented in [107], where the authors observed the performance degradation of the Kalman filter in the presence of an authentication mechanism.

The impact estimation problem for other types of detectors have also been studied [26, 109–113]. These works were also focused on optimal injection attack strategies. In this set of literature, the work especially relevant to us is [26], which considered a CUSUM detector, and used the infinity norm of critical states to define the impact metric. An important result derived in [26] is that the optimal value of the impact with respect to the infinity norm metric can be computed by solving a set of convex problems. This useful property of the infinity norm metric was also recognized in [20, 114], where the impact with respect to the infinity norm metric was computed by solving a set of linear programs. However, the studies [20, 26, 114]

neglect the influence of noise and do not propose a substitute for the infinity norm metric that can be used in stochastic systems.

Our work presented in Chapter 4 differs from and extends the previous literature in the following aspects:

(i) As opposed to the works on the infinity norm metric [20, 26, 114], we focus on more general stochastic systems. Particularly, we propose two metrics that can substitute the infinity norm metric in these systems, and study the impact estimation problem based on these metrics.

(ii) Compared to the studies on the impact estimation problem that focus on optimal injection attack strategies [26, 101, 103, 105–113], our analysis is more general. Namely, our analysis covers both the optimal FDI and bias injection attack strategies, as well as the DoS [2, 29], replay [28], rerouting [30, 104], sign alternation [31], and combined DoS and FDI [115, 116] attack strategies.

(iii) The studies [26, 101, 103, 105–113] focus their analysis on particular types of anomaly detectors. Thus, the impact analysis is carried out for every detector separately. In our work, we use the idea from [60, 62, 117, 118], and model the stealthiness constraint using the KL–divergence. In this way, we make our analysis independent of the anomaly detector choice.

2.5 Security measure allocation

Previous works on the security measure allocation problem have mostly been inspired by power grid monitoring [58]. In [58], the grid was modeled as a static linear system, and a particular combination of an estimator and an anomaly detector was used. It was shown that if the attacker compromises a right combination of sensors, then he/she is able to conduct a stealthy attack [58]. In this case, the security measure allocation problem can be formulated as securing some of the existing sensors, and/or placing additional secured sensors, to make stealthy attacks harder to conduct. To solve this problem in large-scale power grids, many different approaches have been taken [19, 119–125].

For example, Bobba *et al.* proved that it suffices to protect the set of so-called basic sensors to prevent stealthy attacks, and used LU decomposition to find these sensors [119]. Kim and Poor approximated the attacker's resources needed to conduct a stealthy attack with the optimal value of a linear program, and used greedy algorithms to select sensors to secure such as to maximize these resources [120]. Vuković *et al.* allocated security measures based on the so-called security index using an iterative algorithm [19]. This work also introduced more detailed models of communication networks and security measures compared to the other studies.

The problem of allocating security measures in dynamical control systems has attracted less attention. In [109], several methods for estimating the attack impact were introduced. It was also hinted that these methods can be used to select sensors/actuators to protect. In [126], a flexible risk model based on which security measures can be allocated was proposed. Additionally, it was discussed how to use this model to determine where to invest the security budget. In [103], an estimation problem in presence of bias injection attacks was considered. To mitigate the impact of these attacks, a method for selecting sensors to secure was introduced.

The above-mentioned literature can be extended in the following directions:

(i) A framework for allocating security measures based on dynamical models of control systems is lacking. The studies [103, 109, 126] do mention this problem, but do not provide a systematic way to allocate security measures when their number is large. Moreover, the tools developed for allocating security measures in power grids cannot be straightforwardly extended to dynamical systems, since they rely heavily on the model setup from [58].

(ii) To tackle the problem of classifying and preventing security vulnerabilities presented in the introduction, we need a model that captures both the cyber and the physical part of a control system. However, such a model is missing. The issue is partially addressed in [19], but the authors were mostly concerned with modeling the interaction between the communication infrastructure and the physical process.

(iii) The optimality of the approaches for solving the security measure allocation problem is rarely discussed. Hence, objective values obtained using these approaches can be arbitrarily far from the optimal value.

Chapter 5 addresses these issues. Particularly, we propose a security measure allocation framework that is suitable for dynamical systems. Our framework captures the cyber-physical interaction, which allows us to study the problem of classifying and preventing security vulnerabilities. Additionally, the framework includes tools for systemically constructing and solving the security measure allocation problem when the number of vulnerabilities and security measures is large. Furthermore, we show that the security measure allocation problem has a suitable submodular structure in our case. This allows us to use a polynomial time algorithm to compute a suboptimal solution of the problem with performance guarantees.

2.6 Actuator security indices

The first security index α was introduced to localize the most vulnerable sensors in a power grid [127]. Particularly, the security index $\alpha(y_i)$ is defined for every sensor

y_i, and it equals to the optimal value of the following optimization problem:

$$\begin{aligned} &\underset{x}{\text{minimize}} \ \ \|y\|_0 \\ &\text{subject to} \ \ y = Cx, \ y_i \neq 0. \end{aligned} \tag{2.1}$$

Here, $y \in \mathbb{R}^{n_y}$ are the sensor measurements, $x \in \mathbb{R}^{n_x}$ are the grid states, and $C \in \mathbb{R}^{n_y \times n_x}$ models the grid. The first constraint imposes that the attacked sensor measurements need to correspond to a feasible power grid state, which ensures attack stealthiness (see Example 2.1). The second constraint ensures that y_i is attacked. Thus, $\alpha(y_i)$ characterizes the minimum number of sensors needed to attack y_i and remain stealthy. Naturally, sensors with low values of α are the most vulnerable. Once these sensors are localized, the operator can allocate additional security measures to protect them [19].

Although α proved to be a useful tool for both vulnerability analysis and development of defense strategies, there exist two issues related to this index. Firstly, α is NP-hard to compute [128]. This issue is addressed in [128–132]. For instance, [129] proposes an upper bound on α that can be computed in polynomial time by solving the minimum s-t cut problem. Additionally, this bound is tight in several cases of interest. Secondly, α is defined for static systems and cannot be used to characterize vulnerable components in dynamical systems. In contrast to the first issue that is well studied, the second has been addressed only by a few works [133–135].

In [133], Chong and Kuijper introduced the index that can characterizes vulnerability of the entire system, but not system components such as actuators. In [135], Zhao and Pasqualetti introduced the notion of nodal energy. It was also hinted that this notion can be used as a measure of security and robustness in the control system. However, no connection was made with any strategic attack. In [134], Sandberg and Teixeira proposed a security index that resembles the static security index α. This index is based on the definition of undetectability [59], and characterizes the vulnerability of sensors and actuators within the system. Yet, this work neither addressed the problems that appear in large-scale systems nor explained how this index can be used for defense purposes.

Chapter 6 introduces the actuator security indices δ and δ_r. The main difference compared to the studies on the static index α is that our indices are defined for dynamical systems, and can be used to characterize the vulnerability of actuators. We also show that some of the conclusions derived for the index α can be extended to our dynamical indices. For example, the problem of computing the robust security index δ_r can be formulated as the minimum s-t cut problem.

Compared to the related study on the dynamic index [134], our work differs in three aspects. Firstly, while [134] introduced the index based on the definition of undetectability, our indices are based on the definition of perfect undetectability. Hence, a different approach is needed to analyze and compute these indices. Additionally, perfectly undetectable attacks are more dangerous than undetectable attacks, since

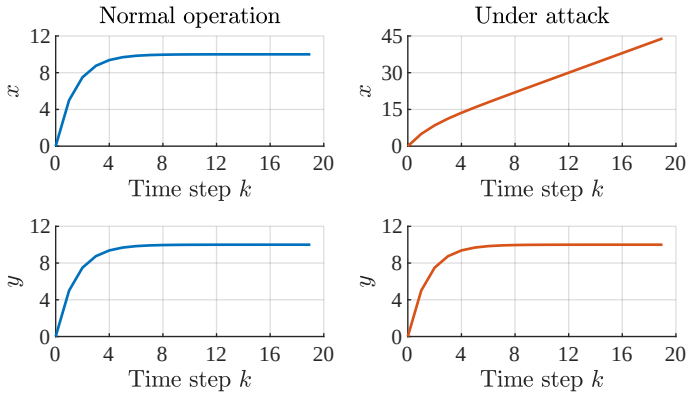

Figure 2.4: An illustration of a perfectly undetectable attack. Observe that the attack manages to drive the state x far away from the origin, while not leaving any trace in the sensor measurement y.

they do not leave traces in the measurements (see Example 2.2). Secondly, in contrast to [134], we discuss the issues that appear in large-scale systems and provide a possible approach to overcome these issues. Thirdly, we discuss how to improve the indices by placing additional sensors.

Example 2.2. *Consider the system*

$$\begin{aligned} x(k+1) &= 0.5x(k) + u(k) + a_1(k), \\ y(k) &= x(k) + a_2(k). \end{aligned}$$

Let $u(k) = 5$ *and* $a_1(k) = k$ *for every* $k \in \mathbb{Z}_{\geq 0}$, $a_2(k) = 0.5a_2(k-1) - a_1(k-1)$, *and* $x(0) = 0$. *The trajectories of the system state and the measurement in absence and under the attack are shown in Figure 2.4. We observe that the attack is perfectly undetectable, since it does not leave any trace in the sensor measurement.*

Our work is also related to the studies on perfectly undetectable attacks [37, 136]. In [136], perfectly undetectable attacks were introduced, and algebraic conditions for the existence of these attacks were derived. These conditions were generalized in [37]. The study [37] also derived graph theoretic conditions for the existence of perfectly undetectable attacks, and proposed a way to design a system such as to make perfectly undetectable attacks harder to conduct.

In our work, we rely on the aforementioned algebraic and graph theoretic conditions to compute the indices δ and δ_r. However, these conditions need to be extended to be applied in our study. Furthermore, the studies [37, 136] do not consider actuator security indices, do not discuss attackers with limited model knowledge, and do not study sensor allocation strategies for improving security indices.

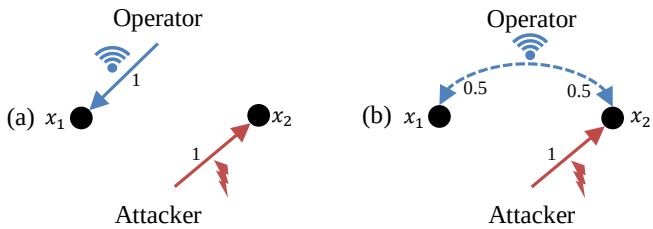

Figure 2.5: The problem of monitoring two decoupled states using one sensor. (a) If the operator uses a static placement, then he/she can monitor only one state. The strategic attacker then targets the other, inflicting one unit of damage to the operator. (b) If the operator uses a mixed strategy and monitors each state with probability 0.5, then he/she ensures the average worst case loss of 0.5 units.

2.7 Allocation of protected sensors

So far, a number of sensor allocation problems have been considered within the control community. The objective of the allocation can be to minimize the estimation error [137], achieve optimal coverage [138], detect and isolate faults [139], or improve the system's security level [140]. From this set of literature, the work most relevant to us is [140]. This work introduced and analyzed actuator security indices, and proposed two static sensor allocation approaches to improve these indices. While we also consider sensor allocation strategies based on actuator security indices, we adopt a game theoretic approach to this problem and focus on randomized (mixed) strategies. Thus, the theoretical analysis used to derive the main technical results completely differs from the one in [140].

The existing works on game theoretic sensor allocation considered developing both static [141–143] and mixed strategies [144, 145]. We focus on mixed strategies, which are recognized to be more effective than static when the number of sensors to allocate is limited [144, 145]. A simple example from [145] illustrates why.

Example 2.3. *Consider the system consisting of two physical states $\mathcal{X} = \{x_1, x_2\}$ shown in Figure 2.5. The operator can monitor only one state at a time. Additionally, the states are decoupled, so measuring one state does not give any information about the other. Assume that an attack against a state that is not measured inflicts one unit of damage to the operator. By using a static placement, the operator can monitor at most one state. Hence, the strategic attacker targets the other, inflicting one unit of damage to the operator (Figure 2.5.(a)). However, by using a mixed strategy where the sensor measures each state with probability 0.5, the operator ensures the average worst case loss of 0.5 units (Figure 2.5.(b)).*

Our game is related to the one in [144], where the operator seeks allocating sensors to maximize the number of detected attacks, while the attacker seeks attacking com-

ponents with the opposite objective. The authors assumed homogeneous system components, and derived an approximate NE (ϵ-NE) of the game. In this equilibrium, the monitoring (resp. attack) strategy is derived from a solution to the minimum set cover (resp. maximum set packing) problem. A similar approach for characterizing equilibria was used in [146–148], but for specific models and player's resources. Additionally, [144] proposed the numerical CGP as a way to improve the set cover strategy. However, CGP was neither implemented nor tested in [144], since the strategies performed well.

We differ from and extend [144] as follows:

(i) We consider a more general game where system components (actuators) have heterogeneous security indices associated to them;

(ii) We introduce ϵ-NE strategies that reveal some fundamental differences of our game and the game from [144];

(iii) We show that the strategies from [144] are a special case of our strategies when the security indices are homogeneous;

(iv) We show that CGP can be used in our game as well, implement it, and test it on a benchmark of a large-scale power grid.

Chapter 3

Mathematical preliminaries

This chapter introduces the mathematical preliminaries required to follow the thesis. Section 3.1 presents some terminology from graph theory. Section 3.2 considers control system models. Section 3.3 introduces relevant results and terminology from optimization theory. Section 3.4 revisits the KL–divergence. Section 3.5 reviews needed results from game theory.

3.1 Graph theory

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph with a set of nodes $\mathcal{V}$ and a set of directed edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Nodes v and w are adjacent if there exists an edge between them and non-adjacent otherwise. A directed path from v_1 to v_n is a sequence of nodes $v_1, v_2, \ldots, v_n$, where $(v_k, v_{k+1}) \in \mathcal{E}$ for all $k \in \{1, \ldots, n-1\}$. A directed path that does not contain repeated nodes is a simple directed path. The in-neighborhood of a node v is defined by $\mathcal{N}_v^{\text{in}} = \{w \in \mathcal{V} : (w, v) \in \mathcal{E}\}$. A vertex (resp. an edge) separator of non-adjacent nodes v and w is a subset of nodes $V \subseteq \mathcal{V} \setminus \{v, w\}$ (resp. edges $E \subseteq \mathcal{E}$) whose removal eliminates all the directed paths from v to w. If each edge (v, w) is assigned with a weight $c_{vw} \in \mathbb{R}$, then the cost of an edge separator E is $\sum_{(v,w)\in E} c_{vw}$. Since edge and vertex separators are important for the derivation of some results in Chapters 6 and 7, we illustrate them in Figure 3.1.

3.2 Linear time-invariant systems and structured systems

Consider a linear time-invariant system

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ev(k), \\ y(k) &= Cx(k) + Du(k) + Fv(k), \end{aligned} \tag{3.1}$$

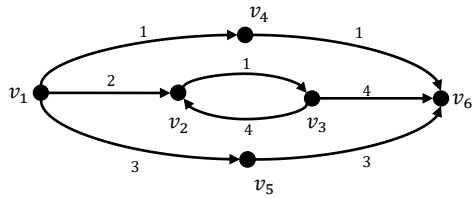

Figure 3.1: A vertex separator of v_1 and v_6 is $V = \{v_2, v_4, v_5\}$. An edge separator of v_1 and v_6 is $E = \{(v_1, v_2), (v_1, v_4), (v_1, v_5)\}$. The cost of E is 6.

where $x(k) \in \mathbb{R}^{n_x}$ are the system states, $u(k) \in \mathbb{R}^{n_u}$ are the control actions, $y(k) \in \mathbb{R}^{n_y}$ are the sensor measurements, and $v(k) \in \mathbb{R}^{n_v}$ are the disturbances present in the system. We utilize linear-time invariant systems to estimate the impact of attacks (Chapter 4), and to study the actuator security index δ (Chapter 6).

A convenient property of the system (3.1) that we exploit on several occasions is that the system states $x(k)$ and the sensor measurements $y(k)$ can be written as the sum of responses to the initial states $x(0)$ and each of the inputs. Particularly, let us define the operators $\mathcal{C}_N(P, Q) = \begin{bmatrix} P^{N-1}Q & \dots & PQ & Q \end{bmatrix}$,

$$\mathcal{O}_N(P, R) = \begin{bmatrix} R \\ RP \\ \vdots \\ RP^N \end{bmatrix}, \quad \mathcal{T}_N(P, Q, R, S) = \begin{bmatrix} S & 0_{p\times m} & \dots & 0_{p\times m} \\ RQ & S & \dots & 0_{p\times m} \\ \vdots & \vdots & \ddots & \vdots \\ RP^{N-1}Q & RP^{N-2}Q & \dots & S \end{bmatrix},$$

where $P \in \mathbb{R}^{n\times n}$, $Q \in \mathbb{R}^{n\times m}$, $R \in \mathbb{R}^{p\times n}$, and $S \in \mathbb{R}^{p\times m}$. The system states at time step $N \in \mathbb{N}$ are then given by

$$x(N) = A^N x(0) + \mathcal{C}_N\big(A, B\big)u_{0:N-1} + \mathcal{C}_N\big(A, E\big)v_{0:N-1}, \tag{3.2}$$

and the measurements $y_{0:N}$ by

$$y_{0:N} = \mathcal{O}_N(A, C)x(0) + \mathcal{T}_N\big(A, B, C, D\big)u_{0:N} + \mathcal{T}_N\big(A, E, C, F\big)v_{0:N}. \tag{3.3}$$

Next, we consider a simplified version of the system (3.1)

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k) + Du(k), \end{aligned} \tag{3.4}$$

and introduce structural representation of this system [38]. The structural system is defined by the binary matrices $[A] \in \{0,1\}^{n_x\times n_x}$, $[B] \in \{0,1\}^{n_x\times n_u}$, $[C] \in \{0,1\}^{n_y\times n_x}$, and $[D] \in \{0,1\}^{n_y\times n_u}$. A realization of the system (3.4) (given by the matrices A, B, C, D) is a feasible realization of the structural system $[A], [B], [C], [D]$, if we can obtain it from the latter by replacing values equal to one with any number in $\mathbb{R}$. In the next example, we further clarify what we mean by a feasible realization.

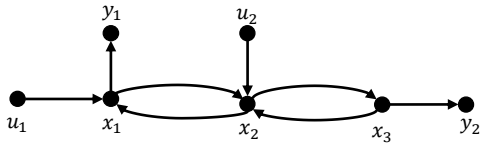

Figure 3.2: The structural graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ that corresponds to the structural system from Example 3.1.

Example 3.1. *Let the structural system be given by*

$$[A] = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad [B] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad [C] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad [D] = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \quad (3.5)$$

Consider the following realizations of A, B, C, D:

$$\textit{(i)}\ A = \begin{bmatrix} 0 & 2 & 0 \\ 0.2 & 0 & 7 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 0 \\ 0 & 2 \\ 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix};$$

$$\textit{(ii)}\ A = \begin{bmatrix} 3 & 1 & 0 \\ 0.1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

The realization (i) is a feasible realization of the structural system (3.5)*. Note that having $A(3,2) = 0$ although $[A](3,2) = 1$ is permitted. However, the realization (ii) is not a feasible realization of* (3.5) *because $A(1,1) = 3$ and $[A](1,1) = 0$.*

The idea behind the structural analysis is to derive properties that hold for all, or almost all, feasible realizations of the system (3.4) by analyzing the structural system [A],[B],[C],[D]. To derive such results, it is sometimes convenient to represent the structural system by the structural graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

The set of nodes of this graph is $\mathcal{V} = \mathcal{X} \cup \mathcal{U} \cup \mathcal{Y}$, where $\mathcal{X} = \{x_1, \ldots, x_{n_x}\}$ is the set of states, $\mathcal{U} = \{u_1, \ldots, u_{n_u}\}$ is the set of actuators, and $\mathcal{Y} = \{y_1, \ldots, y_{n_y}\}$ is the set of sensors. The set of edges is given by $\mathcal{E} = \mathcal{E}_{ux} \cup \mathcal{E}_{xx} \cup \mathcal{E}_{xy} \cup \mathcal{E}_{uy}$, where $\mathcal{E}_{ux} = \{(u_j, x_i) : [B](i,j) \neq 0\}$ is the set of edges from the actuators to the states, $\mathcal{E}_{xx} = \{(x_j, x_i) : [A](i,j) \neq 0\}$ is the set of edges between the states, $\mathcal{E}_{xy} = \{(x_j, y_i) : [C](i,j) \neq 0\}$ is the set of edges from the states to the sensors, and $\mathcal{E}_{uy} = \{(u_j, y_i) : [D](i,j) \neq 0\}$ is the set of edges from the actuators to the sensors. The following example further clarifies the structural graph.

Example 3.2. *Consider the structural system from Example 3.1. The corresponding structural graph is shown in Figure 3.2.*

This graph representation of the structural system is beneficial for several reasons [38]. Firstly, it captures the same information as the structural matrices. Secondly, many of the system properties of interest can be characterized through easily understandable graph conditions. For example, graph conditions for determining structural controllability [149], observability [150], and many other system properties [38] are well-known. Thirdly, some of these graph conditions can be verified efficiently, which is especially useful when studying properties of large-scale systems. In Chapter 6, we utilize the structural systems and their convenient graph representations to study the robust security index δ_r.

3.3 Optimization theory

We now briefly revisit some results and terminology from convex and submodular optimization theory that we use in the thesis. We mostly rely on the books [151,152].

We begin by introducing some basic notation and terminology concerning general optimization problems. An optimization problem is usually written as follows:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0, \quad \forall i \in \{1,\dots,m\}, \\ & h_i(x) = 0, \quad \forall i \in \{1,\dots,p\}, \end{aligned} \tag{3.6}$$

where $x \in \mathcal{X}$ are the decision variables, $f_0 : \mathcal{X} \to \mathbb{R}$ is the objective function, $f_i : \mathcal{X} \to \mathbb{R}$ for all $i \in \{1,\dots,m\}$ are the functions that determine inequality constraints, and $h_i : \mathcal{X} \to \mathbb{R}$ for all $i \in \{1,\dots,p\}$ are the functions that determine equality constraints. An optimization problem is continuous (resp. discrete) if the decision variables are continuous (resp. discrete). Note that the problem aiming to maximize the objective function $f_0(x)$ can be formulated as the problem of minimizing the function $-f_0(x)$.

The optimal value f_0^* of the problem (3.6) is defined by

$$f_0^* = \inf\big\{ f_0(x) \mid f_i(x) \leq 0,\ \forall i \in \{1,\dots,m\},\ h_i(x) = 0,\ \forall i \in \{1,\dots,p\} \big\}.$$

A point $x \in \mathcal{X}$ is a feasible point of the problem (3.6) if it satisfies all the inequality and equality constraints. That is, if the following holds:

$$f_i(x) \leq 0, \quad \forall i \in \{1,\dots,m\}, \quad h_i(x) = 0, \quad \forall i \in \{1,\dots,p\}.$$

If for every feasible point x we also have that $-x$ is a feasible point, then the constraints are said to be symmetric. A point $x^* \in \mathcal{X}$ is a solution to the problem (3.6) if x^* is a feasible point and $f_0(x^*) = f_0^*$.

The problem (3.6) is infeasible if there are no feasible points, and unbounded if $f_0^* = -\infty$. Two optimization problems are equivalent if a solution of one can be readily recovered from a solution of the other, and vice versa.

In this thesis, we study several optimization problems, both continuous and discrete. Our objective is to solve, or find a good approximate solutions, of these problems efficiently. Yet, this is generally possible only for certain instances of optimization problems. In the following, we revisit some of the instances that we encounter.

3.3.1 Convex optimization problems

Prior to introducing convex problems, we define convex sets and functions. A set C is convex if

$$\theta x_1 + (1-\theta)x_2 \in C$$

holds for all $x_1, x_2 \in C$ and all $\theta \in [0,1]$. A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if its domain is a convex set and

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta) f(x_2)$$

holds for all x_1, x_2 in the domain of f and all $\theta \in [0,1]$. A convex optimization problem can then be written as follows:

$$\begin{array}{ll} \underset{x}{\text{minimize}} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \qquad \forall i \in \{1,\ldots,m\}, \\ & a_i^T x = 0, \qquad \forall i \in \{1,\ldots,p\}, \end{array}$$

where $f_0, f_1, \ldots, f_m$ are convex functions and $a_1, \ldots, a_p$ are real vectors of appropriate size. Algorithms that solve convex problems efficiently are well-known [153]. We utilize this fact to estimate the impact of attacks efficiently (Chapter 4).

3.3.2 Submodular optimization problems

While convexity is important for continuous optimization, submodularity is important for discrete optimization [154]. Namely, certain classes of combinatorial optimization problems that have submodular structures can be approximately solved with performance guarantees in polynomial time. To define the problems relevant to our study, we first define submodular functions.

Let $\mathcal{A}$ be a finite non-empty set, and $F : 2^{\mathcal{A}} \to \mathbb{R}$ be a set function. The set function F is submodular if

$$F(A \cup a) - F(A) \geq F(A' \cup a) - F(A')$$

holds for all $A \subseteq A'$ and all $a \in \mathcal{A} \setminus A'$. We also recall that F is nondecreasing if $F(A) \leq F(A')$ holds for all $A \subseteq A'$.

In words, if a set function is submodular, then adding an element to a set A results in a larger gain than adding it to a set containing A. For this reason, submodularity is

Algorithm 3.1 A greedy heuristic for Problem 3.1 [156]

Input: $\mathcal{A} = \{a_1, \ldots, a_n\}$, F, $c_{a_1}, \ldots, c_{a_n}$
Output: A_G
$A_G \leftarrow \emptyset$
while $F(A_G) < F(\mathcal{A})$ **do**
  $a^* \leftarrow \operatorname{argmin}\{c_a/(F(A_G \cup a) - F(A_G)) : a \in \mathcal{A} \setminus A_G\}$
  $A_G \leftarrow A_G \cup a^*$
end while

often called a diminishing returns property. The following properties of submodular functions are well-known [155].

Lemma 3.1. *If $F_1, \ldots, F_n$ are submodular and nondecreasing set functions, then $\sum_{i=1}^{n} F_i(A)$ is a submodular and nondecreasing set function.*

Lemma 3.2. *Let $c \in \mathbb{R}$ and F be a set function. If F is submodular and nondecreasing, then $g(A) = \min\{F(A), c\}$ is a submodular and nondecreasing set function.*

In what follows, we briefly introduce some of the problems with submodular structure that we encounter in the thesis.

Minimizing a linear set function subject to a submodular constraint

We first consider the following problem.

Problem 3.1. *Minimizing a linear function subject to a submodular constraint*

$$\begin{aligned} &\underset{A}{\text{minimize}} && \sum_{a \in A} c_a \\ &\text{subject to} && F(A) = F(\mathcal{A}). \end{aligned}$$

Here, $c_a \in \mathbb{R}^+$ for every $a \in \mathcal{A}$, F is a submodular, nondecreasing, and integer valued set function, and $F(\emptyset) = 0$. We encounter Problem 3.1 in Chapter 5, where we show that the security measure allocation problem is an instance of Problem 3.1.

The optimal value of Problem 3.1 can be approximated by Algorithm 3.1 in polynomial time [156]. Algorithm 3.1 first creates an empty set A_G. In every iteration, the algorithm computes the cost benefit ratio $c_a/(F(A_G \cup a) - F(A_G))$ for every $a \in \mathcal{A} \setminus A_G$. An element that corresponds to the lowest value of the cost benefit ratio is added to A_G. If $F(A_G) = F(\mathcal{A})$, then the algorithm terminates. Otherwise, the process is repeated until the constraint is satisfied. The performance guarantees of Algorithm 3.1 are provided in Lemma 3.3.

Lemma 3.3. (Theorem 1 [156]) *Let c^* be the optimal value of Problem 3.1, c_G be the objective value computed by Algorithm 3.1, and $H(n) = \sum_{i=1}^{n} i^{-1}$. Then*

$$c_G \leq H\big(\max_{a \in \mathcal{A}} F(a)\big) c^*. \tag{3.7}$$

We stress that the bound (3.7) is the worst-case theoretical bound. Hence, Algorithm 3.1 can perform better in practice.

The set cover problem

In this problem, we are given a set of elements $\mathcal{U}$ (called the universe) and a collection $\mathcal{S} = \{S_1, \ldots, S_n\}$ of subsets of $\mathcal{U}$. The goal is to pick the minimum number of elements from $\mathcal{S}$ whose union equals $\mathcal{U}$. We use this problem to establish NP-hardness of the security measure allocation problem (Chapter 5), and to derive a sensor allocation strategy (Chapter 7).

Although it is known that the set cover problem is generally an NP-hard problem [157], there exists several approaches to tackle this problem. Particularly, the set cover problem can be formulated as the following integer linear program [158]:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum_{S \in \mathcal{S}} x_S \\ \text{subject to} \quad & \sum_{S \in \mathcal{S}} \mathbb{1}_{[e \in S]} x_S \geq 1, \ \ \forall e \in \mathcal{U}, \\ & x_S \in \{0, 1\}, \ \ \forall S \in \mathcal{S}. \end{aligned}$$

Hence, one can try to utilize integer linear program solvers to tackle the set cover problem. In fact, modern-day solvers can solve large instances of this problem in practice [144]. Furthermore, this problem represents an instance of Problem 3.1. Hence, Algorithm 3.1 can be used to approximate the optimal value of the set cover efficiently. Other efficient ways to approximate the optimal value of the set cover problem are also known [158].

The minimum s-t cut problem

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a directed graph, and $c_{vw} \in \mathbb{R}_{\geq 0}$ be the weight associated to every edge $(v, w) \in \mathcal{E}$. Let the source s and the sink t be the elements of $\mathcal{V}$. An s-t cut is a partition of the node set $\mathcal{V}$ into V_s and $V_t = \mathcal{V} \setminus V_s$, such that $s \in V_s$ and $t \in V_t$. The cut capacity is defined by $C(V_s) = \sum_{\{(v,w) \in \mathcal{E} \,:\, v \in V_s, w \in V_t\}} c_{vw}$. The minimum s-t cut problem can then be written as follows:

$$\begin{aligned} \underset{V_s}{\text{minimize}} \quad & C(V_s) \\ \text{subject to} \quad & V_s \text{ and } V_t \text{ form an } s\text{-}t \text{ cut.} \end{aligned} \tag{3.8}$$

This problem can also be interpreted as the problem of finding a minimum cost edge separator of s and t. Once (3.8) is solved, this separator can be recovered from V_s as $E_c = \{(v, w) \in \mathcal{E} : v \in V_s, w \in V_t\}$. The cost of E_c is $C(V_s)$.

The minimum s-t cut problem can be solved in polynomial time using well-known algorithms [159]. We use this fact in Chapter 6 to compute the robust security index δ_r efficiently. It is also worth mentioning that the cut capacity $C(V_s)$ is a submodular function [152]. Hence, the minimum s-t cut problem is a constrained submodular minimization problem.

3.4 The KL–divergence

The KL–divergence $\mathcal{D}(p||q)$ gives a distance between probability density functions p and q. If p and q are continuous probability distributions over a sample space X, then the KL–divergence is defined as follows [160]:

$$\mathcal{D}(p||q) = \int_X \log \frac{p(x)}{q(x)} p(x) dx.$$

We have that $\mathcal{D}(p||q)$ is non-negative and equals to zero if and only if p equals q almost everywhere. If p and q are Gaussian distributions, then the KL–divergence can be expressed in a closed form, as the next lemma asserts [161, Section 9].

Lemma 3.4. *Let $p = \mathcal{N}(\mu_1, \Sigma_1)$ and $q = \mathcal{N}(\mu_2, \Sigma_2)$. If Σ_1 and Σ_2 are positive definite matrices, then*

$$\mathcal{D}\left(p||q\right) = \frac{1}{2}\left(\mathrm{Tr}(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^T \Sigma_2^{-1}(\mu_2 - \mu_1) + \log \frac{\det(\Sigma_2)}{\det(\Sigma_1)} - n\right).$$

The KL–divergence has been used in numerous fields, such as information theory [162], machine learning [163], and neuroscience [164]. In the area of control system security, the KL–divergence is used for modeling stealthiness constraints [60–62, 117, 118]. In Chapter 4 of this thesis, we use the KL–divergence for the same purpose.

3.5 Zero-sum games

Game theory studies interactions between multiple strategic players [165]. Each of the players tries to minimize or maximize his/her objective function by selecting one of the available strategies. However, the objective function of a player depends on the strategies of other players. Hence, besides his/her own strategies, a player needs to consider the strategies of other players as well. Such situations arise naturally in

control theory [166, 167]. For example, game theory has been used to design controllers and estimators [168,169], develop charging plans for electrical vehicles [170], and achieve cooperation in multi-agent systems [171]. Game theory has also been used for studying various security-related problems. Examples include the analysis of attacks [172–177], the design of defense strategies [146,178–180], the allocation of security investment [181,182], the design and tuning of anomaly detectors [183–186], the network interdiction [147, 148], and the sensor allocation [141–145].

In Chapter 7, we are interested in two-player zero-sum games. In this type of games, the player's objectives are opposite. That is, the loss of the first player is the gain of the second. A two-player zero-sum game Γ is defined by a tuple

$$\Gamma = \langle \{\text{P1}, \text{P2}\}, (\mathcal{A}_1, \mathcal{A}_2), f \rangle,$$

where P1 (resp. P2) denotes the first (resp. second) player, $\mathcal{A}_1$ (resp. $\mathcal{A}_2$) is a set of pure strategies available to P1 (resp. P2), and $f : \mathcal{A}_1 \times \mathcal{A}_2 \to \mathbb{R}$ is the payoff function that P1 (resp. P2) aims to maximize (resp. minimize). The following example aims to clarify the terminology.

Example 3.3. *Consider the matching pennies game where P1 and P2 have a penny whose sides are referred to as heads and tails. The players secretly select the sides of their pennies and then simultaneously reveal their selections. If the pennies match, then P1 wins P2's penny. Otherwise, P2 wins P1's penny. Hence, the sets of pure strategies $\mathcal{A}_1$ and $\mathcal{A}_2$ are given by: $\mathcal{A}_1 = \mathcal{A}_2 = \{Heads, Tails\}$. The payoff function can be defined by $f(a_1, a_2) = \mathbb{1}_{[a_1 = a_2]} - \mathbb{1}_{[a_1 \neq a_2]}$.*

Besides pure strategies, the players may also use mixed strategies. A mixed strategy of a player is a probability distribution over the set of his/her pure strategies. Particularly, the set of mixed strategies of the i^{th} player ($i \in \{0, 1\}$) is defined by

$$\Delta_i = \left\{ \sigma_i \in [0,1]^{|\mathcal{A}_i|} \;\middle|\; \textstyle\sum_{a_i \in \mathcal{A}_i} \sigma_i(a_i) = 1 \right\}.$$

Here, σ_i is a mixed strategy of the i^{th} player that assigns a probability $\sigma_i(a_i)$ for taking a pure strategy a_i. In other words, the i^{th} player selects a pure strategy to play according to a sampling from the probability distribution σ_i. Given a strategy profile $(\sigma_1, \sigma_2) \in \Delta_1 \times \Delta_2$, the expected payoff is defined by

$$F(\sigma_1, \sigma_2) = \sum_{a_1 \in \mathcal{A}_1} \sum_{a_2 \in \mathcal{A}_2} \sigma_1(a_1)\sigma_1(a_2) f(a_1, a_2).$$

The following example clarifies mixed strategies and the expected payoff.

Example 3.4. *Consider again the game matching pennies from Example 3.3. An example of a mixed strategy of P1 (resp. P2) is $\sigma_1 = [0.5\ 0.5]$ (resp. $\sigma_2 = [0.5\ 0.5]$). The expected payoff given these strategies is*

$$F(\sigma_1, \sigma_2) = 0.5 \cdot 0.5 \cdot 1 + 0.5 \cdot 0.5 \cdot (-1) + 0.5 \cdot 0.5 \cdot (-1) + 0.5 \cdot 0.5 \cdot 1 = 0.$$

We are interested in strategy profile(s) that are NE of Γ. A strategy profile $(\sigma_1^*, \sigma_2^*) \in \Delta_1 \times \Delta_2$ is a NE if

$$F(\sigma_1^*, \sigma_2) \geq F(\sigma_1^*, \sigma_2^*) \geq F(\sigma_1, \sigma_2^*)$$

holds for all $(\sigma_1, \sigma_2) \in \Delta_1 \times \Delta_2$. Put differently, if P2 plays according to σ_2^*, P1 cannot perform better than by playing according to σ_1^*. The same holds for σ_2^* and P2. We now provide an example of a NE.

Example 3.5. *In the game matching pennies from Example 3.3, a NE strategies are given by* $\sigma_1^* = [0.5\ 0.5]$ *and* $\sigma_2^* = [0.5\ 0.5]$*, and the expected payoff in a NE is equal to zero (see Example 3.4). Indeed, we have*

$$F(\sigma_1^*, \sigma_2) = 0.5 \cdot \sigma_{21} \cdot 1 + 0.5 \cdot \sigma_{22} \cdot (-1) + 0.5 \cdot \sigma_{21} \cdot (-1) + 0.5 \cdot \sigma_{22} \cdot 1 = 0.$$

Hence, P1 achieves the same payoff regardless of the strategy that P2 decides to play. The same holds for P2 and σ_2^**.*

We conclude this chapter by listing some properties of NE strategies:

(i) By playing σ_1^*, P1 is guaranteed to achieve the payoff of at least $F(\sigma_1^*, \sigma_2^*)$ regardless of P2's strategy. Similarly, by playing σ_2^*, P2 achieves the payoff not grater then $F(\sigma_1^*, \sigma_2^*)$ regardless of P1's strategy.

(ii) If $(\sigma_{11}^*, \sigma_{21}^*)$ and $(\sigma_{12}^*, \sigma_{22}^*)$ are two NE of a zero-sum game, then $(\sigma_{11}^*, \sigma_{22}^*)$ and $(\sigma_{12}^*, \sigma_{21}^*)$ are also NE [187]. In words, the game value $F(\sigma_1^*, \sigma_2^*)$ is the same in any NE. Therefore, it suffices for the players to find a single randomized strategy that lies in a NE.

(iii) In finite two-player zero-sum games and in some security games, NE strategies are also optimal for other solution concepts such as Strong Stackelberg, Min-Max, and Max-Min equilibrium [188].

(iv) If a number of pure strategies is finite, then a NE of a zero-sum game exists. Additionally, it can be obtained by solving a pair of linear programs [187]. Although linear programs often can be solved efficiently, in some zero-sum games these programs can be challenging to solve due to their size. Such a game is the topic of Chapter 7.

Chapter 4

Impact estimation

This chapter studies the impact estimation problem. By solving this problem, we test if an attacker can inflict significant damage to a control system while remaining stealthy. Hence, the objective function of the problem is an impact metric that is maximized, while the constraints include a stealthiness constraint. We consider two impact metrics: The probability that some of the critical states leave a safety region (I_P) and the expected value of the infinity norm of the critical states (I_E). For the stealthiness constraint, we adopt the KL–divergence between attacked and non-attacked residual sequences. Other constraints ensure that the system equations are satisfied, and impose different types of attack strategies.

The main results of the chapter are as follows. We characterize conditions under which the impact estimation problem becomes infeasible or its optimal value equals to the maximum impact. When these conditions are not satisfied, we prove that the optimal value of the metric I_P can be computed by solving a set of convex problems. We also derive efficient to compute lower and upper bounds for the metric I_E. We then show compatibility of our framework with a number of attack strategies proposed throughout the literature, and discuss how properties of these strategies can be used to more efficiently estimate the impact. Finally, we demonstrate on a control system of a chemical process how our framework can be used to compare security vulnerabilities, and illustrate some of the technical results with examples.

The chapter is organized as follows. Section 4.1 introduces the model setup. Section 4.2 presents the impact estimation problem. Section 4.3 contains the main technical results of the chapter. Section 4.4 introduces attack strategies compatible with our framework, and discusses how to more efficiently estimate the impact. Section 4.5 illustrates how our framework can be used to compare security vulnerabilities, and clarifies some of the technical results through examples. Section 4.6 concludes the chapter. The appendix contains some formulas and lengthy proofs.

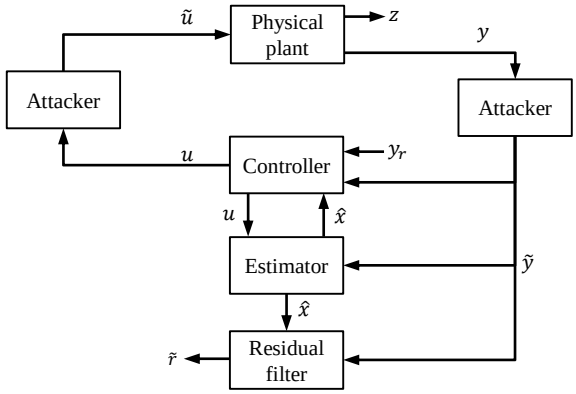

Figure 4.1: A schematic of an attacked control system. The controller computes the control actions u based on the references y_r, the state estimates $\hat{x}$, and the received measurements $\tilde{y}$. Due to attacks against sensors, the received measurements $\tilde{y}$ differ from the measurements y collected from the plant. Due to attacks against actuators, the corrupted control actions $\tilde{u}$ are applied to the plant instead of u. The critical states z are used to define the impact metrics, and the residuals $\tilde{r}$ are used to define the stealthiness constraint.

4.1 Model setup

This section presents the control system model. As shown in Figure 4.1, the system consists of the physical plant, the estimator, the controller, the residual filter, and the attacker. In the following, we introduce each block in more detail.

4.1.1 Physical plant

The physical plant is modeled by

$$\begin{aligned} x(k+1) &= Ax(k) + B\tilde{u}(k) + v_x(k), \\ y(k) &= Cx(k) + v_y(k), \\ z(k) &= C_z x(k), \end{aligned} \tag{4.1}$$

where $x(k) \in \mathbb{R}^{n_x}$ are the plant states, $y(k) \in \mathbb{R}^{n_y}$ are the sensor measurements, $\tilde{u}(k) \in \mathbb{R}^{n_u}$ are the control actions applied to the plant, $v_x(k) \in \mathbb{R}^{n_x}$ is the process noise, $v_y(k) \in \mathbb{R}^{n_y}$ is the measurement noise, and $z(k) \in \mathbb{R}^{n_z}$ are the critical states. The critical states may model the flow of energy through a power line that should be maintained within predefined bounds, or a temperature that should not exceed some safety limit. These states are later used to define the impact metrics.

We assume the following: (i) v_x and v_y are independent, zero-mean, white Gaussian processes with covariance matrices $\Sigma_{v_x} \succ 0$ and $\Sigma_{v_y} \succ 0$, respectively; (ii) the pair (C, A) is observable and the pair (B, A) is controllable; and (iii) C_z is a full row rank scaling matrix. The matrix C_z is chosen in such a way that having any of the critical states' magnitude larger than one indicates a dangerous system state. The following example clarifies how C_z is constructed.

Example 4.1. *Let $x = [x_1\ x_2]^T$ be the plant states. Let x_2 be the critical state that should be kept within the interval $[-2, 2]$. We then have $C_z = [0\ 1/2]$. Therefore, if $|x_2(k)| > 2$, then $|z(k)| > 1$.*

4.1.2 Estimator

The estimator is a steady state Kalman filter defined by

$$\hat{x}(k+1) = (A - KC)\hat{x}(k) + Bu(k) + K\tilde{y}(k), \tag{4.2}$$

where $\hat{x}(k) \in \mathbb{R}^{n_x}$ are the state estimates, $u(k) \in \mathbb{R}^{n_u}$ are the control actions computed by the controller, and $\tilde{y}(k) \in \mathbb{R}^{n_y}$ are the measurements received by the estimator. The steady state Kalman gain is given by

$$K = A\Sigma_e C^T (C\Sigma_e C^T + \Sigma_{v_y})^{-1},$$

where Σ_e is the error covariance matrix obtained by solving the Riccati equation

$$\Sigma_e = A\Sigma_e A^T + \Sigma_{v_x} - A\Sigma_e C^T (C\Sigma_e C^T + \Sigma_{v_y})^{-1} C\Sigma_e A^T.$$

The gain K exists under the introduced assumptions, and it is known that $A - KC$ is asymptotically stable [44].

4.1.3 Controller

The controller is defined by

$$u(k) = -L_1\hat{x}(k) - L_2\tilde{y}(k) + L_3 y_r(k), \tag{4.3}$$

where $y_r(k) \in \mathbb{R}^{n_{y_r}}$ are the references. We assume that the controller ensures asymptotic stability and satisfactory performances in the absence of attacks. Additionally, since the references are usually constants that are not updated often, we adopt the following standing assumption.

Assumption 4.1. *We assume that $y_r(k) = y_r$ holds for every time step $k \in \mathbb{Z}$. The system has reached a stationary regime before an attack starts.*

4.1.4 Residual filter

The residuals are defined by

$$\tilde{r}(k) = \Sigma_r^{-\frac{1}{2}}\big(\tilde{y}(k) - C\hat{x}(k)\big), \tag{4.4}$$

where $\Sigma_r = C\Sigma_e C^T + \Sigma_{v_y}$. The residuals are used in the next section to define the stealthiness constraint. In the absence of attacks, the sequence of residuals is a zero-mean white Gaussian process with the identity covariance matrix. We denote by r the non-attacked residuals to distinguish them from $\tilde{r}$.

4.1.5 Attacker

The attacked control actions $\tilde{u}$ and the attacked measurements $\tilde{y}$ are defined by

$$\begin{aligned} \tilde{u}(k) &= \Lambda_u u(k) + \Gamma_u a_u(k) + \Gamma_u a_{su}(k), \\ \tilde{y}(k) &= \Lambda_y y(k) + \Gamma_y a_y(k) + \Gamma_y a_{sy}(k), \end{aligned} \tag{4.5}$$

where $a_u(k) \in \mathbb{R}^{n_u}$ (resp. $a_{su}(k) \in \mathbb{R}^{n_u}$) are the deterministic (resp. stochastic) attacks against the actuators, $a_y(k) \in \mathbb{R}^{n_y}$ (resp. $a_{sy}(k) \in \mathbb{R}^{n_y}$) are the deterministic (resp. stochastic) attacks against the sensors, and the matrices Γ_u, Γ_y, Λ_u, and Λ_y depend on an attack strategy and the attacker's resources. Section 4.4 explains how these matrices are formed. We assume that an attack starts at $k = 0$.

4.1.6 Extended system model

From (4.1)–(4.5), the system dynamics under an attack become

$$\begin{aligned} x_e(k+1) &= \tilde{A}_e x_e(k) + \tilde{B}_e v(k) + \tilde{E}_e y_r + \tilde{G}_e a(k) + \tilde{G}_e a_s(k), \\ \tilde{r}(k) &= \tilde{C}_r x_e(k) + \tilde{D}_r v(k) + \tilde{F}_r y_r + \tilde{H}_r a(k) + \tilde{H}_r a_s(k), \\ z(k) &= \tilde{C}_z x_e(k) + \tilde{D}_z v(k) + \tilde{F}_z y_r + \tilde{H}_z a(k) + \tilde{H}_z a_s(k), \end{aligned} \tag{C1}$$

where $x_e(k) = [x(k)^T\, \hat{x}(k)^T]^T$, $v(k) = [v_x(k)^T\, v_y(k)^T]^T$, $a(k) = [a_u(k)^T\, a_y(k)^T]^T$, and $a_s(k) = [a_{su}(k)^T\, a_{sy}(k)^T]^T$. We denote by n_a the dimension of $a(k)$ and $a_s(k)$, n_v the dimension of $v(k)$, and Σ_v the covariance matrix of $v(k)$.

Finally, we introduce the equations for x_e and y in the absence of attacks

$$\begin{aligned} x_e(k+1) &= A_e x_e(k) + B_e v(k) + E_e y_r, \\ y(k) &= C_y x_e(k) + D_y v(k) + F_y y_r, \end{aligned} \tag{4.6}$$

which we later use in some derivations. We remark that the formulas for the matrices from Equations (C1) and (4.6) are provided in Appendix 4.A.

4.2 Problem formulation

This section defines the impact estimation problem. Prior to introducing the problem, we introduce the decision variables, the impact metrics, and the constraints.

4.2.1 Decision variables

Let $N \in \mathbb{N}$ be the length of a finite time horizon over which we want to estimate the impact. We define the decision variables by

$$d = \begin{bmatrix} a_{0:N} \\ y_r \end{bmatrix}. \tag{4.7}$$

Although the system trajectory is influenced by other signals as well, we show that the impact metrics and the constraints are only affected by the references y_r and the attack sequence $a_{0:N}$. Since we perform off-line impact analysis, the exact value of y_r at the beginning of the attack is unknown. The sequence $a_{0:N}$ is also unknown, since it depends on the attacker's choice. Hence, by optimizing over d, we identify the worst-case impact.

4.2.2 Impact metrics

In the related work on deterministic systems [26], the impact metric was defined by $\|z_{1:N}\|_\infty$. If $\|z_{1:N}\|_\infty > 1$, then the attacker can drive some of the critical states outside the safety region in N time steps. Yet, in our work, the states are influenced by the noise in addition to attacks. Hence, some of the critical states can leave the safety region with non-zero probability even in the absence of attacks.

To make the impact metric suitable for stochastic systems, we define a new metric

$$I_P(d) = \max_{i \in \mathcal{I}} \ \mathbb{P}\big(|z^{(i)}_{1:N}| > 1 \,;\, d\big),$$

where $\mathcal{I} = \{1, 2, \ldots, Nn_z\}$. The worst-case for the operator occurs when $I_P(d) \approx 1$. This implies that some of the critical states leave the safety region with high probability. The best case for the operator occurs when $I_P(d) \approx 0$. In this case, the critical states stay within the safety region with high probability. Another possible impact metric based on the ∞-norm is the expected value of $\|z_{1:N}\|_\infty$, that is,

$$I_E(d) = \mathbb{E}\{\, \|z_{1:N}\|_\infty \,;\, d \,\}.$$

Unfortunately, I_E does not have a closed form expression and is hard to evaluate in general. Thus, we are primarily focused on the metric I_P in this chapter.

4.2.3 Constraints

The problem constraints are denoted by (C1)–(C5). Constraint (C1) has already been introduced, and it imposes that the system equations have to be satisfied. Constraint (C2) is the reference constraint, which we define by

$$\|Q_{y_r} y_r\|_\infty \leq 1, \tag{C2}$$

where $Q_{y_r} \in \mathbb{R}^{n_{y_r} \times n_{y_r}}$ is a scaling matrix. Constraint (C3) is the stealthiness constraint defined by

$$\frac{1}{N+1} \mathcal{D}(\tilde{r}_{0:N} || r_{0:N}) \leq \epsilon, \tag{C3}$$

where $\mathcal{D}(\tilde{r}_{0:N} || r_{0:N})$ is the KL–divergence between the probability density functions of attacked $\tilde{r}_{0:N}$ and non-attacked $r_{0:N}$ residual sequences, and $\epsilon \in \mathbb{R}_{\geq 0}$ is the stealthiness level. As explained in Section 3.4, the KL–divergence measures similarity between probability density functions. Thus, if $\mathcal{D}(\tilde{r}_{0:N} || r_{0:N})$ is small, then the density functions of $\tilde{r}_{0:N}$ and $r_{0:N}$ are similar, and the attacker is assumed to stay stealthy. Finally, (C4) and (C5) are given by

$$F_a a_{0:N} = 0_{n_{F_a}}, \tag{C4}$$

$$a_{s0:N} = T_1 x_e(N_s) + T_2 y_r + T_3 v_{N_s:-1}. \tag{C5}$$

Here, $N_s \in \mathbb{Z}^-$, the matrices T_1, T_2, T_3, and F_a have appropriate dimensions, and n_{F_a} is the number of rows of the matrix F_a. These constraints enable us to impose different attack strategies (see Section 4.4).

4.2.4 Problem

Let $I \in \{I_P, I_E\}$. The impact estimation problem can then be formulated as follows:

Problem 4.1. *The impact estimation*

$$\begin{aligned} \underset{d}{\text{maximize}} \quad & I(d) \\ \text{subject to} \quad & \text{(C1)–(C5)}. \end{aligned}$$

Problem 4.1 is a non-convex constrained maximization problem. Efficient algorithms for solving these type of problems are generally unknown. Nevertheless, we derive an efficient way to compute the optimal value of the metric I_P. Additionally, we derive lower and upper bounds for the metric I_E. Before we move to the analysis, we outline some properties of this problem.

Remark 4.1. *The tuning parameters in Problem 4.1 are the length of the horizon N and the stealthiness level ϵ. Naturally, we first want to discover stealthy attacks that*

result in a high impact in a short amount of time. Thus, choosing small values of N and ϵ is a good starting point for the analysis. One can then start increasing N and ϵ to discover less dangerous attacks.

Remark 4.2. *One can also consider maximizing the impact in N_z steps and imposing the stealthiness in $N_r \neq N_z$ steps. The case $N_z < N_r$ captures attacks that maximize the impact in N_z steps and ensure stealthiness in additional $N_r - N_z$ steps. The case $N_z > N_r$ models ambush attacks [189], where the attacker stealthily prepares N_r steps, and then launches a not necessarily stealthy attack in the remaining time. Although we focus on the case where $N_r = N_z = N$, the analysis in the next section can be extended to cover the aforementioned cases.*

Remark 4.3. *Some of the advantages of using the KL-divergence to model the stealthiness constraint are as follows: (i) As shown in Section 4.3,* (C3) *is a convex and symmetric constraint in d; (ii) The analysis is made independent of the choice of the anomaly detector; (iii) Generating attacks that satisfy* (C3) *can be a reasonable choice by the attacker that does not know which anomaly detector is deployed; and (iv) Some other types of stealthiness constraints can be replaced by a KL–divergence based constraint [105].*

Remark 4.4. *Problem 4.1 can be infeasible due to* (C3)*. If that is the case, then we define the impact to be zero.*

4.3 Main results

This section shows that the optimal value of the metric I_P can be computed by solving a set of convex problems (Theorem 4.1), and derives lower and upper bounds for the metric I_E (Theorem 4.2). Prior to presenting Theorems 4.1 and 4.2, we introduce some auxiliary lemmas, present a problem crucial for deriving Theorems 4.1 and 4.2, and characterize when Problem 4.1 becomes infeasible or unbounded.

4.3.1 Preliminary analysis

We first establish the distribution of the extended state x_e prior to attacks.

Lemma 4.1. *Let $N_s \in \mathbb{Z}^-$. The extended state $x_e(N_s)$ is distributed according to $\mathcal{N}(T_0 y_r, \Sigma_0)$, where $T_0 = (I_{2n_x} - A_e)^{-1} E_e$, and the covariance matrix Σ_0 is the solution of the Lyapunov equation $\Sigma_0 = A_e \Sigma_0 A_e^T + B_e \Sigma_v B_e^T$.*

Proof. Since $N_s < 0$, attacks are not present, and the state x_e propagates according to (4.6). Assume that $y_r = 0_{n_{y_r}}$. Since A_e is asymptotically stable and the system is assumed to be in the stationary regime, then $x_e(N_s)$ is a zero mean Gaussian

vector, and its covariance matrix is of the desired form (see [44, Chapter 4]). If $y_r \neq 0_{n_{y_r}}$, then only the mean value of $x_e(N_s)$ changes. We then have

$$\mathbb{E}\{x_e(N_s)\} \overset{\text{(i)}}{=} \mathbb{E}\{x_e(N_s+1)\} \overset{\text{(ii)}}{=} A_e\mathbb{E}\{x_e(N_s)\} + E_e y_r, \tag{4.8}$$

where: (i) holds since the system (4.6) has reached the stationary regime (Assumption 4.1); and (ii) follows from (4.6), linearity of the expectation, and $\mathbb{E}\{v(N_s)\}=0_{n_v}$. From (4.8) and the fact that the inverse of $I_{2n_x} - A_e$ exists (A_e is assumed to be asymptotically stable), we have $\mathbb{E}\{x_e(N_s)\} = (I_{2n_x} - A_e)^{-1}E_e y_r = T_0 y_r$. ■

In the following, we use Lemma 4.1 to characterize probability density functions of the vectors $z_{1:N}$ and $\tilde{r}_{0:N}$.

Lemma 4.2. *Under Constraints* (C1) *and* (C5), $z_{1:N}$ *is distributed according to* $\mathcal{N}(T_Z d, \Sigma_Z)$ *and* $\tilde{r}_{0:N}$ *according to* $\mathcal{N}(T_R d, \Sigma_R)$. *The matrices* T_Z, T_R, Σ_Z, Σ_R *are independent of* d, *and* $\Sigma_Z \succ 0$ *is satisfied.*

Proof. We refer the reader to Appendix 4.B. ■

We can now use Lemma 4.2 to show that the stealthiness constraint (C3) is a convex and symmetric constraint in d. To show this claim, we also need $\Sigma_R \succ 0$ to hold. However, this condition is not always satisfied in the presence of attacks. In what follows, we assume $\Sigma_R \succ 0$, and later justify this assumption.

Assumption 4.2. *The covariance matrix* Σ_R *is a positive definite matrix.*

Lemma 4.3. *Under Assumption 4.2, Constraint* (C3) *becomes* $\|T_R d\|_2^2 \leq \epsilon'$, *where* $\epsilon' = (N+1)(2\epsilon + n_y) - \mathrm{Tr}(\Sigma_R) + \ln \det(\Sigma_R)$.

Proof. The attacked residual sequence $\tilde{r}_{0:N}$ is distributed according to $\mathcal{N}(T_R d, \Sigma_R)$, and the non-attacked residual sequence $r_{0:N}$ according to $\mathcal{N}(0_{(N+1)n_y}, I_{(N+1)n_y})$. From the latter and Lemma 3.4, we have

$$\mathcal{D}(\tilde{r}_{0:N}||r_{0:N}) = \frac{1}{2}\big(\mathrm{Tr}(\Sigma_R) + \|T_R d\|_2^2 - (N+1)n_y - \log\det(\Sigma_R)\big) = \frac{1}{2}\|T_R d\|_2^2 + c,$$

where $c = (\mathrm{Tr}(\Sigma_R) - (N+1)n_y - \log\det(\Sigma_R))/2$. Hence, (C3) can be rewritten as

$$\|T_R d\|_2^2 \leq 2((N+1)\epsilon - c) = (N+1)(2\epsilon + n_y) - \mathrm{Tr}(\Sigma_R) + \log\det(\Sigma_R) = \epsilon',$$

as claimed in the statement of the lemma. ■

Remark 4.5. *As we illustrate in Section 4.5, some attack strategies may result in* ϵ' *being less than zero. Constraint* (C3) *is then impossible to satisfy, and Problem 4.1 is infeasible. Particularly,* ϵ' *approaches* $-\infty$ *when an eigenvalue of* Σ_R *approaches zero. This justifies focusing on the cases where* Σ_R *is positive definite. Thus, if an attack is such that* Σ_R *is not positive definite, then we adopt the impact to be zero.*

Next, consider the problem

$$\begin{array}{lll} \mathcal{P}_i: & \underset{d}{\text{maximize}} & \left|\mathbb{E}\{z_{1:N}^{(i)}; d\}\right| \\ & \text{subject to} & \text{(C1)–(C5)}, \end{array}$$

where $i \in \mathcal{I}$. This problem is crucial for computing the optimal value of the metric I_P and deriving bounds for the metric I_E. In what follows, we use Lemmas 4.2 and 4.3 to show that $\mathcal{P}_i$ is equivalent to a convex problem with symmetric constraints. Thus, $\mathcal{P}_i$ can be solved efficiently using well known algorithms.

Lemma 4.4. *Under Assumption 4.2, $\mathcal{P}_i$ is equivalent to the problem*

$$\begin{array}{ll} \underset{d}{\text{maximize}} & T_Z(i,:)d \\ \text{subject to} & \|Qd\|_\infty \leq 1, \;\; \|T_R d\|_2^2 \leq \epsilon', \;\; Fd = 0_{n_{F_a}}, \end{array} \tag{4.9}$$

where $Q = [0_{n_{y_r} \times (N+1)n_a} \; Q_{y_r}]$, and $F = [F_a \;\; 0_{n_{F_a} \times n_{y_r}}]$.

Proof. We first rewrite $\mathcal{P}_i$ in a more convenient way. From Lemma 4.2, (C1) and (C5) impose that $z_{1:N} \sim \mathcal{N}(T_Z d, \Sigma_Z)$ and $\tilde{r}_{0:N} \sim \mathcal{N}(T_R d, \Sigma_R)$. Thus, the objective function of $\mathcal{P}_i$ is equal to $|T_Z(i,:)d|$. Since

$$\|Q_{y_r} y_r\|_\infty = \|[0_{n_{y_r} \times (N+1)n_a} \; Q_{y_r}][a_{0:N}^T \; y_r^T]^T\|_\infty = \|Qd\|_\infty \leq 1,$$

(C2) can be rewritten as the first constraint in the problem (4.9). From Lemma 4.3, (C3) reduces to the second constraint in (4.9). Finally, (C4) can be rewritten as

$$F_a a_{0:N} = [F_a \;\; 0_{n_{F_a} \times n_{y_r}}][a_{0:N}^T \; y_r^T]^T = Fd,$$

which is the third constraint in (4.9). Therefore, $\mathcal{P}_i$ is equivalent to

$$\begin{array}{lll} \underset{d}{\text{maximize}} & |T_Z(i,:)d| & \\ \text{subject to} & z_{1:N} \sim \mathcal{N}(T_Z d, \Sigma_Z), \;\; r_{0:N} \sim \mathcal{N}(T_R d, \Sigma_R), & \text{(C1')} \\ & \|Qd\|_\infty \leq 1, \quad \|T_R d\|_2^2 \leq \epsilon', \quad Fd = 0_{n_{F_a}}. & \text{(C2'–C4')} \end{array} \tag{4.10}$$

Since (C1') does not impose any restriction on d, and neither $z_{1:N}$ nor $r_{0:N}$ appears in the objective function and the remaining constraints, we can eliminate (C1'). The remaining constraints are symmetric in d, so we can substitute $|T_Z(i,:)d|$ with $T_Z(i,:)d$ without affecting the optimal value. After these simplifications, the problem (4.10) reduces to the problem (4.9). Thus, $\mathcal{P}_i$ is equivalent to (4.9). ■

Next, we investigate when $\mathcal{P}_i$ is infeasible or unbounded, and then explain the importance of this result.

Proposition 4.1. *The following statements hold:*

(i) $\mathcal{P}_i$ *is infeasible for any* $i \in \mathcal{I}$ *if and only if* $\epsilon' < 0$ *holds.*

(ii) Let $\epsilon' \geq 0$*. The problem* $\mathcal{P}_i$ *is unbounded for at least one* $i \in \mathcal{I}$ *if and only if* $\text{null}([Q^T\ T_R^T\ F^T]^T) \not\subseteq \text{null}(\ T_Z)$ *holds.*

Proof. Statement (i): ($\Rightarrow$) If $\epsilon' \geq 0$, then we can see from (4.9) that $d = 0$ is a feasible point of $\mathcal{P}_i$ for any $i \in \mathcal{I}$. Hence, $\epsilon' < 0$ has to hold.

($\Leftarrow$) If $\epsilon' < 0$, then $\|T_R d\|_2^2 \leq \epsilon'$ cannot be satisfied for any d, so the claim holds.

Statement (ii): ($\Rightarrow$) The proof is by contradiction. If $\text{null}([Q^T\ T_R^T\ F^T]^T) \subseteq \text{null}(\ T_Z)$ and $\epsilon' \geq 0$, then we have $[Q^T\ T_R^T\ F^T]^T d \neq 0$ for every d for which $T_Z d \neq 0$. Hence, $T_Z(i,:)d$ cannot be made arbitrarily large for any $i \in \mathcal{I}$, since that would violate at least one of the constraints.

($\Leftarrow$) If $\text{null}([Q^T\ T_R^T\ F^T]^T) \not\subseteq \text{null}(\ T_Z)$ and $\epsilon' \geq 0$, then there exists d that satisfies $T_Z d \neq 0$ and $[Q^T\ T_R^T\ F^T]^T d = 0$. By increasing the magnitude of this d while keeping its direction fixed, we can make $T_Z(i,:)d$ unbounded for at least one i and simultaneously keep the constraints satisfied. ■

Proposition 4.1 has two important consequences. Firstly, if $\mathcal{P}_i$ is unbounded, then the system is seriously vulnerable. Namely, when the conditions $\epsilon' \geq 0$ and $\text{null}([Q^T\ T_R^T\ F^T]^T) \not\subseteq \text{null}(\ T_Z)$ are satisfied, the attacker can make the deterministic part of at least one critical state arbitrarily large while remaining stealthy. The influence of the stochastic component then becomes negligible, and the optimal value of Problem 4.1 for the metric I_P (resp. I_E) goes to 1 (resp. $+\infty$). In words, the attack results in the maximum impact.

Secondly, $\mathcal{P}_i$ is infeasible if and only if Problem 4.1 is infeasible, since these problems have the same constraints. Hence, Problem 4.1 is infeasible if and only if the attacker cannot satisfy a predefined stealthiness level.

Since Proposition 4.1 tells us the impact when $\mathcal{P}_i$ is infeasible or unbounded, in the remainder we focus on the case where $\mathcal{P}_i$ is feasible and bounded. Thus, we introduce the following assumption.

Assumption 4.3. *We assume that* $\epsilon' \geq 0$ *and* $\text{null}([Q^T\ T_R^T\ F^T]^T) \subseteq \text{null}(\ T_Z)$.

4.3.2 Computing the optimal value of the metric I_P

We now introduce Algorithm 4.1 that solves Problem 4.1 when $I = I_P$. For every $i \in \mathcal{I}$, Algorithm 4.1 computes a solution d_i^* of $\mathcal{P}_i$. Based on d_i^*, the algorithm computes the probability

$$\hat{P}_i^* = \mathbb{P}\big(\,|z_{1:N}^{(i)}| > 1\,;\, d_i^*\,\big).$$

Algorithm 4.1 Computing the optimal value of the metric I_P

1: **Input:** T_R, T_Z, Σ_Z, Σ_R, Q, F, ϵ
2: **Output:** $\hat{I}_P^*$
3: **for** every $i \in \mathcal{I}$ **do**
4: Compute a solution d_i^* of $\mathcal{P}_i$
5: Compute $\hat{P}_i^* = \mathbb{P}(|z_{1:N}^{(i)}| > 1\,;\, d_i^*)$
6: **end for**
7: $\hat{I}_P^* = \max_{i\in\mathcal{I}} \hat{P}_i^*$

Since $z_{1:N}^{(i)}$ is a Gaussian random variable (Lemma 4.2), $\hat{P}_i^*$ can be computed efficiently and accurately given d_i^*. Finally, the algorithm returns $\hat{I}_P^* = \max_{i\in\mathcal{I}} \hat{P}_i^*$ as the attack impact. We now establish that $\hat{I}_P^*$ is the optimal value of Problem 4.1.

Theorem 4.1. *Let Assumption 4.3 be satisfied and $I = I_P$. If I_P^* is the optimal value of Problem 4.1 and $\hat{I}_P^*$ is the value returned by Algorithm 4.1, then $I_P^* = \hat{I}_P^*$.*

Proof. We refer the reader to Appendix 4.C. ■

Theorem 4.1 represents an interesting extension of the work [26] that considered the impact metric $\|z_{1:N}\|_\infty$. Particularly, Theorem 4.1 shows that the optimal value of the metric I_P can be computed by solving $\mathcal{P}_i$ $n_z N$ times, which is equivalent to solving the convex problem (4.9) $n_z N$ times. This is the same favorable property that the impact metric $\|z_{1:N}\|_\infty$ has. Furthermore, note that $\mathcal{P}_i$ can be solved in parallel for every $i \in \mathcal{I}$. Thus, the time needed to estimate the impact can be considerably reduced using parallel computing. Moreover, we discuss in Section 4.4 how to further reduce this time by using properties of attack strategies.

4.3.3 Computing lower and upper bounds for the metric I_E

We now use $\mathcal{P}_i$ to bound the metric I_E. Let us define

$$\hat{I}_E^* = \max_{i\in\mathcal{I}} \mu_i^*, \tag{4.11}$$

where μ_i^* is the optimal value of $\mathcal{P}_i$ corresponding to i. Theorem 4.2 provides lower and upper bounds for the metric I_E based on $\hat{I}_E^*$.

Theorem 4.2. *Let Assumption 4.3 be satisfied and $I = I_E$. If I_E^* is the optimal value of Problem 4.1 and $\hat{I}_E^*$ is defined as in* (4.11)*, then*

$$\hat{I}_E^* \;\leq\; I_E^* \;\leq \hat{I}_E^* + \sum_{i=1}^{Nn_z} \sqrt{\frac{2\Sigma_Z(i,i)}{\pi}}. \tag{4.12}$$

Proof. We refer the reader to Appendix 4.D. ■

We highlight two consequences of Theorem 4.2. Firstly, we can see that the bounds are tight in at least two cases: (i) $\hat{I}_E^*$ is considerably larger than the sum from (4.12); and (ii) $\Sigma_Z(i,i)$ has a small value for every i (noise is negligible). The bounds can be useful even if the tightness cannot be established. If the lower (resp. upper) bound is large (resp. small), then I_E^* is for sure large (resp. small). Secondly, note that Σ_Z is independent of d (Lemma 4.2). Hence, we only need $\hat{I}_E^*$ to compute the bounds. Therefore, the bounds can be computed by solving $\mathcal{P}_i$ Nn_z times, same as the optimal value of Problem 4.1.

4.4 Attack strategies compatible with our framework

This section introduces attack strategies whose impact can be computed using our framework, and discusses how properties of these strategies can be used to more efficiently estimate the impact.

4.4.1 Attacks strategies

Prior to presenting the strategies, we introduce some notation. We denote by $\mathcal{Y} = \{y_1, \ldots, y_{n_y}\}$ the set of sensors and by $\mathcal{U} = \{u_1, \ldots, u_{n_u}\}$ the set of actuators. We also assume that by exploiting a group of security vulnerabilities, the attacker gains control over a subset of sensors $Y_a \subseteq \mathcal{Y}$ and a subset of actuators $U_a \subseteq \mathcal{U}$.

DoS, rerouting, and sign alternation strategies

We first consider three attack strategies that can be modeled by

$$\tilde{y}(k) = \Lambda_y y(k), \qquad \tilde{u}(k) = \Lambda_u u(k). \tag{4.13}$$

The first such a strategy is the DoS attack strategy [2, 29], where the attacker prevents the sensor measurements Y_a and the control actions U_a from reaching their destination. For example, the attacker can physically damage the corresponding sensors and actuators, or jam a network over which these signals are transmitted [2]. Here, Λ_y and Λ_u are diagonal matrices defined by

$$\Lambda_y(i,i) = \begin{cases} 0, & y_i \in Y_a, \\ 1, & y_i \notin Y_a, \end{cases} \qquad \Lambda_u(i,i) = \begin{cases} 0, & u_i \in U_a, \\ 1, & u_i \notin U_a. \end{cases} \tag{4.14}$$

In the sign alternation attack strategy [31, 101], the attacker flips the sign of the measurements Y_a and the control actions U_a. Such an attack can turn negative

feedback into positive, and potentially destabilize the system. Moreover, in certain configurations with the Kalman filter, sign alternation attacks may be strictly stealthy [101]. In this case, Λ_u and Λ_y are diagonal matrices given by

$$\Lambda_y(i,i) = \begin{cases} -1, & y_i \in Y_a, \\ 1, & y_i \notin Y_a, \end{cases} \qquad \Lambda_u(i,i) = \begin{cases} -1, & u_i \in U_a, \\ 1, & u_i \notin U_a. \end{cases}$$

Finally, the rerouting attack strategy consists of the attacker permuting the values of the measurements Y_a and the control actions U_a [30, 104]. The attack can be performed by modifying the respective senders' identifiers, or by physically re-wiring the cables [30]. In this attack, Λ_y and Λ_u are permutation matrices that satisfy $\Lambda_y(i,i) = 1$ for $y_i \notin Y_a$ and $\Lambda_u(i,i) = 1$ for $u_i \notin U_a$.

The following proposition establishes compatibility of the above mentioned attack strategies with our framework.

Proposition 4.2. *The impact estimation problems on the DoS, rerouting, and sign alternation attack strategies can be formulated as Problem 4.1.*

Proof. It suffices to show that these attack strategies can be imposed through (C4) and (C5). From (4.13), it follows that $a_y \equiv 0$, $a_u \equiv 0$, and $a_s \equiv 0$. These constraints on a_y and a_u can be modeled by (C4), by setting $F_a = I_{(N+1)n_a}$. The constraint on a_s can be modeled by (C5), by setting T_1, T_2, T_3 to zero. ■

Remark 4.6. *If every measurement Y_a and every control action U_a can be blocked separately, then the total number of choices for the matrices Λ_y and Λ_u equals to $2^{|Y_a|+|U_a|}$. Thus, computing the worst-case attack impact for all possible DoS attacks can be expensive if $|Y_a| + |U_a|$ is large. A similar observation holds for sign alternation attacks. A way to reduce this number is to use the nature of a vulnerability that enables the attacker to corrupt U_a or Y_a. For instance, if the attacker jams the network over which multiple control or measurement signals are transmitted, then the access to all of these signals is denied.*

Remark 4.7. *Computing the impact for all possible rerouting attacks can also be computationally expensive. Namely, it can be shown that the total number of possible choices for Λ_y and Λ_u is equal to $|Y_a|!|U_a|!$. A way to reduce the number of combinations is by selecting combinations that are more likely to happen. For example, to avoid easy detection, the attacker would do well to exchange two measurements or control actions that are of similar nature.*

Optimal FDI, bias injection, and combined FDI and DoS strategies

In the optimal FDI attack strategy [26, 98], the attacker uses the model knowledge to construct an optimal attack sequence $a_{0:N}$. The signals $\tilde{y}$ and $\tilde{u}$ are given by

$$\tilde{y}(k) = y(k) + \Gamma_y a_y(k), \qquad \tilde{u}(k) = u(k) + \Gamma_u a_u(k), \tag{4.15}$$

where Γ_y and Γ_u are diagonal matrices defined by

$$\Gamma_y(i,i) = \begin{cases} 1, & y_i \in Y_a, \\ 0, & y_i \notin Y_a, \end{cases} \qquad \Gamma_u(i,i) = \begin{cases} 1, & u_i \in U_a, \\ 0, & u_i \notin U_a. \end{cases} \tag{4.16}$$

In the bias injection attack strategy, the attacker injects a constant bias to the measurements Y_a and the control actions U_a [27,112]. Hence, $\tilde{y}$ and $\tilde{u}$ are given by

$$\tilde{y}(k) = y(k) + \Gamma_y a_y(0), \qquad \tilde{u}(k) = u(k) + \Gamma_u a_u(0), \tag{4.17}$$

where Γ_y and Γ_u are defined in (4.16). One can notice that the only difference between (4.15) and (4.17) is that a_u and a_y are now constant.

Finally, one can imagine a situation where the attacker can inject corrupted data to measurements Y_I and control actions U_I, but can only deny access to measurements Y_D and control actions U_D. In this case, the attacker can use the combined FDI and DoS attack strategy [115,116], in which $\tilde{y}$ and $\tilde{u}$ are given by

$$\tilde{y}(k) = \Lambda_y y(k) + \Gamma_y a_y(k), \qquad \tilde{u}(k) = \Lambda_u u(k) + \Gamma_u a_u(k). \tag{4.18}$$

Here, Λ_y and Λ_u are defined based on Y_D and U_D as in (4.14), and Γ_y and Γ_u are defined based on Y_I and U_I as in (4.16).

The above-mentioned injection strategies are also compatible with our framework.

Proposition 4.3. *The impact estimation problems on the optimal FDI, bias, and combined FDI and DoS attack strategies can be formulated as Problem 4.1.*

Proof. Same as in the previous proof, we show that the attack strategies can be imposed through (C4) and (C5). Consider first the optimal FDI attack strategy. In this strategy, a_u and a_y are free to choose. This can be modeled by (C4) by setting F_a to zero. Next, note that $a_s \equiv 0$ can be modeled by (C5) by setting T_1, T_2, T_3 to zero. Hence, the optimal FDI attack strategy is compatible with our framework.

The proof for the combined FDI and DoS attack strategy is the same as for the optimal FDI attack strategy, since a_u and a_y are free to choose, and $a_s \equiv 0$.

The proof for the bias injection attack strategy is similar. The only difference are the constraints $a_y(k) = a_y(0)$, $a_u(k) = a_u(0)$, for every $k \in \{1, \ldots, N\}$. These constraints are linear equality constraints that can be modeled by (C4). ■

Replay strategy

The replay attack strategy is inspired by the Stuxnet attack [6]. A replay attack on sensors can be modeled by

$$\tilde{y}(k) = \Lambda_y y(k) + \Gamma_y a_{sy}(k), \tag{4.19}$$

where Λ_y is defined in (4.14), Γ_y is defined in (4.16), and

$$a_{sy}(k) = y(k - N - 1). \tag{4.20}$$

Put differently, the attacker replaces the attacked measurements with the measurements of the normal operation previously recorded at the time steps $-N-1, \ldots, -1$. The purpose of attacking the sensors Y_a is to cover an attack against the actuators U_a. We model the attack against the actuators U_a as a DoS attack

$$\tilde{u}(k) = \Lambda_u u(k), \tag{4.21}$$

where Λ_u is defined in (4.14). We remark that the attack against the actuators can be modeled in other ways as well [190].

The replay attack strategy defined in this way is also compatible with our framework, as shown in the following proposition.

Proposition 4.4. *The impact estimation problem on the replay attack strategy can be formulated as Problem 4.1.*

Proof. We again prove the claim by showing that the replay attack strategy can be imposed through (C4) and (C5). From (4.19) and (4.21), we have $a \equiv 0$, which can be modeled by (C4) by setting $F_a = I_{(N+1)n_a}$.

It remains to show that a_s can be expressed as in (C5). Let $N_s = -N - 1$. From (4.20), we have $a_{sy0:N} = y_{N_s:-1}$. From the latter, (4.6), and (3.3), it follows that

$$a_{sy0:N} = y_{N_s:-1} = T_1' x_e(N_s) + T_2' v_{N_s:-1} + T_3'(1_{N+1} \otimes I_{n_{y_r}}) y_r,$$

where $T_1' = \mathcal{O}_N(A_e, C_y)$, $T_2' = \mathcal{T}_N(A_e, B_e, C_y, D_y)$, and $T_3' = \mathcal{T}_N(A_e, E_e, C_y, F_y)$. Therefore, the constraint (4.20) on a_{sy} has the same form as (C5). Additionally, we have $a_{su} \equiv 0$, which can also be imposed through (C5). Hence, it follows that the replay attack strategy is compatible with our framework. ■

4.4.2 Estimating impact more efficiently

Recall that we need to solve $\mathcal{P}_i$ multiple times to compute the optimal value of the metric I_P and the bounds for the metric I_E. We now discuss how to use properties of the attack strategies to simplify $\mathcal{P}_i$ or reduce the number of times we solve this problem, and in that way, estimate the impact more efficiently. To simplify some formulas from this section, we denote the dimension of $a_{0:N}$ by n_A.

DoS, rerouting, sign alternation, and replay strategies

In the case of the DoS, rerouting, sign alternation, and replay attack strategies, we have $a \equiv 0$. This implies that we can simplify $\mathcal{P}_i$ by eliminating the decision variables $a_{0:N}$ and the constraint $Fd = 0_{n_{F_a}}$. This is stated in Proposition 4.5.

Proposition 4.5. *Let Assumption 4.2 be satisfied. In the case of the DoS, rerouting, sign alternation, and replay attack strategies, $\mathcal{P}_i$ is equivalent to*

$$\begin{aligned} \underset{y_r}{\text{maximize}} \quad & T'_Z(i,:)y_r \\ \text{subject to} \quad & \|Q_{y_r}y_r\|_\infty \leq 1, \;\; \|T'_R y_r\|_2^2 \leq \epsilon', \end{aligned} \tag{4.22}$$

where $T'_Z = T_Z(:, n_A+1 : n_A+n_{y_r})$ and $T'_R = T_R(:, n_A+1 : n_A+n_{y_r})$.

Proof. Under Assumption 4.2, $\mathcal{P}_i$ is equivalent to the problem (4.9). The objective function of (4.9) can be rewritten as

$$T_Z(i,:)d \overset{a\equiv 0,(4.7)}{=} T_Z(i, n_A+1 : n_A+n_{y_r})y_r = T'_Z(i,:)y_r.$$

The first constraint of (4.9) is equivalent to $\|Q_{y_r}y_r\|_\infty \leq 1$ by the definition of Q. The second constraint of (4.9) can be rewritten as

$$\|T_R d\|_2^2 \overset{a\equiv 0,(4.7)}{=} \|T_R(:, n_A+1 : n_A+n_{y_r})y_r\|_2^2 = \|T'_R y_r\|_2^2 \leq \epsilon'.$$

Finally, the last constraint of (4.9) becomes $F_a a_{0:N} = 0_{n_{F_a}}$ by the definition of F. Since we showed that the objective function and the remaining constraints are not dependent on $a_{0:N}$, the last constraint of (4.9) and the decision variables $a_{0:N}$ can be eliminated without affecting the optimal value. ■

Bias injection strategy

In the bias injection attack strategy, we have

$$a(k) = a(0), \qquad \forall k \in \{1, \dots, N\}. \tag{4.23}$$

This constraint can be used to simplify $\mathcal{P}_i$ by eliminating the decision variables $a_{1:N}$ and the constraint $Fd = 0_{n_{F_a}}$, as established in the following proposition.

Proposition 4.6. *Let Assumption 4.2 be satisfied. In the case of the bias injection attack strategy, $\mathcal{P}_i$ is equivalent to*

$$\begin{aligned} \underset{a(0),y_r}{\text{maximize}} \quad & T'_Z(i,:)a(0) + T''_Z(i,:)y_r \\ \text{subject to} \quad & \|Q_{y_r}y_r\|_\infty \leq 1, \;\; \|T'_R a(0)\|_2^2 \leq \epsilon', \end{aligned}$$

where $T'_Z = T_Z(:, 1 : n_A)(1_{N+1} \otimes I_{n_a})$, $T''_Z = T_Z(:, n_A+1 : n_A+n_{y_r})$, and $T'_R = T_R(:, 1 : n_A)(1_{N+1} \otimes I_{n_a})$.

Proof. We refer the reader to Appendix 4.E. ■

Optimal FDI strategy

In the case of the optimal FDI attack strategy, the number of times we solve the problem $\mathcal{P}_i$ can be reduced from Nn_z to n_z. To explain why is this the case, we need the following proposition.

Proposition 4.7. *Let Assumption 4.3 hold, i be arbitrarily selected from $\{1,\dots,n_z\}$, j be arbitrarily selected from $\{1,\dots,N\}$, and assume that the attacker uses the optimal FDI attack strategy. Consider the problems*

$$\mathcal{P}_{ij}^{(1)}: \quad \underset{d}{\text{maximize}} \ |\mathbb{E}\{z_i(j)\,;\,d\}| \quad \text{subject to (C1)–(C5)},$$

$$\mathcal{P}_{ij}^{(2)}: \quad \underset{d}{\text{maximize}} \ \mathbb{P}(|z_i(j)|>1\,;\,d) \quad \text{subject to (C1)–(C5)}.$$

If $\mu^*(i,j)$ is the optimal value of $\mathcal{P}_{ij}^{(1)}$ and $P^*(i,j)$ is the optimal value of $\mathcal{P}_{ij}^{(2)}$, then $\mu^*(i,j)\le\mu^*(i,N)$ and $P^*(i,j)\le P^*(i,N)$ hold.

Proof. We refer the reader to Appendix 4.F.

We now explain how Proposition 4.7 can be used to reduce the number of executions of $\mathcal{P}_i$. Firstly, note that $\mathcal{P}_{ij}^{(1)}$ and $\mathcal{P}_i$ are the same problems written out in different ways. By adopting the new notation, we emphasize that we focus on the i^{th} critical state at the j^{th} time step. Secondly, the proof of Theorem 4.1 shows that computing the optimal value of I_P requires us to:

(i) solve $\mathcal{P}_{ij}^{(1)}$ for every $(i,j)\in\{1,\dots,n_z\}\times\{1,\dots,N\}$ to compute $P^*(i,j)$;

(ii) compute the optimal impact I_P^* as the maximum of $P^*(i,j)$ over all i and j.

From Proposition 4.7, we have

$$\max_{i\in\{1,\dots,n_z\}}\max_{j\in\{1,\dots,N\}}P^*(i,j)\le\max_{i\in\{1,\dots,n_z\}}P^*(i,N).$$

Therefore, it suffices to fix $j=N$ and solve $\mathcal{P}_{ij}^{(1)}$ for every i to compute the optimal impact I_P^*. In words, $\mathcal{P}_{ij}^{(1)}$ is solved n_z times instead of Nn_z times. Similarly, since $\mu^*(i,j)\le\mu^*(i,N)$, the bounds for I_E can be computed by solving $\mathcal{P}_{ij}^{(1)}$ n_z times.

Remark 4.8. *Attack strategies other than optimal FDI do not generally satisfy this useful monotonicity property. Counterexamples are provided in the next section.*

4.5 Illustrative examples

This section illustrates how the modeling framework we propose can be used for comparison of security vulnerabilities, and discusses some of the technical results through examples. We begin by introducing the control system model.

4.5.1 Model: Chemical process

We consider a chemical plant from [41] shown in Figure 4.2 (a). The states are the volume in Tank 3 (x_1), the volume in Tank 2 (x_2), and the temperature in Tank 2 (x_3). The control actions 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). We assume that the control objective is to maintain a constant temperature in Tank 2. The objective is achieved by injecting hot water from Tank 1, and cold water from Tank 3. The matrices describing the physical plant are given by

$$A = \begin{bmatrix} 0.9550 & 0 & 0 \\ 0.0442 & 0.9675 & 0 \\ -0.0444 & 0.0007 & 0.8958 \end{bmatrix}, B = \begin{bmatrix} 8.7961 & -2.2479 & 0 & 0 \\ 0.2016 & 2.2109 & 4.9184 & 0 \\ -0.2051 & -2.2194 & 1.8958 & 21.1173 \end{bmatrix},$$

$C = I_3$, $\Sigma_{v_x} = 0.05\, I_3$, and $\Sigma_{v_y} = 0.01\, I_3$. The controller matrices are given by

$$L_1 = \begin{bmatrix} 0.1034 & 0.0182 & -0.0012 \\ -0.0196 & 0.0714 & -0.0049 \\ 0.0135 & 0.1632 & 0.0022 \\ -0.0044 & -0.0069 & 0.0417 \end{bmatrix}, \quad L_3 = \begin{bmatrix} 0.1109 & 0.1126 & 0.0000 \\ -0.0102 & 0.4405 & 0.0000 \\ 0 & 0 & 0 \\ 0.0000 & 0.0474 & 0.0474 \end{bmatrix},$$

and $L_2 = 0_{n_u \times n_y}$. We adopt $Q_{y_r} = 0.4\, I_3$, and use the steady state Kalman filter as an estimator.

We assume a cyber-infrastructure shown in Figure 4.2 (b). The communication link between Slave PLC 1 and Master PLC is unprotected (vulnerability V_1). The same holds for the link between Slave PLC 2 and Master PLC (vulnerability V_2). If the attacker exploits V_1, then he/she gains control over the components y_2, y_3, u_3, u_4. As for V_2, the attacker gains control over y_1, u_1, u_2.

4.5.2 Example 1: Comparison of security vulnerabilities

We first illustrate how our framework can be used to compare security vulnerabilities. We set $N = 10$, $\epsilon = 0.3$, $C_z = [0_{1\times 2}\ \ 1/3]$, and $I = I_P$. We then compute the impacts of the DoS, rerouting, replay, and bias injection attacks in the cases when either V_1 or V_2 are exploited. Since the attacker can conduct DoS and rerouting attacks in multiple ways, we compute the worst-case impact over all possible implementations. For replay attacks, we assume that the attacker launches a DoS attack against all the actuators under control.

The results of the analysis are illustrated in Figure 4.3. Note that the impact of different attacks may result in different conclusions concerning the importance of vulnerabilities. On the one hand, based on the impact of the DoS attacks, it follows that V_2 is more important to be prevented than V_1. On the other hand, based on the impact of the replay, optimal FDI, and bias attacks, V_1 is more critical. The

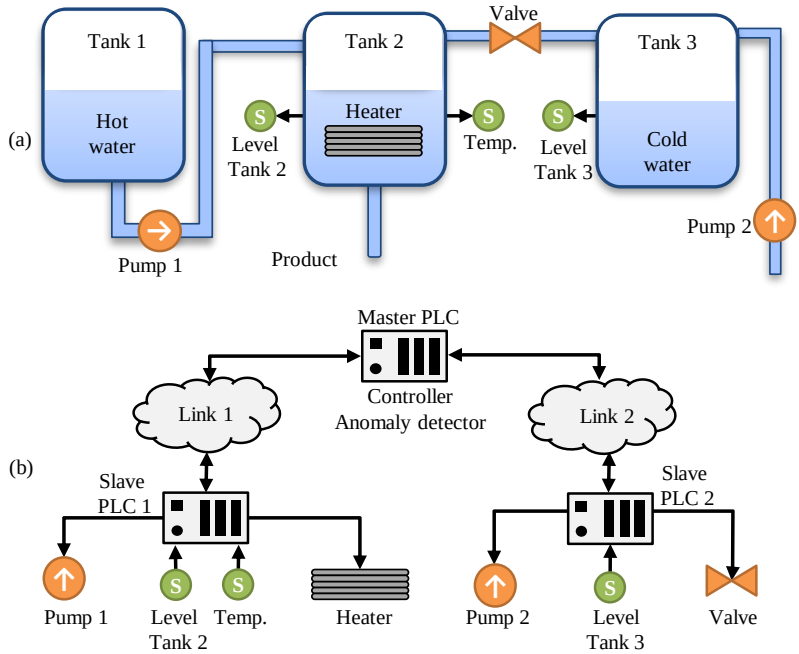

Figure 4.2: (a) The physical part of the system. There are four actuators (two pumps, one heater, and one valve), and three sensors (two level sensors and one temperature sensor). (b) The cyber part of the system. The communication link between Slave PLC 1 and Master PLC is unprotected (vulnerability V_1). The same holds for the link between Slave PLC 2 and Master PLC (vulnerability V_2).

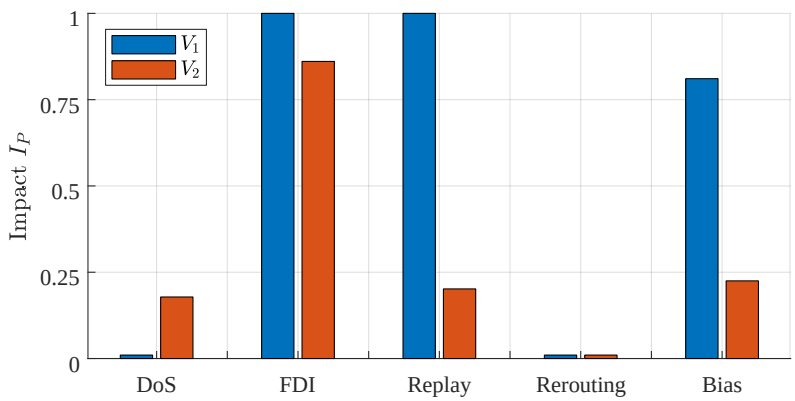

Figure 4.3: The impact of different attacks when vulnerability V_1 is exploited (blue bars) and when vulnerability V_2 is exploited (red bars).

impact of rerouting attacks was not informative, since it was equal to zero in both cases. Taking all the attacks into account, we can give a higher priority to V_1, since the impact in the majority of the cases is larger when V_1 is exploited.

We also observe that sometimes less complex attack strategies can be just as dangerous as the optimal FDI attack strategy. For example, in the case of V_1, the replay attack resulted in the same impact as the optimal FDI attack.

4.5.3 Example 2: Maximum impact condition

Observe from Figure 4.3 that if the attacker exploits V_1 and uses the optimal FDI attack strategy, then he/she can conduct an attack that results in the maximum impact. In fact, this is an example of a scenario where the attacker can make the deterministic part of the critical state x_3 arbitrarily large, since the maximum impact condition from Proposition 4.1 is satisfied. Namely, by manipulating the compromised actuators, the attacker affects the volume x_2 and the temperature x_3 of Tank 2. The changes in the system behavior caused by the attack cannot be seen neither from the sensors y_2 and y_3 (the effect is removed by the attacker), nor from the sensor y_1 (x_2 and x_3 do not affect y_1).

4.5.4 Example 3: Feasibility of the impact estimation problem

We now illustrate how the stealthiness level ϵ influences the feasibility of the impact estimation problem. We vary ϵ in the range $[0, 2]$, and adopt the other modeling parameters to be the same as in Example 1. The following attacks and attack resources are considered: (i) a DoS attack against u_1 and u_2; (ii) a replay attack against y_1 combined with a DoS attack against u_1 and u_2; (iii) a rerouting attack against y_1 and y_2; and (iv) an optimal FDI attack against u_1 and y_1. A plot of the impact of these attacks with respect to ϵ is shown on Figure 4.4.

As we can see, the impact is non-decreasing with respect to ϵ in all four cases. This is expected, since by increasing ϵ, the stealthiness constraint (C3) becomes easier to satisfy. However, an important point is that some of the attacks require ϵ to be larger than a certain threshold to have an impact larger than zero. For example, in the case of the replay (resp. DoS) attack, ϵ needs to be larger than 0.1 (resp. 0.5). In the case of rerouting attack, the impact is equal to zero for all the values of ϵ we consider. The explanation is as follows.

From Lemma 4.3, the stealthiness constraint (C3) can be written as $\|T_R d\|_2^2 \leq \epsilon'$, where $\epsilon' = (N+1)(2\epsilon + n_y) - \mathrm{tr}(\Sigma_R) + \log\det(\Sigma_R)$. The replay, DoS, and rerouting attack strategies are multiplicative in nature ($\Lambda_u \neq I_{n_u}$, $\Lambda_y \neq I_{n_y}$), and may change the residual covariance matrix Σ_R. If ϵ is not large enough, then these attacks may result in ϵ' being less than zero. Problem 4.1 is then infeasible (Proposition 4.1), and the impact is zero by the convention. In contrast, the optimal FDI attack

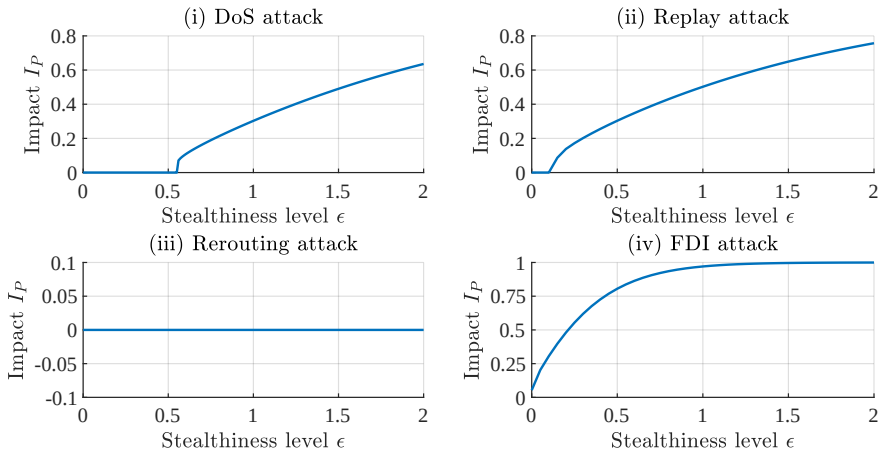

Figure 4.4: The impact of attacks specified in Example 3 with respect to ϵ.

strategy is additive ($\Lambda_u = I_{n_u}$, $\Lambda_y = I_{n_y}$), and does not affect Σ_R. We then have $\Sigma_R = I_{(N+1)n_y}$ and $\epsilon' = 2(N+1)\epsilon \geq 0$, so Problem 4.1 is always feasible in the case of this strategy.

4.5.5 Example 4: Monotonicity property from Proposition 4.7

This example shows that the attack strategies other than the optimal FDI generally do not satisfy the monotonicity property from Proposition 4.7. However, we also illustrate that these strategies can posses this property in some cases.

The following attacks and attacker's resources are considered: (i) a DoS attack against u_2; (ii) a DoS attack against all the actuators; (iii) a replay attack against y_1 combined with a DoS attack against u_1 and u_2; (iv) a replay attack against y_1 combined with a DoS attack against u_1 and u_2; (v) a bias injection attack against all the actuators; and (vi) a bias injection attack against all the sensors. In the cases (i), (iii), and (v), we use the same system model and the same values of N, ϵ, and C_z as in Example 1. In the cases (ii), (iv), and (vi), we deviate from the model as follows: (ii) $\epsilon = 4$; (iv) $N = 40$; (vi) $N = 20$, $\epsilon = 0.2$, $C_z = [0 \;\; 1/3 \;\; 0]$,

$$L_1 = -\begin{bmatrix} 0.0864 & 0.1151 & -0.0012 \\ -0.0862 & 0.0498 & -0.0049 \\ -0.1388 & 0.1694 & 0.0022 \\ 0.0021 & -0.0088 & 0.0417 \end{bmatrix}, L_3 = \begin{bmatrix} 0.0093 & 0.2128 & -0.0000 \\ -0.4079 & 0.4324 & 0.0000 \\ 0 & 0 & 0 \\ -0.0428 & 0.0475 & 0.0473 \end{bmatrix}.$$

Plots of $P^*(1, j)$ with respect to j for the above-mentioned attacks are shown in Figure 4.5. As we can see, the attacks we consider generally do not possess the

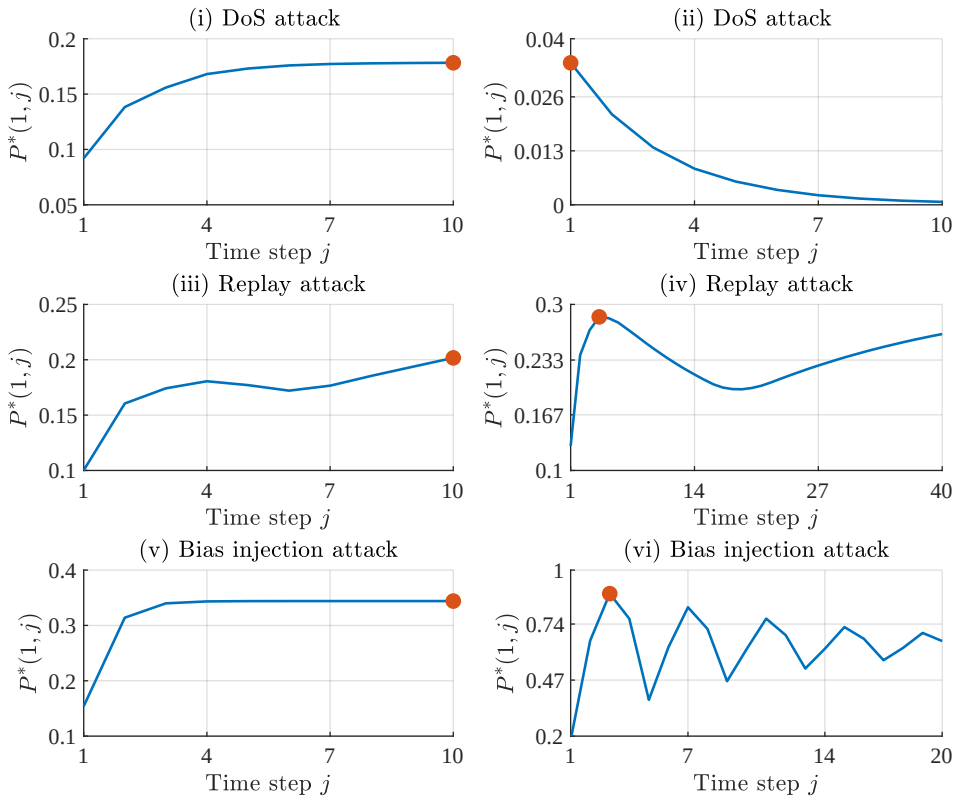

Figure 4.5: The value of $P^*(1,j)$ with respect to j for the attacks specified in Example 4. The maximum of the curve is indicated with a red circle.

property from Proposition 4.7. In (ii), $P^*(1,j)$ reaches the maximum value at the beginning of the horizon. In (iv) and (vi), $P^*(1,j)$ reaches the maximum value between the beginning and the end of the horizon. However, one can observe that the attacks specified in (i), (iii), and (v) do possess the property from Proposition 4.7. That is, $P^*(1,j)$ reaches the maximum value at the end of the horizon. Thus, it would be interesting to derive under which conditions this useful property holds for the attack strategies other than optimal FDI.

4.6 Summary

This chapter considered the impact estimation problem. Two impact metrics suitable for stochastic systems were proposed and studied. We derived conditions under which the impact estimation problem is infeasible or its optimal value equals to the

maximum impact, proved that the optimal value of the first metric can be computed by solving a set of convex problems, and derived lower and upper bounds for the second metric. Additionally, we showed that our impact estimation framework is compatible with a range of attack strategies, discussed how to use properties of these strategies to estimate the impact more efficiently, and demonstrated how the framework can be used to compare security vulnerabilities. We now move to the next chapter, where we utilize the framework in allocation of security measures.

Appendix to Chapter 4

4.A The matrices from Equations (C1) and (4.6)

$$\tilde{A}_e = \begin{bmatrix} A - B\Lambda_u L_2 \Lambda_y C & -B\Lambda_u L_1 \\ (K - BL_2)\Lambda_y C & A - KC - BL_1 \end{bmatrix} \quad \tilde{B}_e = \begin{bmatrix} I_{n_x} & -B\Lambda_u L_2 \Lambda_y \\ 0_{n_x \times n_x} & (K - BL_2)\Lambda_y \end{bmatrix}$$

$$\tilde{E}_e = \begin{bmatrix} B\Lambda_u L_3 \\ BL_3 \end{bmatrix} \quad \tilde{G}_e = \begin{bmatrix} B\Gamma_u & -B\Lambda_u L_2 \Gamma_y \\ 0_{n_x \times n_{a_u}} & (K - BL_2)\Gamma_y \end{bmatrix}$$

$$\tilde{C}_r = \Sigma_r^{-\frac{1}{2}} \begin{bmatrix} \Lambda_y C & -C \end{bmatrix} \quad \tilde{D}_r = \Sigma_r^{-\frac{1}{2}} \begin{bmatrix} 0_{n_y \times n_x} & \Lambda_y \end{bmatrix}$$

$$\tilde{H}_r = \Sigma_r^{-\frac{1}{2}} \begin{bmatrix} 0_{n_y \times n_u} & \Gamma_y \end{bmatrix} \quad \tilde{F}_r = 0_{n_y \times n_{y_r}}$$

$$\tilde{C}_z = \begin{bmatrix} C_z & 0_{n_z \times n_x} \end{bmatrix} \quad \tilde{D}_z = 0_{n_z \times n_v}$$

$$\tilde{F}_z = 0_{n_z \times n_{y_r}} \quad \tilde{H}_z = 0_{n_z \times n_a}$$

$$A_e = \begin{bmatrix} A - BL_2 C & -BL_1 \\ (K - BL_2)C & A - KC - BL_1 \end{bmatrix} \quad B_e = \begin{bmatrix} I_{n_x} & -BL_2 \\ 0_{n_x \times n_x} & K - BL_2 \end{bmatrix}$$

$$E_e = \begin{bmatrix} BL_3 \\ BL_3 \end{bmatrix} \quad C_y = \begin{bmatrix} C & 0_{n_y \times n_x} \end{bmatrix}$$

$$D_y = \begin{bmatrix} 0_{n_y \times n_x} & I_{n_y} \end{bmatrix} \quad F_y = 0_{n_y \times n_{y_r}}$$

4.B Proof of Lemma 4.2

We first prove that $z_{1:N}$ is distributed according to $\mathcal{N}(T_Z d, \Sigma_Z)$. Consider the non-attacked system (4.6). From (3.2), we have

$$x_e(0) = P_{1a} x_e(N_s) + P_{2a} v_{N_s:-1} + P_{3a} y_{rN_s:-1}, \tag{4.24}$$

where $P_{1a} = A_e^{|N_s|}$, $P_{2a} = \mathcal{C}_{|N_s|}(A_e, B_e)$, and $P_{3a} = \mathcal{C}_{|N_s|}(A_e, E_e)$. Consider now the attacked system (C1). From (3.3), it follows that

$$z_{0:N} = P_{1b} x_e(0) + P_{2b} v_{0:N} + P_{3b} y_{r0:N} + P_{4b} a_{0:N} + P_{4b} a_{s0:N}, \tag{4.25}$$

where $P_{1b} = \mathcal{O}_N(\tilde{A}_e, \tilde{C}_z)$, $P_{2b} = \mathcal{T}_N(\tilde{A}_e, \tilde{B}_e, \tilde{C}_z, \tilde{D}_z)$, $P_{3b} = \mathcal{T}_N(\tilde{A}_e, \tilde{E}_e, \tilde{C}_z, \tilde{F}_z)$, and $P_{4b} = \mathcal{T}_N(\tilde{A}_e, \tilde{G}_e, \tilde{C}_z, \tilde{H}_z)$. By combining (4.24) and (4.25), we obtain

$$z_{0:N} = P_{1c}x_e(N_s) + P_{2c}v_{N_s:N} + P_{3c}y_r + P_{4b}a_{0:N} + P_{4b}a_{s0:N}, \tag{4.26}$$

where $P_{1c} = P_{1b}P_{1a}$, $P_{2c} = [P_{1b}P_{2a}\ P_{2b}]$, and $P_{3c} = [P_{1b}P_{3a}\ \ P_{3b}](1_{|N_s|+N+1} \otimes I_{n_{y_r}})$. Let $P_l = [0_{Nn_z \times n_z}\ \ I_{Nn_z}]$. From (4.26) and (C5), it follows that

$$z_{1:N} = P_1 x_e(N_s) + P_2 v_{N_s:N} + P_3 y_r + P_4 a_{0:N}, \tag{4.27}$$

where $P_1 = P_l(P_{1c} + P_{4b}T_1)$, $P_2 = P_l(P_{2c} + [P_{4b}T_2\ 0_{(N+1)n_z \times (N+1)n_v}])$, $P_3 = P_l(P_{3c} + P_{4b}T_3)$, and $P_4 = P_l P_{4b}$.

Note that $P_1 x_e(N_s)$ and $P_2 v_{N_s:N}$ are independent Gaussian vectors. Additionally, observe that $P_3 y_r$ and $P_4 a_{0:N}$ are deterministic vectors. Since the sum of independent Gaussian vectors and deterministic vectors is a Gaussian vector, we conclude from (4.27) that $z_{1:N}$ is a Gaussian vector. From linearity of the expected value operator, the fact that $x_e(N_s)$ has the mean value $T_0 y_r$ (Lemma 4.1), and the fact that the noise is zero mean, we obtain

$$\mathbb{E}\{z_{1:N}\} = P_1 T_0 y_r + P_3 y_r + P_4 a_{0:N} \overset{(4.7)}{=} T_Z d,$$

where $T_Z = [P_4\ P_1 T_0 + P_3]$.

Next, since v is a white Gaussian sequence, the covariance matrix of $v_{N_s:N}$ is given by $\Sigma_V = I_{|N_s|+N+1} \otimes \Sigma_v$. Additionally, $P_1 x_e(N_s)$ and $P_2 v_{N_s:N}$ are independent Gaussian vectors. Thus, the covariance matrix of $z_{1:N}$ equals to the sum of the covariance matrices of $P_1 x_e(N_s)$ and $P_2 v_{N_s:N}$, that is,

$$\Sigma_Z = P_1 \Sigma_0 P_1^T + P_2 \Sigma_V P_2^T.$$

Finally, notice that T_Z and Σ_Z are independent of d.

The proof that $\tilde{r}_{0:N}$ is distributed according to $\mathcal{N}(T_R d, \Sigma_R)$ is similar, so we briefly summarize it. From (3.2), (3.3), (C1), and (4.6), $\tilde{r}_{0:N}$ can be written as

$$\tilde{r}_{0:N} = M_{1a}x_e(N_s) + M_{2a}v_{N_s:N} + M_{3a}y_r + M_4 a_{0:N} + M_4 a_{s0:N}, \tag{4.28}$$

where $M_{1a} = \mathcal{O}_N(\tilde{A}_e, \tilde{C}_r)A_e^{|N_s|}$, $M_{2a} = [\mathcal{O}_N(\tilde{A}_e, \tilde{C}_r)\mathcal{C}_{|N_s|}(A_e, B_e)\ \ \mathcal{T}_N(\tilde{A}_e, \tilde{B}_e, \tilde{C}_r, \tilde{D}_r)]$,

$$M_{3a} = [\mathcal{O}_N(\tilde{A}_e, \tilde{C}_r)\mathcal{C}_{|N_s|}(A_e, E_e)\ \ \mathcal{T}_N(\tilde{A}_e, \tilde{E}_e, \tilde{C}_r, \tilde{F}_r)](1_{|N_s|+N+1} \otimes I_{n_{y_r}}),$$

and $M_4 = \mathcal{T}_N(\tilde{A}_e, \tilde{G}_e, \tilde{C}_r, \tilde{H}_r)$. From (4.28) and (C5), it follows that

$$\tilde{r}_{0:N} = M_1 x_e(N_s) + M_2 v_{N_s:N} + M_3 y_r + M_4 a_{0:N},$$

where $M_1 = M_{1a} + M_4 T_1$, $M_2 = M_{2a} + [M_4 T_2\ 0_{(N+1)n_y \times (N+1)n_v}]$, and $M_3 = M_{3a} + M_4 T_3$. Using the same arguments as in the case of $z_{1:N}$, it can be shown

that the mean value of $\tilde{r}_{0:N}$ is $\mathbb{E}\{\tilde{r}_{0:N}\} = T_R d$, where $T_R = [M_4\ M_1 T_0 + M_3]$. The covariance matrix of $\tilde{r}_{0:N}$ is given by $\Sigma_R = M_1 \Sigma_0 M_1^T + M_2 \Sigma_V M_2^T$. We can see that T_R and Σ_R are independent of d.

Finally, we prove that $\Sigma_Z \succ 0$. Equation (4.27) can be rewritten as

$$z_{1:N} \;=\; P_1 x_e(N_s) \;+\; P_2' v_{N_s:-1} \;+\; P_2'' v_{x0:N} \;+\; P_2''' v_{y0:N} \;+\; P_3 y_r \;+\; P_4 a_{0:N},$$

where the exact formulas for the matrices P_2', P_2'', and P_2''' are omitted for the sake of brevity. Since $P_1 x_e(N_s)$, $P_2' v_{N_s:-1}$, $P_2'' v_{x0:N}$, $P_2''' v_{y0:N}$ are independent Gaussian vectors, Σ_Z is the sum of the covariance matrices of these vectors. Thus, it suffices to prove that one of these vectors has a positive definite covariance matrix.

From (4.1), P_2'' is of the form

$$P_2'' = \begin{bmatrix} C_z & 0_{n_z \times n_x} & \dots & 0_{n_z \times n_x} & 0_{n_z \times n_x} \\ \times & C_z & \dots & 0_{n_z \times n_x} & 0_{n_z \times n_x} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \times & \times & \dots & C_z & 0_{n_z \times n_x} \end{bmatrix}.$$

Since C_z has a full row rank, we have $\mathrm{null}((P_2'')^T) = \emptyset$. Since v_x is a white Gaussian process, the covariance matrix of $v_{x0:N}$ is given by $\Sigma_{V_x} = I_{N+1} \otimes \Sigma_{v_x}$. Additionally, from $\Sigma_{v_x} \succ 0$, we have $\Sigma_{V_x} \succ 0$. It then follows that $P_2'' \Sigma_{V_x} (P_2'')^T \succ 0$, and we conclude that $\Sigma_Z \succ 0$ holds.

4.C Proof of Theorem 4.1

Since the constraints of Problem 4.1 are independent of i, the optimal value I_P^* can be obtained in the following two steps:

(i) We compute the optimal value P_i^* of the optimization problem

$$\underset{d}{\text{maximize}}\ \mathbb{P}(|z_{0:N}^{(i)}| > 1\,;\, d) \quad \text{subject to (C1)–(C5)} \tag{4.29}$$

for every $i \in \mathcal{I}$.

(ii) We compute I_P^* as $I_P^* = \max_{i \in \mathcal{I}} P_i^*$.

We now prove that Algorithm 4.1 performs these two steps.

Step 1. Note that Algorithm 4.1 computes a solution d_i^* of $\mathcal{P}_i$ for every $i \in \mathcal{I}$. Under Assumption 4.2, d_i^* always exists. Based on d_i^*, Algorithm 4.1 computes $\hat{P}_i^* = \mathbb{P}(|z_{1:N}^{(i)}| > 1\,;\, d_i^*)$ (Lines 3–6). In the following, we prove that d_i^* is also a solution of the problem (4.29) for every $i \in \mathcal{I}$. This implies that $\hat{P}_i^* = P_i^*$, and shows that Algorithm 4.1 performs Step 1.

Let $i \in \mathcal{I}$ be arbitrarily selected. Since $\Sigma_Z \succ 0$, then $\mathcal{N}(T_Z d, \Sigma_Z)$ is a non-degenerate Gaussian distribution. Hence, $z^{(i)}_{1:N}$ is a Gaussian random variable with the mean value $\mu = T_Z(i,:)d$ and the variance $\sigma^2 = \Sigma_Z(i,i)$. Since Σ_Z is not affected by d, it follows that d influences the probability $\mathbb{P}(|z^{(i)}_{1:N}| > 1\,;\, d^*_i)$ only through μ. We now analyze how $\mathbb{P}(|z^{(i)}_{1:N}| > 1\,;\, d^*_i)$ changes with respect to μ.

Let $c = (\sqrt{2}\sigma)^{-1}$, $\tilde{z} \sim \mathcal{N}(\mu, \sigma^2)$, and $f(\mu) = \mathbb{P}(|\tilde{z}| > 1\,;\, \mu)$. Observe that $|\tilde{z}|$ is distributed according to the folded normal distribution [191]. Therefore, we have

$$f(\mu) = 1 - \frac{1}{2}\mathrm{erf}(c - c\mu) - \frac{1}{2}\mathrm{erf}(c + c\mu), \tag{4.30}$$

where $\mathrm{erf}(x) = \pi^{-1/2}\int_{-x}^{x} e^{-t^2} dt$ is the error function (for instance, see [191]). From (4.30), it follows that $f(-\mu) = f(\mu)$, so $f(\mu)$ is symmetric in μ (Property 1).

Using the formula $d\,\mathrm{erf}(z)/dz = 2\pi^{-1/2}e^{-z^2}$, we obtain

$$\frac{df(\mu)}{d\mu} = \frac{c}{\sqrt{\pi}} e^{-c^2(1-\mu)^2} - \frac{c}{\sqrt{\pi}} e^{-c^2(1+\mu)^2}.$$

Note that $e^{-c^2(1+\mu)^2} < e^{-c^2(1-\mu)^2}$ for $\mu > 0$, and $e^{-c^2(1+\mu)^2} > e^{-c^2(1-\mu)^2}$ for $\mu < 0$. Hence, it follows that f is monotonically increasing with respect to μ on the interval $(0, +\infty)$, and decreasing with respect to μ on the interval $(-\infty, 0)$. Due to this fact and Property 1, we have that $f(\mu)$ is increasing with respect to $|\mu|$ (Property 2).

Next, recall that $\mathcal{P}_i$ is on maximizing $|\mu|$ under the constraints (C1)–(C5). From Property 2, it follows that d^*_i that maximizes $|\mu|$ under (C1)–(C5), also maximizes $\mathbb{P}(|z^{(i)}_{1:N}| > 1\,;\, d)$ under (C1)–(C5). Hence, d^*_i is also a solution of (4.29).

Step 2. From Step 1, $\hat{P}^*_i = P^*_i$ holds for every $i \in \mathcal{I}$. From the latter, it directly follows that $\max_{i\in\mathcal{I}} P^*_i = \max_{i\in\mathcal{I}} \hat{P}^*_i$. Therefore, Algorithm 4.1 performs Step 2 as well, and we conclude that $I^*_P = \hat{I}^*_P$ holds.

4.D Proof of Theorem 4.2

Before we establish the bounds, we outline the connection between $\hat{I}^*_E$ and the infinity norm. Since $\mathbb{E}\{z_{1:N}\} = T_Z d$ (Lemma 4.2), then $\hat{I}^*_E$ is the optimal value of the following optimization problem

$$\underset{i\in\mathcal{I}}{\text{maximize}} \;\; \underset{d}{\text{maximize}} \; |T_Z(i,:)d| \quad \text{subject to (C1)–(C5).} \tag{4.31}$$

Since $\|T_Z d\|_\infty = \max_{i\in\mathcal{I}} |T_Z(i,:)d|$ and (C1)–(C5) are independent of i, the problem (4.31) can be rewritten as

$$\underset{d}{\text{maximize}} \; \|T_Z d\|_\infty \quad \text{subject to (C1)–(C5).} \tag{4.32}$$

Note that both (4.31) and (4.32) are feasible, since $\epsilon' \geq 0$. Hence, if d' is a solution of (4.32), then $\hat{I}_E^* = \|T_Z d'\|_\infty$. We are now ready to establish the bounds.

Lower bound. Let $Z' \sim \mathcal{N}(T_Z d', \Sigma_Z)$, and note that Z' is with the finite mean value (integrable) under Assumption 4.3. We then have

$$\hat{I}_E^* = \|\mathbb{E}\{Z'\,;\,d'\}\|_\infty \overset{(*)}{\leq} \mathbb{E}\{\|Z'\|_\infty\,;\,d'\} = I_E(d') \overset{(**)}{\leq} I_E^*.$$

Here, (*) follows from the convexity of the infinity norm and Jensen's inequality [192]. Additionally, (**) follows from the fact that d' is a feasible point of Problem 4.1, so $I_E(d')$ has to be lower than the optimal value I_E^* of Problem 4.1.

Upper bound. Let d^* be a solution of Problem 4.1 when $I = I_E$, and let $Z^* \sim \mathcal{N}(T_Z d^*, \Sigma_Z)$. Note that Z^* can be written as $Z^* = T_Z d^* + Z$, where $Z \sim \mathcal{N}(0_{Nn_z}, \Sigma_Z)$. Since $\Sigma_Z \succ 0$ (Lemma 4.2), then $\mathcal{N}(0, \Sigma_Z)$ is a non-degenerate Gaussian distribution. Hence, $Z_i \sim \mathcal{N}(0, \Sigma_Z(i,i))$, and $|Z_i|$ is a random variable distributed according to the folded normal distribution [191]. We then have

$$\begin{aligned} I_E^* = \mathbb{E}\{\|Z^*\|_\infty\,;\,d^*\} &\overset{\text{(i)}}{\leq} \mathbb{E}\{\|Z\|_\infty\,;\,d^*\} + \|T_Z d^*\|_\infty \overset{\text{(ii)}}{\leq} \mathbb{E}\{\|Z\|_\infty\,;\,d^*\} + \hat{I}_E^* \\ &\overset{\text{(iii)}}{\leq} \sum_{i=1}^{Nn_z} \mathbb{E}\{|Z_i|\,;\,d^*\} + \hat{I}_E^* \overset{\text{(iv)}}{=} \sum_{i=1}^{Nn_z} \sqrt{\frac{2\Sigma_Z(i,i)}{\pi}} + \hat{I}_E^*, \end{aligned}$$

where:

(i) follows from the triangle inequality and linearity of the expectation;

(ii) follows from $\|T_Z d^*\|_\infty \leq \|T_Z d'\|_\infty = \hat{I}_E^*$, since d^* is a feasible point and d' is a solution of (4.32);

(iii) follows from $\|Z\|_\infty \leq \sum_{i=1}^{Nn_z} |Z_i|$ and linearity of the expectation;

(iv) follows from $\mathbb{E}\{|Z_i|\,;\,d^*\} = \sqrt{2\Sigma_Z(i,i)/\pi}$ (for instance, see [191]).

4.E Proof of Proposition 4.6

Under Assumption 4.2, $\mathcal{P}_i$ is equivalent to the problem (4.9). The objective function of (4.9) can be rewritten as

$$T_Z(i,:)d \overset{(4.23)}{=} T_Z(i, 1:n_A)(1_{N+1} \otimes I_{n_a})a(0) + T_Z(i, n_A+1 : n_A+n_{y_r})y_r,$$

which equals to $T'_Z(i,:)a(0) + T''_Z(i,:)y_r$.

The first constraint of (4.9) can be rewritten as $\|Q_{y_r} y_r\|_\infty \leq 1$ by the definition of Q. The second constraint of (4.9) reduces to $\|T_R' a(0)\|_2^2 \leq \epsilon'$, since

$$\begin{aligned}\|T_R d\|_2^2 &\overset{(4.23)}{=} \|T_R(:, 1:n_A)(1_{N+1} \otimes I_{n_a})a(0) + T_R(:, n_A+1 : n_A+n_{y_r})y_r\|_2^2 \\ &\overset{(*)}{=} \|T_R' a(0)\|_2^2,\end{aligned}$$

where $(*)$ follows from the fact that the references y_r do not affect the residuals $\tilde{r}$. To show this, let us define the estimation errors by $e(k) = x(k) - \hat{x}(k)$. From the formulas in Appendix 4.A, $\Lambda_y = I_{n_y}$, and $\Lambda_u = I_{n_u}$, we have

$$e(k+1) = \tilde{A}_e' e(k) + \tilde{B}_e' v(k) + \tilde{E}_e' y_r + \tilde{G}_e' a(k),$$

where $\tilde{A}_e' = A - KC$, $\tilde{B}_e' = \begin{bmatrix} I_{n_x} & -K \end{bmatrix}$, $\tilde{E}_e' = 0_{n_x \times n_{y_r}}$, and $\tilde{G}_e' = \begin{bmatrix} B\Gamma_u & -K\Gamma_y \end{bmatrix}$. Since $\tilde{E}_e' = 0_{n_x \times n_{y_r}}$, the estimation errors $e(k)$ are not affected by the references y_r. Additionally, the residuals $\tilde{r}(k)$ can be rewritten as

$$\begin{aligned}\tilde{r}(k) &\overset{(4.4)}{=} \Sigma_r^{-\frac{1}{2}}\big(\tilde{y}(k) - C\hat{x}(k)\big) \overset{(4.5),(4.17)}{=} \Sigma_r^{-\frac{1}{2}}\big(y(k) + \Gamma_y a_y(k) - C\hat{x}(k)\big), \\ &\overset{(4.1)}{=} \Sigma_r^{-\frac{1}{2}}\big(Ce(k) + v_y(k) + \Gamma_y a_y(k)\big).\end{aligned} \tag{4.33}$$

Hence, we conclude that the references y_r do not affect the residuals $\tilde{r}$.

Finally, the last constraint of (4.9) reduces to (4.23). Since we showed that the objective function and the first two constraints are not affected by $a_{1:N}$, this constraint and the decision variables $a_{1:N}$ can be eliminated.

4.F Proof of Proposition 4.7

The proof relies on Lemmas 4.5 and 4.6 that we introduce next.

Lemma 4.5. *Let Assumption 4.2 be satisfied and $T_R' = T_R(:, 1:n_A)$. In the case of the optimal FDI attack strategy, $\mathcal{P}_{ij}^{(1)}$ can be rewritten as follows:*

$$\begin{aligned}\underset{y_r, a_{0:N}}{\text{maximize}} \quad & |\tilde{C}_z(i,:)(\mathcal{C}_j(\tilde{A}_e, \tilde{G}_e)a_{0:j-1} + T_0 y_r)| \\ \text{subject to} \quad & \|Q_{y_r} y_r\|_\infty \leq 1, \;\; \|T_R' a_{0:N}\|_2^2 \leq \epsilon'.\end{aligned} \tag{4.34}$$

Proof. Under Assumption 4.2, $\mathcal{P}_{ij}^{(1)}$ can be rewritten as

$$\begin{aligned}\underset{d}{\text{maximize}} \quad & |\mathbb{E}\{z_i(j)\,;\, d\}| \\ \text{subject to} \quad & \|Qd\|_\infty \leq 1, \; \|T_R d\|_2^2 \leq \epsilon', \; Fd = 0_{n_{F_a}}.\end{aligned} \tag{4.35}$$

The proof follows the same steps as the proof of Lemma 4.4. We first analyze the objective function of (4.35). Since optimal FDI attacks are purely additive ($\Lambda_u = I_{n_u}$, $\Lambda_y = I_{n_y}$) and the system is linear, we can write

$$\mathbb{E}\{x_e(k)\,;\, d\} = \mu_h(k) + \mu_a(k).$$

Here, $\mu_h(k)$ (resp. $\mu_a(k)$) characterizes influence of $x_e(0)$ and y_r (resp. $a_{0:k-1}$) on the mean value of $x_e(k)$. Firstly, since the system has entered the stationary regime prior to the attack and optimal FDI attacks do not affect the system matrices ($\tilde{A}_e = A_e$, $\tilde{B}_e = B_e$, $\tilde{E}_e = E_e$), we have $\mu_h(k) = T_0 y_r$. Secondly, from (C1) and (3.2), we have $\mu_a(k) = \mathcal{C}_k(\tilde{A}_e, \tilde{G}_e)a_{0:k-1}$. Therefore, it follows that

$$\mathbb{E}\{z_i(j)\,;\, d\} = \mathbb{E}\{\tilde{C}_z(i,:)x_e(j)\,;\, d\} = \tilde{C}_z(i,:)\big(T_0 y_r + \mathcal{C}_j(\tilde{A}_e, \tilde{G}_e)a_{0:j-1}\big).$$

Hence, the objective functions of (4.34) and (4.35) are equal.

Next, from the definition of Q, we have $\|Qd\|_\infty = \|Q_{y_r} y_r\|_\infty$. Since $\tilde{r}$ is decoupled from y_r when $\Lambda_u = I_{n_u}$ and $\Lambda_y = I_{n_y}$ (see the proof of Proposition 4.6), we have $\|T_R d\|_2^2 = \|T'_R a_{0:N}\|_2^2$. Finally, since no constraints are imposed on $a_{0:N}$ in the optimal FDI attack strategy, we can simply discard the constraint $Fd = 0_{n_{F_a}}$. ■

Lemma 4.6. *Let i be arbitrarily selected from $\{1, \ldots, n_z\}$. Under the optimal FDI attack strategy, $z_i(j)$ has the variance $\Sigma_Z(i,i)$ for any $j \in \{1, \ldots, N\}$.*

Proof. Since optimal FDI attacks are deterministic and purely additive, only the mean value of $z(k)$ changes. Hence, the variance of the critical state z_i is equal to the one in stationary regime, and remains the same for any time step $j \in \{1, \ldots, N\}$. Since the critical state $z_i(1)$ has the variance $\Sigma_Z(i,i)$, the claim holds. ■

We are now ready to prove Proposition 4.7. We first establish $\mu^*(i,j) \leq \mu^*(i,N)$ for any $j \in \{1, \ldots, N\}$ using contradiction. Let us assume that there exists $j^* \in \{1, \ldots, N-1\}$ for which $\mu^*(i,j^*) > \mu^*(i,N)$ holds. Let

$$d^* = \begin{bmatrix} a^*_{0:N} \\ y^*_r \end{bmatrix}$$

be a solution of $\mathcal{P}^{(1)}_{ij^*}$, and let us define

$$k^* = N - j^*, \qquad a'_{0:N} = \begin{bmatrix} 0_{k^* n_a} \\ a^*_{0:j^*} \end{bmatrix}, \qquad d' = \begin{bmatrix} a'_{0:N} \\ y^*_r \end{bmatrix}.$$

In the following, we show that d' is a feasible point of $\mathcal{P}^{(1)}_{iN}$ (Claim 1) and that it yields the objective value larger than or equal to $\mu^*(i,j^*)$ (Claim 2). This contradicts existence of j^* and concludes the first part of the proof.

Claim 1. From Lemma 4.5, it follows that $\mathcal{P}^{(1)}_{iN}$ can be rewritten as (4.34). The first constraint of (4.34) is only affected by the references. Hence, d' automatically satisfies this constraint. To show that d' satisfies the second constraint of (4.34), we rewrite T'_R in the form $T'_R = \begin{bmatrix} T'_{R1} & T'_{R2} \\ T'_{R3} & T'_{R4} \end{bmatrix}$, where

(i) T'_{R1} maps $a_{0:k^*-1}$ to $r_{0:k^*-1}$;

(ii) T'_{R2} maps $a_{k^*:N}$ to $r_{0:k^*-1}$;

(iii) T'_{R3} maps $a_{0:k^*-1}$ to $r_{k^*:N}$;

(iv) T'_{R4} maps $a_{k^*:N}$ to $r_{k^*:N}$.

Note that T'_{R2} is equal to zero due to the causality of the system. By plugging d' into the second constraint of (4.34), we obtain

$$\|T'_R a'_{0:N}\|_2^2 = \left|\left|\begin{bmatrix} T'_{R1}a'_{0:k^*-1} + T'_{R2}a'_{k^*:N} \\ T'_{R3}a'_{0:k^*-1} + T'_{R4}a'_{k^*:N} \end{bmatrix}\right|\right|_2^2 \overset{(*)}{=} \|T'_{R4}a^*_{0:j^*}\|_2^2, \tag{4.36}$$

where (*) follows from $a'_{0:k^*-1} = 0_{k^* n_a}$, $a'_{k^*:N} = a^*_{0:j^*}$, and T'_{R2} being zero. Let us now rewrite T'_R in the form $T'_R = \begin{bmatrix} T''_{R1} & T''_{R2} \\ T''_{R3} & T''_{R4} \end{bmatrix}$, where

(i) T''_{R1} maps $a_{0:j^*}$ to $r_{0:j^*}$;

(ii) T''_{R2} maps $a_{j^*+1:N}$ to $r_{0:j^*}$;

(iii) T''_{R3} maps $a_{0:j^*}$ to $r_{j^*+1:N}$;

(iv) T''_{R4} maps $a_{j^*+1:N}$ to $r_{j^*+1:N}$.

Additionally, note that T''_{R2} is equal to zero due to the causality of the system, and that $T''_{R1} = T'_{R4}$ due to the fact that the system is time invariant. By plugging d^* into the second constraint of (4.34), we obtain

$$\|T'_R a^*_{0:N}\|_2^2 \overset{(*)}{=} \left|\left|\begin{bmatrix} T''_{R1}a^*_{0:j^*} \\ T''_{R3}a^*_{0:j^*} + T''_{R4}a^*_{j^*+1:N} \end{bmatrix}\right|\right|_2^2 \geq \|T''_{R1}a^*_{0:j^*}\|_2^2, \tag{4.37}$$

where (*) follows from T''_{R2} being zero. We then have

$$\epsilon' \geq \|T'_R a^*_{0:N}\|_2^2 \overset{(4.37)}{\geq} \|T''_{R1}a^*_{0:j^*}\|_2^2 \overset{T''_{R1}=T'_{R4}}{=} \|T'_{R4}a^*_{0:j^*}\|_2^2 \overset{(4.36)}{=} \|T'_R a'_{0:N}\|_2^2.$$

Hence, d' satisfies the second constraint of (4.34), and Claim 1 holds.

Claim 2. We now prove that d' yields the objective value larger than or equal to $\mu^*(i,j^*)$. Let $M = \mathcal{C}_N(\tilde{A}_e, \tilde{G}_e)$ be the mapping from $a_{0:N-1}$ to $x_e(N)$. Note that we can write $M = [M_1 \;\; M_2]$, where M_1 maps $a_{0:k^*-1}$ to $x_e(N)$ and M_2 maps $a_{k^*:N-1}$ to $x_e(N)$. Since the system is time invariant, we have $M_2 = \mathcal{C}_{j^*}(\tilde{A}_e, \tilde{G}_e)$.

From Lemma 4.5, the objective value of $\mathcal{P}_{iN}^{(1)}$ in d' equals to

$$\begin{aligned}\left|\tilde{C}_z(i,:)\begin{bmatrix} M & T_0 \end{bmatrix}\begin{bmatrix} a'_{0:N-1} \\ y^*_r \end{bmatrix}\right| &\overset{(*)}{=} \left|\tilde{C}_z(i,:)\left[M_2 a'_{k^*:N-1} + T_0 y^*_r\right]\right| \\ &\overset{(**)}{=} \left|\tilde{C}_z(i,:)\left[\mathcal{C}_{j^*}(\tilde{A}_e, \tilde{G}_e)a^*_{0:j^*-1} + T_0 y^*_r\right]\right| \overset{(4.34)}{=} \mu^*(i,j^*),\end{aligned}$$

where (*) follows from $a'_{0:k^*-1} = 0_{k^* n_a}$, and (**) follows from $a'_{k^*:N-1} = a^*_{0:j^*-1}$ and $M_2 = \mathcal{C}_{j^*}(\tilde{A}_e, \tilde{G}_e)$. In words, the objective value of $\mathcal{P}_{iN}^{(1)}$ for a feasible point d'

is equal to $\mu^*(i, j^*)$. Since d' is a feasible point of $\mathcal{P}_{iN}^{(1)}$, it yields the objective value smaller than the optimal value $\mu^*(i, N)$. Hence, $\mu^*(i, j^*) \leq \mu^*(i, N)$ has to hold.

We now prove that $P^*(i, j) \leq P^*(i, N)$ for any $j \in \{1, ..., N\}$. Under Assumption 4.3, we know that a solution d_i^* of $\mathcal{P}_{ij}^{(1)}$ exists. Denote by $\mathcal{N}(\mu_{ij}, \sigma_{ij}^2)$ the distribution of $z_i(j)$ assuming $d = d_i^*$. Observe that:

(i) d_i^* is also a solution of $\mathcal{P}_{ij}^{(2)}$ (established in the proof of Theorem 4.1);

(ii) the mean value magnitude $|\mu_{ij}|$ for a fixed i is largest for $j = N$ (this was established earlier in the proof);

(iii) the variance σ_{ij}^2 is equal to $\Sigma_Z(i, i)$ for every $j \in \{1, \ldots, N\}$ (Lemma 4.6);

(iii) if $\tilde{z} \sim \mathcal{N}(\mu, \sigma^2)$, then $\mathbb{P}(|\tilde{z}| > 1\,;\, \mu)$ is monotonically increasing with $|\mu|$ when σ^2 is fixed (established in the proof of Theorem 4.1).

From (i)–(iv), it directly follows that $P^*(i, j) \leq P^*(i, N)$.

Chapter 5

Security measure allocation

The security measure allocation problem consists of computing the least expensive subset of security measures that prevents all the critical vulnerability combinations. This problem is challenging for two reasons. Firstly, to construct an instance of the security measure allocation problem, we need to find the critical vulnerability combinations. If the number of vulnerabilities is large, then it is infeasible to simply search through all the vulnerability combinations to find those that are critical. Secondly, the security measure allocation problem proves to be NP-hard. Hence, known polynomial-time algorithms cannot solve this problem.

To tackle the first challenge, we introduce several tools that can be used to systematically search for the critical vulnerability combinations. Based on these tools, we propose an algorithm that provably finds the critical combinations needed to construct the security measure allocation problem. To tackle the second challenge, we introduce two approaches. The first approach is to simplify the problem, and then use an integer linear program solver to compute a solution. The second approach consists of proving that the problem possesses a suitable submodular structure. This enables us to use a polynomial-time algorithm to compute a suboptimal solution with performance guarantees. We also investigate how to optimize these guarantees. The applicability of our approach is demonstrated on a control system that is used for regulating temperatures. Additionally, we explain how the impact estimation framework from Chapter 4 can be combined with the security measure allocation framework from this chapter.

The chapter is organized as follows. Section 5.1 introduces the security measure allocation problem. Section 5.2 presents the algorithm that systematically constructs the problem. Section 5.3 establishes NP-hardness of the problem and introduces two suboptimal approaches to tackle it. Section 5.4 discusses the security measure allocation problem on an example. Section 5.5 concludes the chapter. The appendix contains lengthy proofs and numerical values of some matrices.

5.1 Model setup and problem formulation

This section introduces the control system model, the risk model, and the security measure allocation problem.

5.1.1 Control system model

We characterize the control system through the following four finite sets:

(i) the set of vulnerabilities $\mathcal{V}$ present in the system;

(ii) the set of security measures $\mathcal{M}$ that can prevent the vulnerabilities;

(iii) the set of actuators $\mathcal{U}$;

(iv) the set of sensors $\mathcal{Y}$.

A vulnerability $v \in \mathcal{V}$ can model a communication link without protection, lack of anti-virus software on computers in a control center, or insufficient physical protection of some control equipment. By exploiting a vulnerability v, the attacker gains control over the actuators $U_v \subseteq \mathcal{U}$ and the sensors $Y_v \subseteq \mathcal{Y}$. He/she can then use these components to attack the physical plant. If vulnerabilities $V \subseteq \mathcal{V}$ are exploited by the attacker, which we refer to as a scenario V, then the actuators U_V and the sensors Y_V under the attacker's control can be written as follows:

$$U_V = \bigcup_{v \in V} U_v, \qquad Y_V = \bigcup_{v \in V} Y_v.$$

The vulnerabilities can be prevented by deploying security measures from $\mathcal{M}$. A security measure $m \in \mathcal{M}$ can model the encryption of a communication link, the installation of anti-virus software, or the deployment of better physical protection. With every $m \in \mathcal{M}$, we associate the cost of deployment $c_m \in \mathbb{R}^+$ and the vulnerabilities $V_m \subseteq \mathcal{V}$ prevented by m. If security measures $M \subseteq \mathcal{M}$ are deployed, then the total cost c_M and the prevented vulnerabilities V_M can be written as follows:

$$c_M = \sum_{m \in M} c_m, \qquad V_M = \bigcup_{m \in M} V_m. \tag{5.1}$$

We assume that the prevented vulnerabilities cannot be exploited by the attacker. Based on this assumption, we say that a scenario V is prevented if it contains any of the prevented vulnerabilities. That is, if $V \cap V_M \neq \emptyset$ holds. We also assume that every vulnerability can be prevented by at least one security measure, which ensures feasibility of the security measure allocation problem. The set containing all the security measures that prevent v is denoted by M_v.

5.1.2 Risk model and critical scenarios

To introduce the risk model, we need to define an impact function and a likelihood function. The impact function $I : 2^{\mathcal{U}} \times 2^{\mathcal{Y}} \to \mathbb{R}^+$ characterizes the negative impact that the attacker can inflict to a physical plant through compromised sensors and actuators. In Section 5.4, we use the impact estimation framework from Chapter 4 to form the impact function I. The likelihood function $\pi : 2^{\mathcal{V}} \to \mathbb{R}^+$ is typically a score representing the belief of a scenario occurring [17]. This score can be formed based on expert knowledge [17, 32, 33], or by using tools developed for this purpose [34, 35]. Factors that can be used to estimate the likelihood include site architecture, security measures that are already installed, cost of attack, and technical difficulty [32]. We assume that I and π have the following properties.

Assumption 5.1. *Let $(U, Y), (U', Y') \in 2^{\mathcal{U}} \times 2^{\mathcal{Y}}$. If $U \subseteq U'$ and $Y \subseteq Y'$, then $I(U, Y) \leq I(U', Y')$ holds.*

Assumption 5.2. *Let $V, V' \in \mathcal{V}$. If $V \subseteq V'$, then $\pi(V) \geq \pi(V')$ holds.*

Assumption 5.1 states that the more resources the attacker compromises, the higher impact he/she can inflict. Assumption 5.2 states that more vulnerabilities the attacker exploits, the less likely the scenario becomes.

We are now ready to introduce the risk model. We consider the model from [193], where the risk was modeled as a set of triplets $\langle$Scenario, Impact, Likelihood$\rangle$. In our context, these triplets can be defined by $\langle V, I(U_V, Y_V), \pi(V) \rangle$. Here, a subset of vulnerabilities V models an attack scenario, $I(U_V, Y_V)$ is the impact if the scenario V occurs, and $\pi(V)$ is the likelihood of V occurring.

We now use the risk model to define the critical scenarios, which are crucial for defining the security measure allocation problem. Essentially, a scenario is critical if it is sufficiently likely to occur and can lead to a sufficiently high impact.

Definition 5.1. *A scenario $V \subseteq \mathcal{V}$ is critical if $I(U_V, Y_V) \geq I_{\min}$ and $\pi(V) \geq \pi_{\min}$, where $I_{\min} \in \mathbb{R}^+$ and $\pi_{\min} \in \mathbb{R}^+$ are predefined thresholds.*

Remark 5.1. *The thresholds $I_{\min}$ and $\pi_{\min}$ should be seen as tuning parameters. One way to tune these thresholds is to initially set them relatively high. In this way, we restrict our attention to scenarios that can have high impact and are highly likely to occur. If these scenarios are inexpensive to prevent, then we can decrease the thresholds and re-solve the problem to prevent less dangerous scenarios.*

5.1.3 Problem formulation

Let $\mathcal{C} \subseteq 2^{\mathcal{V}}$ be the set of all the critical scenarios. The security measure allocation problem consists of computing a subset of security measures that prevents all the

scenarios from $\mathcal{C}$ and has the minimum cost. This problem can be formulated as the following integer linear program:

Problem 5.1. *Security measure allocation*

$$\begin{aligned}
\underset{x}{\text{minimize}} \quad & \sum_{m \in \mathcal{M}} c_m x_m \\
\text{subject to} \quad & \sum_{m \in M_v} x_m = y_v, && \forall v \in \mathcal{V}, && \text{(C1)} \\
& \sum_{v \in V} y_v \geq 1, && \forall V \in \mathcal{C}, && \text{(C2)} \\
& x_m \in \{0, 1\}, && \forall m \in \mathcal{M}. && \text{(C3)}
\end{aligned}$$

Here, every security measure $m \in \mathcal{M}$ is modeled by a decision variable x_m, which equals to one (resp. zero) if m is deployed (resp. not deployed). Thus, the objective function is the total cost of the deployed security measures. Every vulnerability v is modeled with a variable y_v. Constraint (C1) imposes that y_v is greater than or equal to one if v is prevented. Otherwise, y_v is equal to zero. Therefore, Constraint (C2) ensures that all the critical scenarios are prevented.

Solving Problem 5.1 is difficult for two reasons. Firstly, we have to find the set of critical scenarios $\mathcal{C}$ to construct Constraint (C2). This is challenging because the number of possible scenarios equals to the number of subsets of $\mathcal{V}$. Thus, searching through all the subsets of $\mathcal{V}$ to find those that are critical is not tractable when the cardinality of $\mathcal{V}$ is large. Secondly, Section 5.3 shows that Problem 5.1 is NP-hard. Hence, known polynomial-time algorithms cannot solve Problem 5.1. In the following two sections, we address these issues.

5.2 Constructing the security measure allocation problem

This section presents Algorithm 5.1 that systematically constructs Constraint (C2) of Problem 5.1. Before we present Algorithm 5.1, we introduce systematic search tools that the algorithm utilizes.

5.2.1 Systematic search tools

Reducing number of explored scenarios

The first way to reduce the number of explored scenarios is by using the fact that the likelihood function is nonincreasing. Namely, if we find a scenario V for which $\pi(V) < \pi_{\min}$, then we do not need to investigate scenarios that contain V. These scenarios have likelihoods lower than $\pi_{\min}$, and hence, they are not critical.

Lemma 5.1. *If $\pi(V) < \pi_{\min}$ holds for a scenario V, then any scenario V' that satisfies $V \subseteq V'$ is not critical.*

Proof. From Assumption 5.2, we have $\pi(V') \leq \pi(V)$. Hence, $\pi(V') < \pi_{\min}$ holds. From the latter and Definition 5.1, it follows that V' is not critical. ■

The second way is by showing that we do not need to find the entire set of critical scenarios $\mathcal{C}$. Instead, it suffices to find a suitable subset of $\mathcal{C}$. We use an example to explain the idea.

Example 5.1. *Let $\mathcal{V} = \{v_1, v_2, v_3\}$, $\mathcal{C} = \big\{\{v_1\}, \{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}\big\}$, and consider the subset $\mathcal{C}' = \big\{\{v_1\}, \{v_2, v_3\}\big\}$ of $\mathcal{C}$. The scenarios from $\mathcal{C}'$ are prevented if and only if the set of prevented vulnerabilities is one of the following: $\{v_1, v_2\}$, $\{v_1, v_3\}$, or $\{v_1, v_2, v_3\}$. One can observe that the same holds for $\mathcal{C}$. Hence, if we find a subset of security measures $M \subseteq \mathcal{M}$ that prevents all the scenarios from $\mathcal{C}'$, then all the scenarios from $\mathcal{C}$ are also prevented.*

Motivated by the previous example, we define a sufficient representation of $\mathcal{C}$.

Definition 5.2. *A subset $\hat{\mathcal{C}} \subseteq \mathcal{C}$ is a sufficient representation of $\mathcal{C}$, if every $V' \subseteq \mathcal{V}$ that satisfies $V' \cap V \neq \emptyset$ for all $V \in \hat{\mathcal{C}}$ also satisfies $V' \cap V \neq \emptyset$ for all $V \in \mathcal{C}$.*

In words, a sufficient representation $\hat{\mathcal{C}}$ is a subset of $\mathcal{C}$ with the property that when we prevent all the critical scenarios from $\hat{\mathcal{C}}$, all the critical scenarios from $\mathcal{C}$ are also prevented. Thus, it sufficies to find any sufficient representation of $\mathcal{C}$ to construct (C2). We also remark that a sufficient representation is generally not unique. Indeed, we see from Example 5.1 that both $\mathcal{C}$ and $\mathcal{C}'$ satisfy Definition 5.2.

In the following, we focus on finding a sufficient representation that has the minimum cardinality. The reason is twofold. Firstly, this representation helps us to reduce the number of critical scenarios we need to find. Secondly, Section 5.3 shows that such a representation is also beneficial for solving the security measure allocation problem. The following lemma characterizes the sufficient representation of minimum cardinality, and establishes its uniqueness.

Lemma 5.2. *Let $\hat{\mathcal{C}}^*$ be a sufficient representation of $\mathcal{C}$. Then $\hat{\mathcal{C}}^*$ is the unique sufficient representation of minimum cardinality if and only if $V \nsubseteq V'$ holds for any two scenarios $V, V' \in \hat{\mathcal{C}}^*$.*

Proof. We refer the reader to Appendix 5.A. ■

Put differently, if a scenario V belongs to $\hat{\mathcal{C}}^*$, then any other scenario that contains V does not belong to $\hat{\mathcal{C}}^*$. Hence, if we find a scenario that belongs to $\hat{\mathcal{C}}^*$, then we do not need to explore any other scenario that contains V.

Reducing number of executions of the impact set function

We now provide a way to reduce the number of executions of the impact function I. This can be useful when I is costly to evaluate. The idea is to store combinations of sensors and actuators for which we evaluate I. These combinations can then be divided into the lists $\mathcal{K}_+$ and $\mathcal{K}_-$. The list $\mathcal{K}_+$ contains combinations of sensors and actuators for which the impact is greater than or equal to $I_{\min}$. The list $\mathcal{K}_-$ contains combinations that result in the impact less than $I_{\min}$. The following result can then be established.

Lemma 5.3. *Let $(U,Y),(U',Y') \in 2^{\mathcal{U}} \times 2^{\mathcal{Y}}$. The following claims hold:*

(i) If (U',Y') belongs to $\mathcal{K}_+$, $U' \subseteq U$, and $Y' \subseteq Y$, then $I(U,Y) \geq I_{\min}$;

(ii) If (U',Y') belongs to $\mathcal{K}_-$, $U \subseteq U'$, and $Y \subseteq Y'$, then $I(U,Y) < I_{\min}$.

Proof. (i) Firstly, from $U' \subseteq U$, $Y' \subseteq Y$, and Assumption 5.1, it follows that $I(U',Y') \leq I(U,Y)$. Secondly, since (U',Y') belongs to the list $\mathcal{K}^+$, we have $I(U',Y') \geq I_{\min}$. Thus, $I(U,Y) \geq I_{\min}$ holds.

(ii) In this case, $U \subseteq U'$, $Y \subseteq Y'$, and Assumption 5.1 imply that $I(U,Y) \leq I(U',Y')$. Since (U',Y') belongs to the list $\mathcal{K}^-$, we have $I(U',Y') < I_{\min}$. Therefore, we conclude that $I(U,Y) < I_{\min}$ holds. ■

Lemma 5.3 can be used to limit the number of executions of I as follows. Assume that we need to check if $I(U_V,Y_V) \geq I_{\min}$ holds. If the combination (U_V,Y_V) contains a combination from $\mathcal{K}_+$, then we can conclude that $I(U_V,Y_V) \geq I_{\min}$ without evaluating the impact function. Similarly, if the combination (U_V,Y_V) is contained in a combination from $\mathcal{K}_-$, then we know that $I(U_V,Y_V) < I_{\min}$ holds.

Power set enumeration tree

The power set enumeration tree is a graph representation of the power set $2^{\mathcal{V}}$ [194]. Each node of the tree represents one subset of $\mathcal{V}$. The tree has $|\mathcal{V}|+1$ layers enumerated with $0,1,\ldots,|\mathcal{V}|$, where the p^{th} layer contains all the scenarios with the cardinality p. The edges of the tree are determined as follows:

(i) The node $\emptyset$ is connected to all the nodes from the first layer.

(ii) Let $p \in \mathbb{N}$. A node V from the p^{th} layer, is connected to a node $V \cup v_j$ from the $(p+1)^{\text{th}}$ layer if $j < i$ holds for every $v_i \in V$.

For instance, the power set enumeration tree when $\mathcal{V} = \{v_1,v_2,v_3\}$ is shown in Figure 5.1. Note that the node $\{v_2\}$ is connected to $\{v_1,v_2\}$, but not to $\{v_2,v_3\}$.

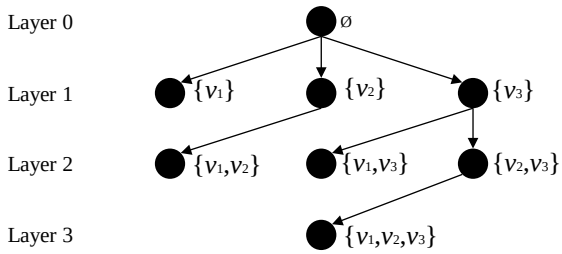

Figure 5.1: The power set enumeration tree when $\mathcal{V} = \{v_1, v_2, v_3\}$.

5.2.2 Algorithm 5.1: Constructing the sufficient representation of minimum cardinality

Recall that we can construct Constraint (C2) by finding the sufficient representation of minimum cardinality. In the following, we introduce Algorithm 5.1 that systematically searches for this sufficient representation.

Working principle of Algorithm 5.1

Algorithm 5.1 explores the power set enumeration tree by layers. In each layer, the algorithm performs scenario classification and scenario generation.

The scenario classification is performed as follows. Algorithm 5.1 receives the list $\mathcal{C}_L$ of scenarios that should be classified in the current layer. For every scenario V from $\mathcal{C}_L$, the algorithm checks if $\pi(V)$ is lower than $\pi_{\min}$. If it is, then the algorithm moves to the next scenario. Otherwise, the algorithm checks if $I(U_V, Y_V) \geq I_{\min}$ holds. The algorithm first tries to determine this based on the lists $\mathcal{K}_+$ and $\mathcal{K}_-$ (Lemma 5.3). If that is not possible, then the impact function is evaluated, and (U_V, Y_V) is stored in $\mathcal{K}_+$ or $\mathcal{K}_-$ depending on the impact. If $I(U_V, Y_V) \geq I_{\min}$, then the algorithm adds V to the list $\tilde{\mathcal{C}}^*$, which corresponds to the sufficient representation of minimum cardinality. Otherwise, V is stored in the list $\mathcal{C}_O$, which is later used to generate new scenarios. When the classification is completed, the list $\mathcal{C}_L$ is emptied, and the algorithm moves to the generation step.

The generation of scenarios is performed as follows. If a scenario V is critical or with a likelihood lower than $\pi_{\min}$, then any scenario that contains V does not belong to the sufficient representation of minimum cardinality (Lemmas 5.1 and 5.2). Hence, we only need to generate new scenarios based on the list $\mathcal{C}_O$. For every $V \in \mathcal{C}_O$, the algorithm adds a scenario $V \cup v_j$ to $\mathcal{C}_L$ if: (i) $j < i$ for every $v_i \in V$; and (ii) there does not exist a critical scenario in $\tilde{\mathcal{C}}^*$ contained in $V \cup v_j$. The first rule follows from the power set enumeration tree structure. The second rule ensures that we obtain the sufficient representation of minimum cardinality (Lemma 5.2).

Algorithm 5.1 Finding the sufficient representation of minimum cardinality

Input: $\mathcal{V} = \{v_1, \ldots, v_{n_v}\}$, $\pi_{\min}$, $I_{\min}$, π, I
Output: $\tilde{\mathcal{C}}^*$
Form the list $\mathcal{C}_L$ (initialized with scenarios of cardinality one)
Form the list $\tilde{\mathcal{C}}^*$ (initially empty)
Form the list $\mathcal{C}_O$ (initially empty)
Form the lists $\mathcal{K}_+, \mathcal{K}_-$ (initially empty)
while $\mathcal{C}_L \neq \emptyset$ **do**
  % Classification step
  for every scenario $V \in \mathcal{C}_L$ **do**
    if $\pi(V) \geq \pi_{\min}$ **then**
      Determine if $I(U_V, Y_V) \geq I_{\min}$ (try using the lists $\mathcal{K}_+$ and $\mathcal{K}_-$ first)
      if $I(U_V, Y_V) \geq I_{\min}$ **then**
        Add V to $\tilde{\mathcal{C}}^*$
        Add (U_V, Y_V) to $\mathcal{K}_+$ if I was evaluated
      else
        Add V to $\mathcal{C}_O$
        Add (U_V, Y_V) to $\mathcal{K}_-$ if I was evaluated
      end if
    end if
  end for
  % Generation step
  Empty $\mathcal{C}_L$
  for every $V \in \mathcal{C}_O$ **do**
    Find a vulnerability $v_i \in V$ with the minimum index
    for every $v_j \in \{v_1, \ldots, v_{i-1}\}$ **do**
      if there does not exist a scenario from $\tilde{\mathcal{C}}^*$ contained in $V \cup v_j$ **then**
        Add $V \cup v_j$ to $\mathcal{C}_L$
      end if
    end for
  end for
  Empty $\mathcal{C}_O$
end while

Once the new scenarios are generated, $\mathcal{C}_O$ is emptied. If $\mathcal{C}_L$ is empty, then the algorithm returns $\tilde{\mathcal{C}}^*$ and terminates. Otherwise, the algorithm moves to the classification step in the next layer, and the whole procedure is repeated again.

Properties of Algorithm 5.1

Before we move to the next section, we list some properties of Algorithm 5.1. Firstly, Algorithm 5.1 returns the sufficient representation of minimum cardinality. This

property of Algorithm 5.1 is formally established in the following theorem.

Theorem 5.1. *If $\tilde{\mathcal{C}}^*$ is the set of scenarios returned by Algorithm 5.1 and $\hat{\mathcal{C}}^*$ is the sufficient representation of minimum cardinality, then $\tilde{\mathcal{C}}^* = \hat{\mathcal{C}}^*$ holds.*

Proof. We refer the reader to Appendix 5.B. ■

Secondly, the running time of Algorithm 5.1 depends on many factors, including the choice of the functions I and π, the thresholds $I_{\min}$ and $\pi_{\min}$, and the size of $\mathcal{V}$ in a nontrivial way. Although it is expected that the number of scenarios we search through would be significantly reduced due to the systematic search tools, Algorithm 5.1 may still end up searching every subset of $\mathcal{V}$.

Based on the available time, the search can be restricted to the first n layers of the power set enumeration tree. In that case, the algorithm searches in the worst case $\sum_{i=1}^{n} \binom{|\mathcal{V}|}{i}$ scenarios. Moreover, let $\mathcal{C}_n$ be the set that contains all the critical scenarios with cardinality less than or equal to n. The following corollary states that if Algorithm 5.1 is restricted to search the first n layers of the tree, then the sufficient representation of minimum cardinality of $\mathcal{C}_n$ is returned.

Corollary 5.1. *Let $\mathcal{C}_n = \{V \in \mathcal{C} \mid |V| \leq n\}$. If Algorithm 5.1 is restricted to search the first n layers of the power set enumeration tree, then it returns the sufficient representation of minimum cardinality of the set $\mathcal{C}_n$.*

Proof. We refer the reader to Appendix 5.C. ■

Finally, if we update the lists $\mathcal{K}_+$ and $\mathcal{K}_-$ after the classification step, then the clasification of scenarios can be executed in parallel [20]. Since the classification step involves evaluating the impact and likelihood functions a large number of times, which is expected to be time consuming, significant reduction in execution time can be achieved with parallelization. However, to simplify the implementation of Algorithm 5.1, we do not consider parallelization in this chapter.

5.3 Solving the security measure allocation problem

This section establishes that the security measure allocation problem is NP-hard, and discusses two suboptimal approaches for solving it.

5.3.1 NP-hardness of the security measure allocation problem

The following proposition establishes NP-hardness of Problem 5.1. Thus, known polynomial-time algorithms cannot solve Problem 5.1.

Proposition 5.1. *The security measure allocation problem is NP-hard.*

Proof. To prove the claim, we show that every instance of the set cover problem introduced in Section 3 can be mapped into Problem 5.1. Since the set cover problem is NP-hard [157], the claim of the proposition immediately follows. Note that every instance of the set cover problem is determined by a universe set $\mathcal{S}$ and a set of subsets $\mathcal{S}_c$. For any $\mathcal{S}$ and $\mathcal{S}_c = \{S_1, \ldots, S_n\}$, we can establish the following mapping between the set cover problem and Problem 5.1: (i) the set of vulnerabilities $\mathcal{V}$ equals $\mathcal{S}$; (ii) the set of critical scenarios $\mathcal{C}$ contains only the subsets of $\mathcal{V}$ with cardinality equal to one; and (iii) we set $\mathcal{M} = \{m_1, \ldots, m_n\}$, $V_{m_1} = S_1, \ldots, V_{m_n} = S_n$, and $c_m = 1$ for every $m \in \mathcal{M}$. Problem 5.1 then reduces to the set covering problem, where the goal is to cover the vulnerability set $\mathcal{V}$ using the sets $V_{m_1}, \ldots V_{m_n}$. This shows that every instance of the NP-hard set cover problem can be mapped into Problem 5.1, and the proof is completed. ■

In the following, we propose two sub-optimal approaches to tackle Problem 5.1. We then show in Section 5.4 that these approaches may compute an exact or a good approximate solution of Problem 5.1 efficiently in spite of NP-hardness.

5.3.2 Approach 1: Simplifying the problem

The first approach consists of two steps. The first step is to simplify Problem 5.1 using the sufficient representation of minimum cardinality $\hat{\mathcal{C}}^*$. Recall that Constraint (C2) imposes that every scenario from $\mathcal{C}$ has to be prevented. Since by preventing all the scenarios from $\hat{\mathcal{C}}^*$ we also prevent all the scenarios from $\mathcal{C}$, Constraint (C2) can be substituted with the following constraint:

$$\sum_{v \in V} y_v \geq 1, \qquad \forall V \in \hat{\mathcal{C}}^*.$$

In this way, we obtain a simplified problem whose number of constraints is by $|\mathcal{C}| - |\hat{\mathcal{C}}^*|$ smaller than the number of constraints of Problem 5.1. The second step is to use an integer linear program solver to tackle the simplified problem.

5.3.3 Approach 2: Exploiting submodularity

Let M be a subset of security measures. Recall that V_M is the subset of vulnerabilities prevented by M, and that scenarios that have a non-empty intersection with V_M are prevented. To model this relation, we define a gain function

$$f_V(M) = \min\{|V_M \cap V|, 1\}$$

with every scenario $V \in \mathcal{C}$. If a scenario V is prevented, then $f_V(M) = 1$. Otherwise, $f_V(M) = 0$ holds. Next, we introduce the total gain

$$F(M) = \sum_{V \in \mathcal{C}} f_V(M) = \sum_{V \in \mathcal{C}} \min\{|V_M \cap V|, 1\}.$$

Note that $F(M) = |\mathcal{C}|$ once all the scenarios from $\mathcal{C}$ are prevented. From the latter, it follows that Problem 5.1 can be reformulated as follows:

$$\begin{aligned} \underset{M}{\text{minimize}} \quad & \sum_{m \in M} c_m \\ \text{subject to} \quad & F(M) = |\mathcal{C}|. \end{aligned} \tag{5.2}$$

Here, the objective function equals to the total cost of deployed security measures. The constraint guarantees that the scenarios from $\mathcal{C}$ are prevented.

In the following, we show that polynomial-time Algorithm 3.1 can compute an approximate solution of the problem (5.2) with performance guarantees. The proof consists of showing that (5.2) has the same submodular structure as Problem 3.1.

Theorem 5.2. *Let c^* be the optimal value of the problem (5.2), c_G be the value found by Algorithm 3.1, and $H(n) = \sum_{i=1}^{n} i^{-1}$. The following then holds:*

$$\frac{c_G}{c^*} \leq H\big(\max_{m \in \mathcal{M}} F(m)\big). \tag{5.3}$$

Proof. We refer the reader to Appendix 5.D. ■

Theorem 5.2 implies that the objective value obtained using Algorithm 3.1 is upper bounded by $H\big(\max_{m \in \mathcal{M}} F(m)\big)c^*$. We point out to two properties of this bound. Firstly, the bound (5.3) has logarithmic growth with respect to $\max_{m \in \mathcal{M}} F(m)$. Secondly, the bound characterizes the worst case performance guarantees of Algorithm 3.1, which means that the algorithm can perform better in practice.

However, one issue that we have not mentioned so far is that the problem (5.2) is difficult to construct. The reason is that we need to find the set of critical scenarios $\mathcal{C}$ to form the total gain F. To overcome this issue, we can form the total gain based on the sufficient representation of minimum cardinality $\hat{\mathcal{C}}^*$:

$$\hat{F}^*(M) = \sum_{V \in \hat{\mathcal{C}}^*} f_V(M). \tag{5.4}$$

The constraint of the problem (5.2) can then be substituted with $\hat{F}^*(M) = |\hat{\mathcal{C}}^*|$.

In fact, one can use any other sufficient representation to form the total gain. Yet, by using $\hat{\mathcal{C}}^*$ for this purpose, we achieve an additional benefit. Particularly, note that the bound (5.3) is dependent on the function used in the constraint of the problem (5.2). We now prove that if we form the total gain using $\hat{\mathcal{C}}^*$, then we minimize the bound. Therefore, the worst case performance guarantees of Algorithm 3.1 are optimized when the total gain is formed based on $\hat{\mathcal{C}}^*$.

Proposition 5.2. *Let $\hat{\mathcal{C}}^*$ be the sufficient representation of minimum cardinality, $\hat{\mathcal{C}}$ be any other sufficient representation, $\hat{F}^*$ be given by (5.4), and $\hat{F}(M) = \sum_{V \in \hat{\mathcal{C}}} \min\{|V_M \cap V|, 1\}$. The following then holds:*

$$H\big(\max_{m \in \mathcal{M}} \hat{F}^*(m)\big) \leq H\big(\max_{m \in \mathcal{M}} \hat{F}(m)\big).$$

Proof. We refer the reader to Appendix 5.E. ■

We conclude this section with the following remark.

Remark 5.2. *If Algorithm 5.1 is stopped after n layers and $\hat{\mathcal{C}}^*$ is not obtained, then the discussion from this section would hold for the set of critical scenarios $\mathcal{C}_n$ and the sufficient representation of minimum cardinality of $\mathcal{C}_n$.*

5.4 Illustrative examples

This section illustrates how our security measure allocation framework can be used in practice, explains how it can be combined with the impact estimation framework from Chapter 4, and tests how fast can we construct and solve Problem 5.1. The experiments are performed on Intel Core i7-8650U computer.

5.4.1 Model: A control system for regulating temperatures

We consider a control system that is used for regulating temperatures within five identical areas. We first introduce the physical part of the system (Figure 5.2 (a)), which was derived in [195]. The i^{th} area is modeled with the states $x_i = [T_{ai}\ T_{wi}\ P_i]^T$. Here, T_{ai} is the temperature of the i^{th} area, T_{wi} is the temperature of the i^{th} evaporator's lumped coil wall, and P_i is the refrigerant's pressure after leaving the i^{th} evaporator. The control actions in the i^{th} area are denoted by $u_i = [\omega_{fi}\ a_{vi}]^T$, where ω_{fi} is the speed of the i^{th} evaporator's fan, and a_{vi} is the control action that changes the fluid resistance of the i^{th} Electronic Expansion Valve (EEV). We model the compressor with a single state $x_c = P_C$, where P_C is the refrigerant pressure after leaving the compressor. The pressure P_C is regulated through the control action $u_c = \omega_K$, where ω_K is the speed of the compressor. For simplicity, we assume that every state of the system is measured.

In summary, the dynamics of the physical part are given by

$$\begin{aligned} x(k+1) &= \begin{bmatrix} A_1 & A_2 & \dots & A_2 & A_3 \\ A_2 & A_1 & \dots & A_2 & A_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ A_2 & A_2 & \dots & A_1 & A_3 \\ A_4 & A_4 & \dots & A_4 & A_5 \end{bmatrix} x(k) + \begin{bmatrix} B_1 & B_2 & \dots & B_2 & B_3 \\ B_2 & B_1 & \dots & B_2 & B_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ B_2 & B_2 & \dots & B_1 & B_3 \\ B_4 & B_4 & \dots & B_4 & B_5 \end{bmatrix} u(k) + v_x(k), \\ y(k) &= I_{16} x(k) + v_y(k), \end{aligned} \tag{5.5}$$

where $x = [x_1^T \dots x_5^T\ x_c]^T$, $u = [u_1^T \dots u_5^T\ u_c]^T$, and the matrices $A_1, \dots, A_5$ and $B_1, \dots, B_5$ are obtained by discretizing the system from [195] with a sampling time 0.1s. The numerical values of these matrices are provided in Appendix 5.F. Since [195] modeled the system as a noiseles, we adopt $\Sigma_{v_x} = \Sigma_{v_y} = 10^{-7}\ I_{16}$.

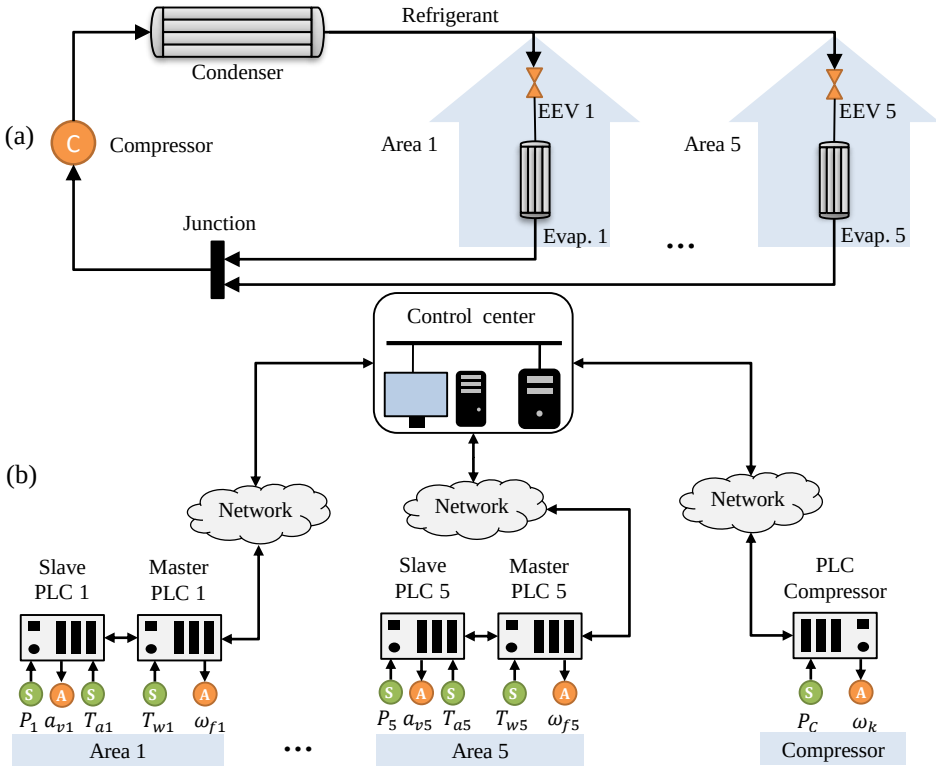

Figure 5.2: The control system used for regulating temperatures in five identical areas. (a) The physical part of the system. (b) The cyber part of the system.

The controller is given by

$$u(k) = \begin{bmatrix} L_{11} & L_{12} & \dots & L_{12} & L_{13} \\ L_{12} & L_{11} & \dots & L_{12} & L_{13} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ L_{12} & L_{12} & \dots & L_{11} & L_{13} \\ L_{14} & L_{14} & \dots & L_{14} & L_{15} \end{bmatrix} \hat{x}(k) + \begin{bmatrix} L_{31} & L_{32} & \dots & L_{32} \\ L_{32} & L_{31} & \dots & L_{32} \\ \vdots & \vdots & \ddots & \vdots \\ L_{32} & L_{32} & \dots & L_{31} \end{bmatrix} y_r, \tag{5.6}$$

where $\hat{x}(k)$ are the state estimates generated by the steady state Kalman filter, and $y_r \in \mathbb{R}^5$ are the references that are used to set desired temperatures in the areas. The references are assumed to satisfy $\|Q_{y_r} y_r\| \leq 1$, where $Q_{y_r} = 0.2\, I_5$. The numerical values of the controller matrices can be found in Appendix 5.F.

We assume the cyber-part of the system to be as shown in Figure 5.2 (b). The control actions are computed in the Control Center (CC) based on the collected measurements, and then sent for the execution to the controllers. In every area, there are two controllers: A master PLC and a slave PLC. The i^{th} master PLC

collects the measurement of T_{wi}, controls the fan through ω_{fi}, and communicates with the i^{th} slave PLC and CC. The i^{th} slave PLC collects the measurements of P_i and T_{ai}, controls the i^{th} EEV through a_{vi}, and communicates with the i^{th} master PLC. The compressor is controlled through a single PLC. This PLC collects the measurement of P_C, regulates P_C through ω_k, and communicates with CC.

The vulnerabilities within the system $\mathcal{V}$ are presented in Table 5.1, and the available security measures $\mathcal{M}$ in Table 5.2. These sets are modeled based on the lists of common vulnerabilities and countermeasures from [15]. We now briefly introduce the vulnerabilities and the security measures.

Firstly, CC is connected to other networks without adequate protection. Additionally, physical ports on the computers in CC are not secured. These vulnerabilities enable the attacker to gain 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 the physical ports.

Secondly, it was identified that the communication links in the system are unsecured. This allows the attacker to intercept the communication and conduct man-in-the-middle attacks. The sensor measurements and control actions that the attacker is able to manipulate are dependent on the link that is compromised (see Table 5.1, Rows 3 and 5). The vulnerabilities of this type can be prevented by implementing encryption and authentication schemes.

Finally, lack of physical protection of control devices is identified. Sensors and actuators that the attacker gains control over are dependent on which device is compromised (see Table 5.1, Rows 4 and 6). Unauthorized access can be prevented by improving security of an area where components are located, or by protecting components individually by locking them in secured cabinets.

5.4.2 Example 1: Impact and likelihood functions

We now provide concrete examples of the impact and likelihood functions. To form the impact function I, we use the impact estimation framework from Chapter 4. Firstly, observe that the only matrices in the system equations (4.1)–(4.5) that are dependent on a scenario V are Γ_u, Γ_y, Λ_u, and Λ_y. These matrices are formed based on the set of attacked sensors Y_V, the set of attacked actuators U_V, and an attack strategy. We focus on the optimal FDI attack strategy, so Γ_u and Γ_y are given by (4.16), and Λ_u and Λ_y are the identity matrices. Secondly, we need to choose the horizon length N, the stealthiness level ϵ, the matrix C_z, and the impact metric I. We use $N = 20$, $\epsilon = 0.1$, $C_z = [0_{1\times 15}\ \ 0.002]$, and $I = I_P$. Finally, we compute the impact $I(U_V, Y_V)$ by solving Problem 4.1.

The impact function formed in this way satisfies Assumption 5.1. Namely, say that $I(U_1, Y_1) = I_1$. The attacker can then make the impact larger than or equal to I_1 with the components U_2 and Y_2, where $U_2 \supseteq U_1$ and $Y_2 \supseteq Y_1$. For example, if

Table 5.1: The vulnerabilities identified within the system. Each row contains the description of a vulnerability v, the sensors Y_v and actuators U_v that the attacker gains control over by exploiting v, and the complexity π_v of exploiting v.

Vulnerability	Compromised sensors and actuators	π_v
CC connected to other networks without appropriate protection	All the sensors and actuators	3
Insecure physical ports in CC	All the sensors and actuators	3
An insecure comm. link between a PLC and a sensor y_i/an actuator u_i	y_i / u_i	2
Insufficient physical protection of a sensor y_i/an actuator u_i	y_i / u_i	2
An insecure comm. link between: (i) a slave PLC and its master PLC (ii) a master PLC and CC (iii) the compressor PLC and CC	The sensors and actuators attached to: (i) the slave PLC (ii) the master and its slave PLC (iii) the compressor PLC	1
Insufficient physical protection of: (i) a slave PLC (ii) a master PLC (iii) the compressor PLC	The sensors and actuators attached to: (i) the slave PLC (ii) the master and its slave PLC (iii) the compressor PLC	1

Table 5.2: The security measures and vulnerabilities prevented by these measures.

Security measure	Prevented vulnerabilities
Separating CC from other networks using firewalls	The attacker cannot access CC from other networks
Locking the physical ports in CC	The attacker cannot inject malware through the physical ports
Protecting a communication link	The attacker cannot intercept and modify the messages going through the link
Physical protection of a sensor y_i/ an actuator u_i/a PLC	The attacker cannot access y_i/u_i/the PLC
Physical protection of an area where a PLC is located	The attacker cannot access the PLC and all the sensors and actuators attached to it, and cannot exploit unprotected comm. links between these control components

he/she injects the same attack signals to U_1 and Y_1, while setting the attack signals corresponding to $U_2 \setminus U_1$ and $Y_2 \setminus Y_1$ to zero, then the impact equals I_1.

We form the likelihood function based on [32]. The first step is to assign complexity of exploitation $\pi_v \in \mathbb{R}^+$ to every vulnerability $v \in \mathcal{V}$. As mentioned earlier, this can be done based on expert knowledge [17, 32, 33], or by using vulnerability ranking tools [34,35]. We assume the complexities from Table 5.1. In the second step, these

complexities π_v are combined to estimate the likelihood of a scenario. Under the assumption that the scenarios that are more complex to conduct are less likely to occur, a possible choice for the likelihood function is

$$\pi(V) = \left(\sum_{v \in V} \pi_v \right)^{-1}. \tag{5.7}$$

This function makes scenarios containing vulnerabilities with higher complexity values π_v less likely than those containing equal number of vulnerabilities with lower values of π_v. This likelihood function is also decreasing with respect to V, so it satisfies Assumption 5.2.

5.4.3 Example 2: Constructing Problem 5.1

This example is on constructing the security measure allocation problem. We focus on the following five vulnerability sets:

(i) vulnerabilities related to Compressor, CC, and Area 1 (22 vulnerabilities);

(ii) vulnerabilities related to Compressor, CC, and Areas 1–2 (36 vulnerabilities);

(iii) vulnerabilities related to Compressor, CC, and Areas 1–3 (50 vulnerabilities);

(iv) vulnerabilities related to Compressor, CC, and Areas 1–4 (64 vulnerabilities);

(v) vulnerabilities related to Compressor, CC, and Areas 1–5 (78 vulnerabilities).

We set the impact threshold $I_{\min} = 9/10$, and consider three values for the likelihood threshold $\pi_{\min}$: 1/4, 1/6, and 1/8. Observe that the minimum value of π_v is one (see Table 5.1). It then follows from (5.7) that all the scenarios consisting of five or more vulnerabilities have the likelihood lower than the threshold $\pi_{\max} = 1/4$. This implies that the critical scenarios belong to the first four layers of the power set enumeration tree if $\pi_{\max} = 1/4$. Similarly, the critical scenarios belong to the first six layers if $\pi_{\min} = 1/6$, and to the first eight layers if $\pi_{\min} = 1/8$.

The execution time of Algorithm 5.1 with respect to the number of vulnerabilities n_v and the threshold $\pi_{\min}$ is shown in Figure 5.3 (a). Observe that the execution time increases when the number of vulnerabilities n_v increases, or when the threshold $\pi_{\min}$ decreases. Nevertheless, Algorithm 5.1 manages to find the sufficient representation of minimum cardinality in all the cases we consider. The highest execution time of Algorithm 5.1 is 50.36 minutes, and it is reached for the vulnerability set consisting of 78 vulnerabilities and $\pi_{\min} = 1/8$.

For the sake of comparison, let us fix $\pi_{\min} = 1/8$ and measure the time needed to brute force search through the first two layers of the power set enumeration tree.

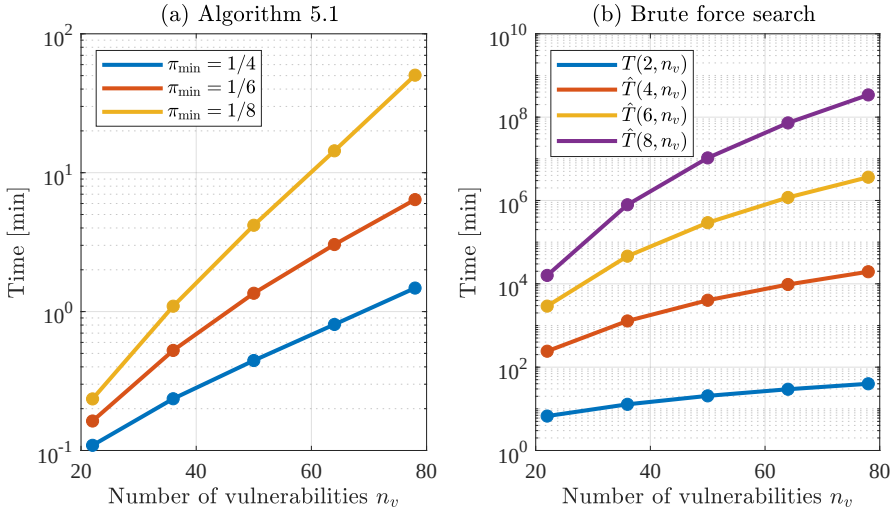

Figure 5.3: (a) The execution time of Algorithm 5.1 with respect to the number of vulnerabilities n_v. Three different values of $\pi_{\min}$ are considered. (b) The time needed to brute force search through the first two layers of the power set enumeration tree, and the estimated times of the brute force search through the first four, six, and eight layers of the tree. We assume $\pi_{\min} = 1/8$ in this experiment.

Based on this time, we estimate the time needed to brute force search through the other layers of the tree according to the following formula:

$$\hat{T}(L, n_v) = \frac{\sum_{i=1}^{L} \binom{n_v}{i}}{\sum_{i=1}^{2} \binom{n_v}{i}} T(2, n_v).$$

Here, $T(2, n_v)$ is the time needed to brute force search through the first two layers of the tree when the vulnerability set contains n_v vulnerabilities, and $\hat{T}(L, n_v)$ is the estimated time to brute force search through the first L layers of the tree corresponding to the same vulnerability set. The measured time $T(2, n_v)$ and the estimates $\hat{T}(4, n_v), \hat{T}(6, n_v), \hat{T}(8, n_v)$ are plotted in Figure 5.3 (b). Observe that $T(2, 78)$ is approximately 40 minutes. This is only 10 minutes less than the time it took Algorithm 5.1 to systematically search through the first eight layers in the case $n_v = 78$. It can also be seen that $\hat{T}(4, 78)$ is greater than 10^4 minutes (6.9 days), $\hat{T}(6, 78)$ is greater than 10^6 minutes (1.9 years), while $\hat{T}(8, 78)$ is greater than 10^8 minutes (190 years).

Overall, this experiment demonstrates that Algorithm 5.1 may indeed allow us to construct the security measure allocation problem in cases when the brute force search is prohibitively time consuming. However, we also see that the execution time of Algorithm 5.1 rapidly increases with the number of vulnerabilities n_v. This

Table 5.3: The number of decision variables and the number of constraints of Problem 5.1 for five vulnerability sets that we consider.

No. of vulnerabilities	No. of decision variables	No. of constraints
22	25	105
36	41	191
50	57	277
64	73	363
78	89	1473

indicates that constructing the security measure allocation problem for the vulnerability sets containing several hundred vulnerabilities may become overly time consuming. If that is the case, then one can try to decrease the execution time of Algorithm 5.1 by using paralelization, or by restricting the search to the first several layers of the power set enumeration tree.

5.4.4 Example 3: Solving Problem 5.1

This example considers solving the security measure allocation problem. We focus on the case $\pi_{\min} = 1/8$, and construct Problem 5.1 using the sufficient representations of minimum cardinality that we computed in the previous example. The number of the decision variables and the constraints of Problem 5.1 for five vulnerability sets that we consider is shown in Table 5.3. The security measure costs are randomly generated 100000 times from the interval [0,10000].

To tackle Problem 5.1, we use the integer linear program solver included in the Matlab package and Algorithm 3.1. First, we compare the execution times of the solver and Algorithm 3.1. The worst case and the mean execution time over all the realizations of costs are plotted in Figure 5.4. Notice that both the solver and Algorithm 3.1 converge in less that 0.1 second in all the cases we consider. Particularly, the longest execution time was 49.86 milliseconds for the solver, and 19.71 milliseconds for Algorithm 3.1. Additionally, Algorithm 3.1 is considerably faster than the solver, both in terms of the worst case and the mean execution time.

Next, we compare the objective value c_G returned by Algorithm 3.1 with the objective value c_I returned by the solver. In Figure 5.5, we plot the largest value of the quotient c_G/c_I, the mean value of the quotient c_G/c_I, and the theoretical worst case bound from Theorem 5.2. We stress that the solver manages to compute the optimal objective value of the problem for all five vulnerability sets and for all realizations of costs. Interestingly, c_G was relatively close to c_I in the average. In the worst case, c_G was 3.4 times larger than c_I. We also observe that the bound from Theorem 5.2 is considerably larger than the largest quotient for all five vulnerability sets. This illustrates that this bound can be conservative.

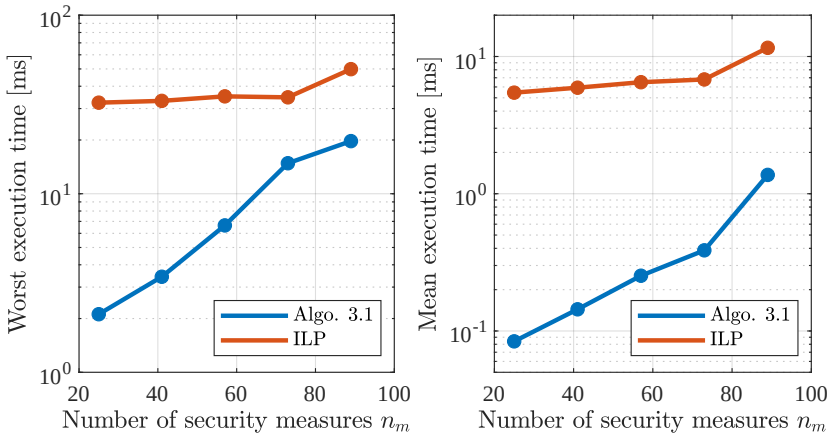

Figure 5.4: Comparison of Algorithm 3.1 and the integer linear program solver in terms of the execution time.

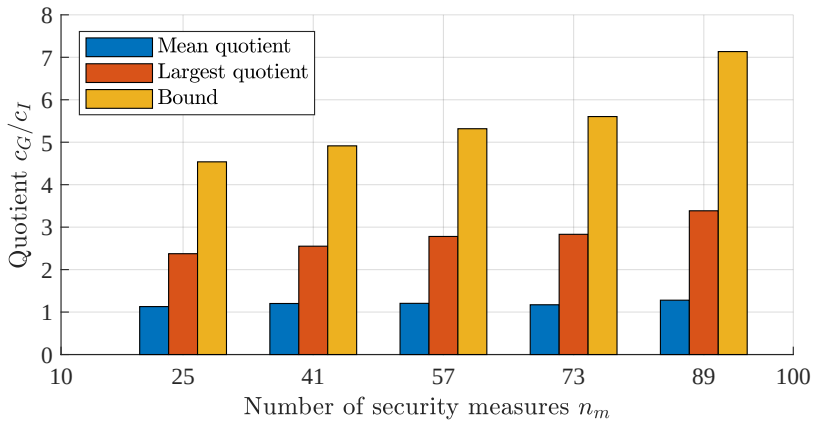

Figure 5.5: Comparison of Algorithm 3.1 and the integer linear program solver in terms of the objective values obtained.

To summarize, these findings demonstrate that although Problem 5.1 is generally NP-hard, the integer linear program solver can sometimes compute a solution of this problem in less than a second. Furthermore, Algorithm 3.1 may return a suboptimal solution close to the optimal one. Algorithm 3.1 also proves to be significantly faster than the integer linear solver. This indicates that we can rely on this algorithm if the use of the solver becomes prohibitively time consuming.

5.5 Summary

This chapter tackled the security measure allocation problem (Problem 5.1). We first derived Algorithm 5.1 that uses several systematic search tools to constructs Problem 5.1. We then showed that Problem 5.1 is NP-hard, and discussed two suboptimal approaches for solving it. The first approach was to first simplify the problem, and then use an integer linear program solver to compute a solution. The second approach was to use Algorithm 3.1 that works in polynomial time and returns a suboptimal solution with performance guarantees. Finally, we conducted an experiment, which demonstrated that: (i) the impact estimation framework from Chapter 4 can be used for security measure allocation; (ii) Algorithm 5.1 can construct Problem 5.1 when the brute force search is prohibitively time consuming; and (iii) a solution of Problem 5.1 can sometimes be efficiently computed using an integer linear program solver, or well approximated using Algorithm 3.1. We now move to Chapter 6, where we study the second motivating application of the thesis.

Appendix to Chapter 5

5.A Proof of Lemma 5.2

($\Rightarrow$) The proof is by contradiction. Assume that $\hat{\mathcal{C}}^*$ is the sufficient representation of minimum cardinality, but there exist two scenarios $V, V' \in \hat{\mathcal{C}}^*$ for which $V \subset V'$ holds. Let us define $\hat{\mathcal{C}} = \hat{\mathcal{C}}^* \setminus V'$, and let V_M be an arbitrary subset of vulnerabilities that has a nonempty intersection with every scenario from $\hat{\mathcal{C}}$. Since $\hat{\mathcal{C}}^* = \hat{\mathcal{C}} \cup V'$, it follows that V_M has a nonempty intersection with every scenario from $\hat{\mathcal{C}}^*$, except perhaps V'. However, since $V \cap V_M \neq \emptyset$ and $V \subset V'$, then $V' \cap V_M \neq \emptyset$ has to hold. Hence, any V_M that intersects every scenario of $\hat{\mathcal{C}}$ also intersects every scenario of $\hat{\mathcal{C}}^*$. Additionally, since $\hat{\mathcal{C}}^*$ is the sufficient representation of minimum cardinality, any V_M that intersects all the scenarios from $\hat{\mathcal{C}}$ also intersects all the scenarios from $\mathcal{C}$. From the latter, the fact that $\hat{\mathcal{C}} \subseteq \mathcal{C}$ holds, and Definition 5.2, it follows that $\hat{\mathcal{C}}$ is a sufficient representation of $\mathcal{C}$. Furthermore, we also have $|\hat{\mathcal{C}}| = |\hat{\mathcal{C}}^* \setminus V'| = |\hat{\mathcal{C}}^*| - 1$. This implies that $\hat{\mathcal{C}}^*$ is not the sufficient representation of minimum cardinality, which contradicts the initial assumption.

($\Leftarrow$) Let $\hat{\mathcal{C}}^*$ be a sufficient representation for which $V \not\subseteq V'$ holds for any two scenarios $V, V' \in \hat{\mathcal{C}}^*$. In what follows, we prove that every scenario that belongs to $\hat{\mathcal{C}}^*$ has to belong to any other sufficient representation. From this fact, it directly follows that $\hat{\mathcal{C}}^*$ is the sufficient representation of minimum cardinality.

Let V_a be an arbitrarily selected scenario from $\hat{\mathcal{C}}^*$, and $\hat{\mathcal{C}}$ be any other sufficient representation of $\mathcal{C}$. Three cases can occur.

Case 1: $V \setminus V_a \neq \emptyset$ holds for every $V \in \hat{\mathcal{C}}$. Let V_M be formed as follows: For every scenario $V \in \hat{\mathcal{C}}$, we add to V_M a vulnerability $v \in V \setminus V_a$. The set V_M constructed in this way intersects all the scenarios from $\hat{\mathcal{C}}$. However, V_M does not intersect V_a, which implies that V_M does not intersect all the scenarios from $\hat{\mathcal{C}}^*$. This is inconsistent with the fact that both $\hat{\mathcal{C}}^*$ and $\hat{\mathcal{C}}$ are sufficient representations of $\mathcal{C}$. Namely, any V_M that intersects all the scenarios from $\hat{\mathcal{C}}$ also intersects all the scenarios from $\mathcal{C}$. Since $\hat{\mathcal{C}}^* \subseteq \mathcal{C}$, all the scenarios from $\hat{\mathcal{C}}^*$ have to be intersected by V_M. Thus, Case 1 is impossible.

Case 2: there exists a scenario $V_b \in \hat{\mathcal{C}}$ for which $V_b \subset V_a$ holds, and $V_a \notin \hat{\mathcal{C}}$. Since $V \not\subseteq V_a$ holds for every $V \in \hat{\mathcal{C}}^* \setminus V_a$, then $V \setminus V_b \neq \emptyset$. Let us define V_M as follows: For every $V \in \hat{\mathcal{C}}^*$, we add to V_M a vulnerability $v \in V \setminus V_b$. By construction, the set V_M intersects all the scenarios from $\hat{\mathcal{C}}^*$. However, V_M does not intersect all the scenarios from $\hat{\mathcal{C}}$, since V_b is not intersected. Thus, Case 2 is also impossible.

Case 3: $V_a \in \hat{\mathcal{C}}$. The only remaining possibility is to have $V_a \in \hat{\mathcal{C}}$. Since V_a was arbitrarily selected, the proof is completed.

5.B Proof of Theorem 5.1

We first prove by induction that $\tilde{\mathcal{C}}^*$ contains $\hat{\mathcal{C}}^*$ (Claim 1). We then show that $\tilde{\mathcal{C}}^*$ is a sufficient representation of $\mathcal{C}$ (Claim 2). Finally, we establish that $\tilde{\mathcal{C}}^*$ is the sufficient representation of minimum cardinality (Claim 3).

Claim 1. Let $p = 1$. Since the list $\mathcal{C}_L$ is initialized with the scenarios from the first layer of the power set enumeration tree, all of the critical scenarios among these are added to $\tilde{\mathcal{C}}^*$. Hence, the claim holds for the first layer. Suppose now that the algorithm reaches the p^{th} layer with a given set $\tilde{\mathcal{C}}^*$ that contains all the scenarios from $\hat{\mathcal{C}}^*$ with the cardinality up to p. In the $(p+1)^{\text{th}}$ layer, all the critical scenarios among the generated ones are added to $\tilde{\mathcal{C}}^*$. The scenarios that are not generated either contain the scenarios that have likelihood lower than $\pi_{\min}$, or a critical scenario added to $\tilde{\mathcal{C}}^*$. From Lemmas 5.1 and 5.2, we know that these scenarios do not belong to $\hat{\mathcal{C}}^*$. Hence, the claim holds for the $(p+1)^{\text{th}}$ layer as well. Therefore, we conclude that $\tilde{\mathcal{C}}^*$ contains $\hat{\mathcal{C}}^*$.

Claim 2. Firstly, since $\tilde{\mathcal{C}}^*$ contains only critical scenarios, we have $\tilde{\mathcal{C}}^* \subseteq \mathcal{C}$. Secondly, since $\tilde{\mathcal{C}}^*$ contains $\hat{\mathcal{C}}^*$, by preventing all the scenarios from $\tilde{\mathcal{C}}^*$, we also prevent all the scenarios from $\hat{\mathcal{C}}^*$. Since $\hat{\mathcal{C}}^*$ is the sufficient representation of minimum car-

dinality, it follows that by preventing all the scenarios from $\tilde{\mathcal{C}}^*$, we also prevent all the scenarios from $\mathcal{C}$. Hence, $\tilde{\mathcal{C}}^*$ is a sufficient representation of $\mathcal{C}$ (Definition 5.2).

Claim 3. To show that $\tilde{\mathcal{C}}^*$ is the sufficient representation of minimum cardinality, we need to prove that $V \not\subseteq V'$ holds for any two scenarios $V, V' \in \tilde{\mathcal{C}}^*$ (Lemma 5.2). If V and V' belong to the same layer, then the condition $V \subseteq V'$ cannot hold. If V and V' belong to different layers, then the condition $V \subseteq V'$ cannot hold either. Namely, when we generate scenarios to be explored in the p^{th} layer, we check if these scenarios contain any of the scenarios previously added to $\tilde{\mathcal{C}}^*$ (Line 26 of Algorithm 5.1). Hence, we conclude that $\tilde{\mathcal{C}}^* = \hat{\mathcal{C}}^*$ holds.

5.C Proof of Corollary 5.1

Let π be an arbitrary selected likelihood function. Consider the function

$$\pi'(V) = \pi(V) \cdot \mathbb{1}_{[|V| \leq n]} + \min\{\pi(V), \pi'_{\min}\} \cdot \mathbb{1}_{[|V| > n]}. \tag{5.8}$$

where $\pi'_{\min} < \pi_{\min}$. If $|V| \leq n$ and $V' \supseteq V$, then

$$\pi'(V) = \pi(V) \overset{\text{Asm. 5.2}}{\geq} \pi(V') \overset{(5.8)}{\geq} \pi'(V'). \tag{5.9}$$

If $|V| > n$ and $V' \supseteq V$, then

$$\pi'(V) = \min\{\pi(V), \pi'_{\min}\} \overset{\text{Asm. 5.2}}{\geq} \min\{\pi(V'), \pi'_{\min}\} \overset{|V'|>n,}{=} \overset{(5.8)}{\pi'(V')}. \tag{5.10}$$

From (5.9) and (5.10), we conclude that π' satisfies Assumption 5.2. Hence, it represents a candidate for the likelihood function.

Note that the set $\mathcal{C}_n$ containing the critical scenarios with cardinalities lower than or equal to n is the same for both π' and π. However, any scenario with a cardinality greater than n is with the likelihood lower than $\pi_{\min}$ when π' is used instead of π. Hence, any scenario with cardinality greater than n is not critical (Lemma 5.1). This implies that the set of critical scenarios is equal to $\mathcal{C}_n$ when π' is used instead of π. Therefore, if we apply Algorithm 5.1 with π' used as a likelihood function, then it follows from Theorem 5.1 that the sufficient representation of minimum cardinality for the set $\mathcal{C}_n$ is returned. The same holds if we stop Algorithm 5.1 after n layers, since the critical scenarios from $\mathcal{C}_n$ lie in the first n layers.

5.D Proof of Theorem 5.2

To prove the claim, we show that F is submodular, nondecreasing, and integer valued. This implies that the problem (5.2) is an instance of Problem 3.1. The claim of the theorem then directly follows from Lemma 3.3.

Claim 1: F is submodular. It suffices to show that f_V is submodular, since submodularity is preserved under a nonnegative sum (Lemma 3.1). Let

$$\Delta_m(M) = f_V(M \cup m) - f_V(M)$$

be the gain achieved by adding a measure m to a set of measures M. We show that $\Delta_m(M)$ can take only values zero or one. The following situations can occur:

(i) the scenario V is prevented by the security measures M, in which case $f_V(M) = 1$ and $f_V(M \cup m) = 1$;

(ii) the scenario V is not prevented by the security measures $M \cup m$, in which case $f_V(M) = 0$ and $f_V(M \cup m) = 0$;

(iii) the scenario V is prevented by the security measure m and not by M, in which case $f_V(M) = 0$ and $f_V(M \cup m) = 1$.

Note that $f_V(M) = 1$ and $f_V(M \cup m) = 0$ cannot hold, since $V_M \subseteq V_{M \cup m}$(from Equation (5.1)). In summary, $\Delta_m(M)$ can be written as follows:

$$\Delta_m(M) = \mathbb{1}_{[V_M \cap V = \emptyset]} \cdot \mathbb{1}_{[V_m \cap V \neq \emptyset]}. \tag{5.11}$$

We now prove by contradiction that f_V is submodular. If f_V is not submodular, then there exist M, $M' \supseteq M$, and $m \in \mathcal{M} \setminus M'$ such that $\Delta_m(M) < \Delta_m(M')$. That is only possible if $\Delta_m(M) = 0$ and $\Delta_m(M') = 1$. From (5.11) and $\Delta_m(M') = 1$, it follows that $V_{M'} \cap V = \emptyset$ and $V_m \cap V \neq \emptyset$. Yet, since $V_M \subseteq V_{M'}$, we have $V_M \cap V = \emptyset$. From the latter, $V_m \cap V \neq \emptyset$, and (5.11), we conclude that $\Delta_m(M) = 1$. Therefore, we have $\Delta_m(M) = \Delta_m(M')$, which leads to a contradiction.

Claim 2: F is nondecreasing. Let $M \subseteq M'$. From (5.1), we have $V_M \subseteq V_{M'}$. This implies that $|V_M \cap V| \leq |V_{M'} \cap V|$ holds for every $V \in \mathcal{C}$. Then it directly follows from the definition of f_V that $f_V(M) \leq f_V(M')$ holds for every $V \in \mathcal{C}$. From the latter, it follows that F is a nonnegative sum of nondecreasing set functions. Thus, we conclude that F is a nondecreasing set function.

Claim 3: F is integer valued. For every $V \in \mathcal{C}$, f_V can take only values zero or one. Since $F(M) = \sum_{V \in \mathcal{C}} f_V(M)$, it follows that F is an integer valued function.

5.E Proof of Proposition 5.2.

In the proof of Lemma 5.2, we showed that the sufficient representation of minimum cardinality belongs to any other sufficient representation. Thus, we can write $\hat{\mathcal{C}} = \hat{\mathcal{C}}^* \cup (\hat{\mathcal{C}} \setminus \hat{\mathcal{C}}^*)$, where $\hat{\mathcal{C}} \setminus \hat{\mathcal{C}}^*$ is a nonempty set.

Let $m \in \mathcal{M}$ be an arbitrarily selected security measure. It then follows that

$$\hat{F}(m) = \sum_{V \in \hat{\mathcal{C}}^*} f_V(M) + \sum_{V \in \hat{\mathcal{C}} \setminus \hat{\mathcal{C}}^*} f_V(M) = \hat{F}^*(m) + \sum_{V \in \hat{\mathcal{C}} \setminus \hat{\mathcal{C}}^*} f_V(M) \overset{(*)}{\geq} \hat{F}^*(m),$$

where (*) holds because f_V is a nonnegative function for any $V \in \mathcal{C}$. Next, since $\hat{F}^*(m) \leq \hat{F}(m)$ for any m, we have

$$n^* = \max_{m \in \mathcal{M}} \hat{F}^*(m) \leq \max_{m \in \mathcal{M}} \hat{F}(m) = \hat{n}.$$

It then follows that

$$H(\max_{m \in \mathcal{M}} \hat{F}^*(m)) = \sum_{i=1}^{n^*} \frac{1}{i} \leq \sum_{i=1}^{n^*} \frac{1}{i} + \sum_{i=n^*+1}^{\hat{n}} \frac{1}{i} = H(\max_{m \in \mathcal{M}} \hat{F}(m))$$

has to hold, which completes the proof.

5.F Numerical values of the matrices used in simulations

$$A_1 = \begin{bmatrix} 0.9998 & 0.0002 & 0 \\ 0.0119 & 0.9788 & 0.0002 \\ 0 & 0 & 0.9030 \end{bmatrix} \qquad A_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -0.0154 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 0 \\ 0 \\ -0.0665 \end{bmatrix} \qquad A_4 = \begin{bmatrix} 0 & 0 & -0.0045 \end{bmatrix}$$

$$A_5 = 0.9758$$

$$B_1 = \begin{bmatrix} 0 & 0 \\ 0.0001 & -0.0015 \\ 0 & 0.1029 \end{bmatrix} \qquad B_2 = 10^{-5} \begin{bmatrix} 0 & 0 \\ 0 & -0.0001 \\ 0 & -0.2653 \end{bmatrix}$$

$$B_3 = 10^{-4} \begin{bmatrix} 0 \\ -0.0001 \\ -0.8509 \end{bmatrix} \qquad B_4 = \begin{bmatrix} 0 & -0.0250 \end{bmatrix}$$

$$B_5 = 0.0024$$

$$L_{11} = \begin{bmatrix} -204.9676 & 84.0710 & 1.0988 \\ -1.1400 & -0.2779 & 8.5412 \end{bmatrix} \qquad L_{12} = \begin{bmatrix} 1.8443 & -0.0319 & -0.0492 \\ -0.0554 & -0.0087 & -0.3643 \end{bmatrix}$$

$$L_{13} = \begin{bmatrix} -0.2158 \\ -1.6118 \end{bmatrix} \qquad L_{14} = \begin{bmatrix} -30.9138 & -5.4842 & 24.2483 \end{bmatrix}$$

$$L_{15} = 108.9814 \qquad L_{31} = \begin{bmatrix} 0 \\ -166.2425 \end{bmatrix}$$

$$L_{32} = \begin{bmatrix} 0 \\ -11.7937 \end{bmatrix} \qquad L_{33} = 0$$

Chapter 6

Actuator security indices

This chapter introduces the security indices δ and δ_r, which are used to characterize vulnerable actuators in control systems. The index $\delta(u_i)$ is defined for every actuator u_i, and it is equal to the minimum number of sensors and actuators that need to be compromised by an attacker to conduct a perfectly undetectable attack against u_i. Since perfectly undetectable attacks do not leave traces in the sensor measurements, an actuator with a small value of δ is very vulnerable. We propose a method to compute δ in small-scale systems and show that δ may be increased by placing additional sensors, or decreased by placing additional actuators. We then identify three issues that appear in large-scale systems. Namely, δ is NP-hard to compute, sensitive to system variations that are expected in large-scale systems, and based on the assumption that the attacker knows the entire system model, which can be a conservative assumption in the case of large-scale systems.

We then introduce the robust security index δ_r, which can characterize actuators vulnerable in any realization of the system. We show that this index can be computed efficiently and related to both full and limited model knowledge attackers. Since our results imply that actuators with a small value of δ_r are very vulnerable, we investigate how to increase δ_r. We show that δ_r is guaranteed to increase if sensors are placed at suitable locations in the system, and then formulate a sensor allocation problem with the objective to increase δ_r. It turns out that this problem has a submodular structure similar to the security measure allocation problem from Chapter 5. This enables us to find its suboptimal solution with guaranteed performance efficiently. We also illustrate the theoretical results through examples.

The chapter is organized as follows. Section 6.1 introduces the security index δ. Section 6.2 investigates properties of δ. Section 6.3 defines the robust security index δ_r. Section 6.4 outlines properties of δ_r. Section 6.5 presents illustrative examples. Section 6.6 concludes the chapter. The appendix contains lengthy proofs.

6.1 The security index δ

The system model we use to define the security index δ is given by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_a a(k), \\ y(k) &= Cx(k) + D_a a(k), \end{aligned} \tag{6.1}$$

where $x(k) \in \mathbb{R}^{n_x}$ are the system states, $u(k) \in \mathbb{R}^{n_u}$ are the control actions, $y(k) \in \mathbb{R}^{n_y+n_e}$ are the sensor measurements, and $a(k) \in \mathbb{R}^{n_u+n_y}$ are the attacks. We allow the last $n_e \geq 0$ elements of y to be protected, so the attacker cannot directly manipulate them. We also assume that the attacker cannot directly manipulate the non-attacked sensors and actuators, so the elements of a that correspond to these components always equal zero. For the analysis that follows, it is convenient to assume that $x(0) = 0_{n_x}$ and $u \equiv 0$. Due to linearity, this assumption is without loss of generality for most results in the chapter. The exceptions are clearly outlined.

We denote by $\mathcal{X} = \{x_1, \ldots, x_{n_x}\}$ the set of states, $\mathcal{U} = \{u_1, \ldots, u_{n_u}\}$ the set of actuators, $\mathcal{Y} = \{y_1, \ldots, y_{n_y+n_e}\}$ the set of sensors, and $\mathcal{I} = \{1, \ldots, n_u + n_y\}$ the attack vector indices. The first n_u elements of a correspond to attacks against the actuators, and the last n_y to attacks against unprotected sensors. Thus, we have

$$B_a = \begin{bmatrix} B & 0_{n_x \times n_y} \end{bmatrix}, \qquad D_a = \begin{bmatrix} 0_{n_y \times n_u} & I_{n_y} \\ 0_{n_e \times n_u} & 0_{n_e \times n_y} \end{bmatrix}.$$

We also assume that B has a full column rank. This excludes the degenerate cases where the attacks trivially cancel each-other, or cases where an actuator does not affect the system. We adopt the following attacker model.

Assumption 6.1. *The attacker can change the values of attacked control actions and measurements arbitrarily, and knows the matrices A, B, C.*

Next, we assume that the attacker seeks to conduct a perfectly undetectable attack [37, 136]. Perfectly undetectable attacks are potentially very dangerous, since they do not leave traces in the sensor measurements.

Definition 6.1. *An attack $a \not\equiv 0$ is perfectly undetectable if $y \equiv 0$.*

We are now ready to introduce the security index δ. The security index $\delta(u_i)$ is defined for every actuator $u_i \in \mathcal{U}$, and it is equal to the minimum number of sensors and actuators that need to be compromised by the attacker to conduct a perfectly undetectable attack. Additionally, the actuator u_i has to be actively used in the attack. This models a goal or intent by the attacker. Hence, the problem of computing $\delta(u_i)$ can be written as follows:

Problem 6.1. *Computing the security index $\delta(u_i)$:*

$$\begin{aligned}
&\underset{a}{\text{minimize}} \quad \|a\|_0 \\
&\text{subject to} \quad x(k+1) = Ax(k) + B_a a(k), && \text{(C1)} \\
&\qquad\qquad y(k) = Cx(k) + D_a a(k), && \text{(C2)} \\
&\qquad\qquad y \equiv 0,\ x(0) = 0_{n_x}, && \text{(C3)} \\
&\qquad\qquad a_i \not\equiv 0. && \text{(C4)}
\end{aligned}$$

The objective function reflects the attacker's desire to find the minimum number of sensors and actuators to conduct a perfectly undetectable attack (sparsest signal a). The constraints (C1) and (C2) ensure that the attack satisfies the system dynamics, (C3) imposes the attack to be perfectly undetectable, and (C4) ensures that the actuator u_i is actively used in the attack.

Before we start analyzing δ, we outline some remarks.

Remark 6.1. *Actuators with small values of δ are more vulnerable than those with large values. The worst case for the operator occurs when $\delta(u_i) = 1$. This implies that the attacker can attack u_i and stay perfectly undetectable without compromising other components.*

Remark 6.2. *Problem 6.1 is not always feasible. The absence of a solution implies that the attacker cannot attack u_i while staying perfectly undetectable. In this case, we adopt $\delta(u_i) = +\infty$.*

Remark 6.3. *Problem 6.1 can be extended to capture the case where sensors and actuators are not equally hard to attack. This can be achieved by changing the objective to $\sum_{j\in\mathcal{I}} c_j \mathbb{1}_{[a_j \neq 0]}$, where $c_j \in \mathbb{R}^+$ models a cost of attacking a component j.*

6.2 Properties of the security index δ

This section shows how to compute δ, analyzes how the deployment of new sensors and actuators affects δ, and outlines issues that appear in large-scale systems.

6.2.1 Computing the security index δ

In the following proposition, we derive a necessary and sufficient condition that a set of attacked components needs to satisfy, so that an attack signal a feasible for Problem 6.1 can be constructed.

Proposition 6.1. *Let G be the transfer function from a to y, U_a be attacked actuators, Y_a be attacked sensors, and $I_a \subseteq \mathcal{I}$ be the indices of a corresponding to U_a*

and Y_a. A perfectly undetectable attack conducted with the components U_a and Y_a with active use of an actuator $u_i \in U_a$ exists if and only if

$$\text{nrank}[\, G^{(I_a)} \,] = \text{nrank}[\, G^{(I_a \setminus i)} \,]. \tag{6.2}$$

Proof. We refer the reader to Appendix 6.A. ■

There are three important consequences of this result. Firstly, we can use the condition (6.2) to compute $\delta(u_i)$ as follows. We form all the combinations of attacked sensors Y_a and actuators U_a for which $u_i \in U_a$ and $|U_a| + |Y_a| = p$. The initial value of p is set to one. For each combination, we check if (6.2) is satisfied, which can be done efficiently (for example, by using the Matlab function `tzero`). If there exists a combination for which (6.2) holds, then we return $\delta(u_i) = p$. Otherwise, we increse p by one and repeat the process.

Secondly, the proof shows that the attacker can cover an arbitrarily large attack signal injected in u_i when (6.2) holds. This malicious signal can damage the actuator, as shown in the Stuxnet attack [6] and the Aurora experiment [36]. Furthermore, since B has a full column rank, the attack necessarily results in some of the system states becoming large. We also point out that such an attack can be constructed off-line using the model knowledge. This makes the attack decoupled from $x(0)$ and u. Thus, the attack remains perfectly undetectable for any $x(0)$ and u, and the assumption $x(0) = 0_{n_x}$ and $u \equiv 0$ is without loss of generality.

Finally, Proposition 6.1 helps us to avoid checking the infinite number of constraints in Problem 6.1. Instead, it suffices to check if the condition (6.2) is satisfied for a given combination of attacked sensors and actuators.

6.2.2 Increasing and decreasing the security index δ

We now investigate how the deployment of new sensors and actuators affects δ.

Proposition 6.2. *Assume that a new component j is deployed. Let $\delta(u_i)$ be the security index of an actuator u_i before the deployment of j. If $\delta'(u_i)$ is the security index of u_i after the deployment of j, then:*

(i) $\delta(u_i) \leq \delta'(u_i) \leq \delta(u_i) + 1$ *holds when j is an unprotected sensor;*

(ii) $\delta(u_i) \leq \delta'(u_i)$ *holds when j is a protected sensor;*

(iii) $\delta(u_i) \geq \delta'(u_i)$ *holds when j is an actuator.*

Proof. The placement of a new sensor introduces additional constraints to Problem 6.1, which restrict the set of possible solutions. Thus, $\delta'(u_i) < \delta(u_i)$ cannot hold. If the new sensor is unprotected, then the attacker can gain control over it. This can be interpreted as removing the aforementioned constraints. Problem 6.1

then becomes the same as before the placement, so $\delta'(u_i) \leq \delta(u_i)+1$ has to hold. By placing a new actuator, the number of decision variables in Problem 6.1 increases, and the constraints remain unchanged. Thus, $\delta'(u_i) \leq \delta(u_i)$ holds. ■

Proposition 6.2 implies that we can potentially increase δ by placing additional sensors to monitor the system. Furthermore, δ can be used to determine which sensor placement is the most beneficial. For example, one optimality criterion can be to select the placement such that the minimum value of δ is as large as possible. If the system is of sufficiently small scale, and if a small number of sensors is being allocated, then we can test all the sensor placements and pick the best.

Proposition 6.2 also illustrates an interesting trade-off between security and safety. On the one hand, to make the system easier to control and more resilient to actuator faults, more actuators should be placed in the system. On the other hand, this may decrease the security indices, making the actuators easier to attack.

6.2.3 Large-scale control systems and the security index δ

We now outline three issues that appear when a control system is large scale.

Issue 1: The security index δ is NP-hard to compute

We now establish that Problem 6.1 is NP-hard. This implies that known polynomial-time algorithms cannot solve this problem.

Theorem 6.1. *Problem 6.1 is NP-hard.*

Proof. We refer the reader to Appendix 6.B. ■

Remark 6.4. *The proof of Theorem 6.1 shows that Problem 6.1 can sometimes be reduced to a problem with a finite number of constraints. Nevertheless, such a problem is still NP-hard to solve due to the ℓ_0-norm in the objective.*

Issue 2: Fragility of the security index δ

Large-scale control systems are complex systems that can change their configuration over time. For example, in a power grid, micro-grids can detach from the grid [196], some power lines may be turned off [197], or certain measurements may become unavailable due to unreliable communication [198]. Unfortunately, the security index can be sensitive with respect to changes in the realization of the system matrices A, B, C. This is illustrated in the following example

Example 6.1. *Let the realization of the system be*

$$A = \begin{bmatrix} 0.1 & 0 \\ 0.01 & 0.1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}. \tag{6.3}$$

Assume that the sensors measuring x_2 are protected. Then $\delta(u_1) = +\infty$ because any input influences the protected outputs. However, if $A(2,1) = 0$, then the transfer matrix from the actuator to the sensors is zero, in which case $\delta(u_1) = 1$.

The lack of robustness of δ has two consequences. Firstly, an actuator that appears to be secure in one realization of the system may be vulnerable in another. Thus, to find actuators that are vulnerable, one should compute δ for different realizations of A, B, C. Due to NP-hardness, this is infeasible in large-scale systems. Secondly, even if we compute indices for all the realizations, ensuring that δ of every actuator is large enough in every realization may be prohibitively expensive.

A reasonable strategy is therefore to first focus on defending those actuators that are vulnerable in any system realization. However, the question to answer is if we can find these actuators efficiently.

Remark 6.5. *We assume that system variations occur infrequently compared to the time scale of the perfectly undetectable attacks. Hence, to the attacker, the system is linear and time-invariant.*

Issue 3: Full model knowledge attacker

If the system is large-scale, then Assumption 6.1, which imposes that the attacker has the exact knowledge of A,B,C, may be conservative. The lack of the full model knowledge represents a serious disadvantage for the attacker and can lead to his/her detection [85] (see also Section 6.5.6). It is therefore relevant to develop indices that are also valid for attackers limited to local model knowledge.

Summary of Section 6.2.3

Due to the previous three issues, δ is not practical to be used in large-scale control systems. Therefore, we introduce the robust security index δ_r that can be used to characterize actuators that are vulnerable in any system realization, is easy to compute, and can be related to attackers with limited model knowledge.

6.3 The robust security index δ_r

The robust security index we introduce in this section is based on a structural model $[A], [B], [C]$ of the system. The structural matrix $[A] \in \{0,1\}^{n_x \times n_x}$ has binary elements. If $[A](i,j) = 0$, then $A(i,j) = 0$ for every realization A. If $[A](i,j) = 1$, then $A(i,j)$ can take any value from $\mathbb{R}$. The same holds for the matrices $[B] \in \{0,1\}^{n_x \times n_u}$ and $[C] \in \{0,1\}^{(n_y+n_e) \times n_x}$. In the remainder of the chapter, we focus our attention on a special case of the matrix $[B]$.

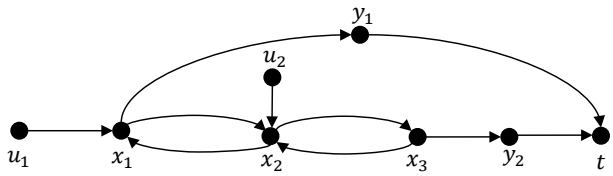

Figure 6.1: The extended graph $\mathcal{G}_t$ corresponding to the structural matrices $[A], [B], [C]$ from Example 6.2.

Assumption 6.2. *We assume that: (i)* $[B] = [e_{i_1} \dots e_{i_{n_u}}]$ *and* $\text{rank}[B] = n_u$*; and (ii)* $[B](i,j) = 1$ *implies that* $B(i,j) \neq 0$ *for every realization of* B*.*

Assumption 6.2 imposes that each actuator directly influences only one state, which is commonly assumed in actuator allocation problems for large-scale systems [199, 200]. Additionally, Assumption 6.2 imposes that every realization of B has a full column rank, which ensures compatibility with the index δ. We remark that Assumption 6.2 is necessary for the derivation of the results that follow. Additionally, even under this assumption, Problem 6.1 remains NP-hard to solve.

Proposition 6.3. *Problem 6.1 remains NP-hard under Assumption 6.2.*

Proof. In the proof of Theorem 6.1, we set $B = I_{n_x}$ to establish NP-hardness of Problem 6.1. Since I_{n_x} is compatible with Assumption 6.2, the claim holds. ■

We now introduce the extended graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E})$ to represent $[A], [B], [C]$. The node set is given by $\mathcal{V} = \mathcal{X} \cup \mathcal{U} \cup \mathcal{Y} \cup t$, where the node t can be seen as a control center that receives the measurements from the process. The edge set is given by $\mathcal{E} = \mathcal{E}_{ux} \cup \mathcal{E}_{xx} \cup \mathcal{E}_{xy} \cup \mathcal{E}_{yt}$, where $\mathcal{E}_{ux} = \{(u_j, x_i) : [B](i,j) = 1\}$ are the edges from the actuators to the states, $\mathcal{E}_{xx} = \{(x_j, x_i) : [A](i,j) = 1\}$ are the edges between the states, $\mathcal{E}_{xy} = \{(x_j, y_i) : [C](i,j) = 1\}$ are the edges from the states to the sensors, and $\mathcal{E}_{yt} = \{(y_i, t) : \forall y_i \in \mathcal{Y}\}$ are the edges from the sensors to t. Since the extended graph $\mathcal{G}_t$ is crucial for analyzing the robust index δ_r that we define in the following, we introduce an example to clarify it.

Example 6.2. *Let the structural matrices be given by*

$$[A] = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad [B] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad [C] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The extended graph $\mathcal{G}_t$ *is shown in Figure 6.1.*

Let $[A], [B], [C]$ be given, and let us define the set $\mathcal{R}$ of system realizations (A, B, C) that have the same structure as $[A], [B], [C]$ and satisfy Assumption 6.2. The robust security index $\delta_r(u_i)$ of an actuator u_i can then be defined as follows:

Problem 6.2. *Computing the robust security index $\delta_r(u_i)$:*

$$\begin{aligned} \underset{I_a \subseteq \mathcal{I}}{\text{minimize}} \quad & |I_a| \\ \text{subject to} \quad & \forall (A,B,C) \in \mathcal{R},\ \exists a: \\ & \operatorname{supp}(a) \subseteq I_a, \\ & x(k+1) = Ax(k) + B_a a(k), \\ & y(k) = Cx(k) + D_a a(k), \\ & y \equiv 0, x(0) = 0_{n_x}, \\ & a_i \not\equiv 0. \end{aligned}$$

Put differently, the robust security index $\delta_r(u_i)$ characterizes the minimum number of sensors and actuators that enables a perfectly undetectable attack against u_i in any system realization $(A,B,C) \in \mathcal{R}$. Thus, a small $\delta_r(u_i)$ indicates a serious vulnerability of the actuator u_i. Particularly, the attacker can not only conduct a perfectly undetectable attack against u_i using a small number of components, but he/she can do that in any system realization from $\mathcal{R}$.

Remark 6.6. *Just as Problem 6.1, Problem 6.2 is not always solvable. This occurs when the attacker cannot gather the resources that allow him/her to conduct a perfectly undetectable attack against u_i in every system realization from $\mathcal{R}$. In that case, we adopt $\delta_r(u_i) = +\infty$.*

Apart from its ability to characterize actuators vulnerable in any system realization, the robust index δ_r has also other favorable properties that we outline next.

6.4 Properties of the robust security index δ_r

This section shows that δ_r can be efficiently computed by solving the minimum s-t cut problem, relates δ_r with different attacker models, and shows how δ_r can be improved through sensor allocation.

6.4.1 Computing the robust security index δ_r

We first introduce Theorem 6.2 that gives a necessary and sufficient condition that a set of attacked components needs to satisfy, so that an attack signal a feasible for Problem 6.1 can be constructed in any system realization $(A,B,C) \in \mathcal{R}$. Theorem 6.2 is inspired by [37], where the connection between the existence of perfectly undetectable attacks and the size of a minimum vertex separator was introduced.

Theorem 6.2. *Let U_a be attacked actuators, Y_a be attacked sensors, and*

$$X_a = \{\, x_j \in \mathcal{X} \,:\, (u_k, x_j) \in \mathcal{E}_{ux},\, u_k \in U_a \setminus u_i \,\}. \tag{6.4}$$

A perfectly undetectable attack conducted with U_a and Y_a with active use of an actuator $u_i \in U_a$ exists in any realization $(A, B, C) \in \mathcal{R}$ if and only if $X_a \cup Y_a$ is a vertex separator of u_i and t in the extended graph $\mathcal{G}_t$.

Proof. We refer the reader to Appendix 6.C ■

The intuition behind Theorem 6.2 is the following. An attack against u_i can be thought of as the attacker injecting a flow into the system through u_i. To stay perfectly undetectable, he/she seeks to prevent the flow from reaching the operator modeled by t. The attacker uses a simple strategy where he/she injects negative flows into the states X_a using the actuators $U_a \setminus u_i$, and in that way, cancels out the flows going through these states. The same applies to Y_a. If $X_a \cup Y_a$ is a vertex separator of u_i and t, then the flow is successfully canceled out, and the attack remains perfectly undetectable. However, if there exists a directed path connecting u_i and t not intersected by $X_a \cup Y_a$, then we can find a realization from $\mathcal{R}$ for which any flow injected in u_i reaches the operator.

From Theorem 6.2, it follows that computing the robust security index $\delta_r(u_i)$ reduces to computing a minimum vertex separator between u_i and t that consists of X_a and Y_a. Hence, Problem 6.2 can be reformulated as follows:

$$\begin{aligned} \underset{U_a, Y_a}{\text{minimize}} \quad & |U_a| + |Y_a| \\ \text{subject to} \quad & X_a \text{ is given by (6.4)}, \\ & Y_a \text{ consists of unprotected sensors}, \\ & X_a \cup Y_a \text{ is a vertex separator of } u_i \text{ and } t, \\ & u_i \in U_a. \end{aligned} \tag{6.5}$$

Here, the objective reflects our goal to find a minimum vertex separator. The first two constraints ensure that the separator consists only of states X_a and unprotected sensors Y_a, the third constraint ensures that $X_a \cup Y_a$ is a vertex separator of u_i and t, and the fourth imposes that u_i is included in the attacked components.

We now show that the problem (6.5) can be reduced to the minimum s-t cut problem (see Section 3.3.2). This implies that the problem (6.5) can be solved in polynomial time using well established algorithms [159]. To prove this claim, we first transform $\mathcal{G}_t$ to a more convenient graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$, with a set of edge weights $\mathcal{W}_i$.

Let a state x_j be of Type 1 if it is adjacent to an actuator from $\mathcal{U} \setminus u_i$, and of Type 2 otherwise. The set $\mathcal{V}_i$ contains u_i (the source), t (the sink), every state of Type 2, and the measurements $\mathcal{Y}$. Additionally, we introduce the nodes $x_{j_{in}}$ and $x_{j_{out}}$ for every state x_j of Type 1, and add them to $\mathcal{V}_i$. The sets $\mathcal{E}_i$ and $\mathcal{W}_i$ are constructed according to the following rules:

(i) If $(u_i, x_j) \in \mathcal{E}_{ux}$, then we add (u_i, x_j) to $\mathcal{E}_i$, and we set $w_{u_i x_j} = +\infty$.

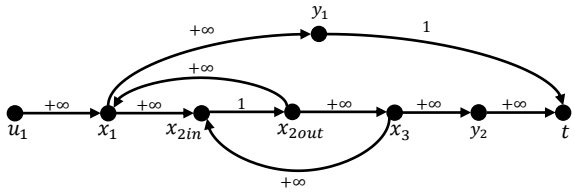

Figure 6.2: The graph $\mathcal{G}_1$ corresponding to the actuator u_1.

(ii) For every $(x_j, x_k) \in \mathcal{E}_{xx}$, $x_j \neq x_k$, we add an edge of the weight $+\infty$ to $\mathcal{E}_i$ subject to the following rules:
- if x_j and x_k are Type 1, then we add $(x_{j_{out}}, x_{k_{in}})$ to $\mathcal{E}_i$;
- if x_j is Type 1 and x_k is Type 2, then we add $(x_{j_{out}}, x_k)$ to $\mathcal{E}_i$;
- if x_j is Type 2 and x_k is Type 1, then we add $(x_j, x_{k_{in}})$ to $\mathcal{E}_i$;
- if x_j and x_k are Type 2, then we add (x_j, x_k) to $\mathcal{E}_i$.

(iii) For every $x_{j_{in}}$ and $x_{j_{out}}$ that correspond to a state x_j of Type 1, we add $(x_{j_{in}}, x_{j_{out}})$ to $\mathcal{E}_i$, and we set $w_{x_{j_{in}} x_{j_{out}}} = 1$.

(iv) For every $(x_j, y_k) \in \mathcal{E}_{xy}$ where x_j is of Type 1, we add $(x_{j_{out}}, y_k)$ to $\mathcal{E}_i$, and we set $w_{x_{j_{out}} y_k} = +\infty$.

(v) For every $(x_j, y_k) \in \mathcal{E}_{xy}$ where x_j is of Type 2, we add (x_j, y_k) to $\mathcal{E}_i$, and we set $w_{x_j y_k} = +\infty$.

(vi) For every $y_j \in \mathcal{Y}$, we add (y_j, t) to $\mathcal{E}_i$. If y_j is a protected sensor, then we set $w_{y_j t} = +\infty$. Otherwise, we set $w_{y_j t} = 1$.

To clarify the graph $\mathcal{G}_i$, we introduce an example.

Example 6.3. *Assume the same structural matrices as in Example 6.2. Let the first sensor be unprotected and the second one protected. The graph $\mathcal{G}_1$ constructed for the purpose of solving the problem* (6.5) *for actuator u_1 is shown in Figure 6.2.*

We now introduce Proposition 6.4, which tells us that we can compute the robust security index $\delta_r(u_i)$ by solving the minimum u_i-t cut problem in $\mathcal{G}_i$.

Proposition 6.4. *Let $\delta_r(u_i)$ be the optimal value of the problem* (6.5)*, and δ^* be the optimal value of the minimum u_i-t cut problem in $\mathcal{G}_i$. If the problem* (6.5) *is feasible, then $\delta_r(u_i) = \delta^* + 1$. Otherwise, $\delta_r(u_i) = \delta^* = +\infty$ holds.*

Proof. We refer the reader to Appendix 6.D. ■

Remark 6.7. *Proposition 6.4 extends the previous findings on the static security index α [129], where α was computed by solving the minimum s-t cut problem.*

6.4.2 The robust security index δ_r and attacker models

We now explain how δ_r is related to the full model knowledge attacker and two limited model knowledge attackers. To distinguish between the different attackers, in the remainder we refer to the full model knowledge attacker as Attacker 1, and to the newly introduced attackers as Attackers 2 and 3.

Attacker 1: The full model knowledge attacker

As mentioned earlier, $\delta_r(u_i)$ characterizes the minimum number of sensors and actuators that enable Attacker 1 to attack u_i and remain perfectly undetectable in any realization from $\mathcal{R}$. Hence, a large (resp. small) $\delta_r(u_i)$ prevents (resp. enables) Attacker 1 to easily gather disruption resources to attack u_i in any system realization. It is also worth mentioning that $\delta_r(u_i)$ upper bounds $\delta(u_i)$ in any system realization. This is established in the following proposition.

Proposition 6.5. *For any $(A, B, C) \in \mathcal{R}$ and $u_i \in \mathcal{U}$, we have $\delta_r(u_i) \geq \delta(u_i)$. If $\delta_r(u_i) = +\infty$, then $\delta(u_i) = +\infty$ for at least one realization $(A, B, C) \in \mathcal{R}$.*

Proof. Case $\delta_r(u_i) \neq +\infty$: Let (U_a, Y_a) be a solution of the problem (6.5). From Theorem 6.2, we know that the attacker can conduct a perfectly undetectable attack against u_i in any system realization from $\mathcal{R}$ using U_a and Y_a. Therefore, for any $(A, B, C) \in \mathcal{R}$, we can find an attack a with the following properties: (i) $\|a\|_0 = |U_a| + |Y_a|$; and (ii) a is a feasible point of Problem 6.1. From the latter, we conclude that $\delta(u_i) \leq |U_a| + |Y_a| = \delta_r(u_i)$ has to hold in any realization $(A, B, C) \in \mathcal{R}$.

Case $\delta_r(u_i) = +\infty$: The proof is by contradiction. Assume that $\delta(u_i) \neq +\infty$ for any $(A, B, C) \in \mathcal{R}$. Thus, Problem 6.1 is feasible in any system realization. Let $U_a = \mathcal{U}$, and let Y_a be the set of all unprotected sensors. If the attacker can conduct a perfectly undetectable attack against u_i in any system realization with some set of components, then he/she can do it with U_a and Y_a as well. However, this implies that (U_a, Y_a) is a feasible point of the problem (6.5), which is impossible since $\delta_r(u_i) = +\infty$. Hence, $\delta(u_i) = +\infty$ has to hold for at least one $(A, B, C) \in \mathcal{R}$. ■

Unfortunately, Section 6.5 illustrates that $\delta_r(u_i)$ is not a tight upper bound of $\delta(u_i)$. Thus, there generally exist realizations in which fewer than $\delta_r(u_i)$ components suffice for Attacker 1 to conduct a perfectly undetectable attack against u_i. However, Attacker 1 needs to be sure that such a realization is present when he/she decides to attack. If these realizations occur rarely, then the attacker may need to wait for a long time, which increases his/her chances to be discovered. To avoid this, Attacker 1 may still want to compromise $\delta_r(u_i)$ components that allow him/her to conduct a perfectly undetectable against u_i in any system realization from $\mathcal{R}$.

Attacker 2: The attacker with local model knowledge

We now show that a small $\delta_r(u_i)$ implies that u_i is vulnerable even if the attacker does not know the entire realization A, B, C. Consider the following attacker.

Assumption 6.3. *Attacker 2:*

(i) can read and change the values for control and measurements signals that correspond to attacked actuators U_a and attacked sensors Y_a;

(ii) knows $[A], [B], [C]$;

(iii) knows the rows $A(j,:)$ and $B(j,:)$ for every state x_j that is adjacent to an actuator from U_a;

(iv) knows for every $k \in \mathbb{Z}_{\geq 0}$: $x_j(k)$ for any x_j that is adjacent to an actuator from U_a, and $x_l(k)$ for any $x_l \in \mathcal{N}_{x_j}^{in}$;

(v) seeks to remain perfectly undetectable.

Attacker 2 does not know the entire realization A, B, C, but only the structural model and the rows of A and B that correspond to the actuators U_a. Attacker 2 is also assumed to know the values of the states adjacent to the actuators U_a and their in-neighbors. He/she can obtain these values by placing additional sensors, but can also get this information free of cost. Namely, control algorithms sometimes base decision on local and neighboring states to achieve better performance [201]. Hence, the neighboring nodes may continue sending the information to the compromised actuator nodes if the attacker remains undetected.

The following proposition relates Attacker 2 to δ_r.

Proposition 6.6. *Let U_a be attacked actuators, Y_a be attacked sensors, $u_i \in U_a$, and X_a be defined as in* (6.4)*. Attacker 2 can conduct a perfectly undetectable attack with active use of u_i in any realization $(A, B, C) \in \mathcal{R}$ if and only if $X_a \cup Y_a$ is a vertex separator of u_i and t in $\mathcal{G}_t$.*

Proof. We refer the reader to Appendix 6.E. ■

Recall that $\delta_r(u_i)$ equals the minimum number of components that ensures $X_a \cup Y_a$ is a vertex separator of u_i and t, and $u_i \in U_a$. Hence, Proposition 6.6 implies that Attacker 2 with the right combination of $\delta_r(u_i)$ components can conduct a perfectly undetectable attack against u_i in any system realization. Therefore, a small $\delta_r(u_i)$ implies that u_i is vulnerable even if the attacker does not know the full model.

We also point out that the assumption $x(0) = 0_{n_x}$ and $u \equiv 0$ is needed for this result to hold. Particularly, we show in the proof that Attacker 2 can construct an attack similar to the one introduced in the proof of Theorem 6.2. However, to

compensate for the lack of the full model knowledge, Attacker 2 implements the strategy in a feedback manner using local states and measurements, and exploits the steady state assumption. Section 6.5.6 illustrates that if u starts changing during the attack, Attacker 2 can be revealed.

Attacker 3: The attacker limited to structural knowledge

While the previous two propositions show that a small $\delta_r(u_i)$ implies that u_i is vulnerable, a perhaps more interesting question to answer is if a large $\delta_r(u_i)$ implies that u_i is secured. Unfortunately, we cannot make such a claim, since Attackers 1 and 2 may be able to conduct a perfectly undetectable attack against u_i with less than $\delta_r(u_i)$ components. However, we do argue that having a large $\delta_r(u_i)$ provides a reasonable level of security.

Intuitively, having a large $\delta_r(u_i)$ implies that attacking u_i can trigger a large number of sensors. To avoid being detected from these sensors, an attacker should make a synchronized attack with attacked sensor and actuators. However, to be able to use the attacked actuators other than u_i for covering an attack against u_i, the attacker should have a precise model. Otherwise, he/she needs to compromise a large number of sensors. To illustrate this point, we introduce Attacker 3.

Assumption 6.4. *Attacker 3: (i) can read and change the values of control and measurement signals that correspond to attacked actuators and attacked sensors; (ii) knows* $[A]$,$[B]$,$[C]$*; and (iii) seeks to remain perfectly undetectable.*

Since Attacker 3 knows only $[A]$,$[B]$,$[C]$, he/she cannot constructively use other actuators to cover an attack against u_i. Namely, he/she does not know what signals to inject in these actuators. Yet, if the system is in a steady state, then Attacker 3 can use the replay attack strategy to conduct a perfectly undetectable attack against u_i. In this strategy, the attacker tries to cover an attack against u_i by replicating previously recorded steady state values from compromised sensors [84].

The following proposition shows that Attacker 3 needs to compromise at least $\delta_r(u_i)-1$ sensors to ensure that an attack against u_i remains perfectly undetectable. Hence, a large $\delta_r(u_i)$ makes more difficult for Attacker 3 to attack u_i.

Proposition 6.7. *Let u_i be an attacked actuator and Y_a be attacked sensors. If Attacker 3 can attack u_i and ensure the attack remains perfectly undetectable, then $|Y_a| \geq \delta_r(u_i) - 1$ has to hold. If $\delta_r(u_i) = +\infty$, then Attacker 3 cannot attack u_i while ensuring perfect undetectability.*

Proof. Case $\delta_r(u_i) \neq +\infty$: We prove the claim by showing that Y_a has to be a vertex separator of u_i and t in $\mathcal{G}_t$. Existence of a path from u_i to t that is not intersected by a separator implies that at least one sensor y_j is not compromised by the attacker. From the proof of Theorem 6.2, we know that there exists at least

one realization in which any attack against u_i triggers y_j. Since Attacker 3 knows only $[A], [B], [C]$, he/she does not know if attacks against u_i are visible in y_j. Thus, Attacker 3 needs to attack y_j to ensure being perfectly undetectable. Therefore, Y_a has to form a vertex separator of u_i and t. From Theorem 6.2, $\delta_r(u_i) - 1$ is the size of a minimum vertex separator of u_i and t in $\mathcal{G}_t$ (we subtract one from $\delta_r(u_i)$ to exclude u_i). Hence, $|Y_a| \geq \delta_r(u_i) - 1$ holds.

Case $\delta_r(u_i) = +\infty$: In this case, there has to exist a directed path between u_i and a protected sensor. This implies that Y_a cannot be a vertex separator. Hence, Attacker 3 cannot ensure that an attack against u_i remains perfectly undetectable, because he/she does not know if the aforementioned protected sensor is triggered when he/she attacks u_i. ■

Summary of Section 6.4.2

The main conclusions are as follows: (i) if $\delta_r(u_i)$ is small, then u_i is vulnerable with respect to Attackers 1 and 2 in any system realization from $\mathcal{R}$; (ii) a large value of $\delta_r(u_i)$ does not imply security with respect to these attackers, but it prevents them from easily gathering resources for attacking u_i in any system realization from $\mathcal{R}$; and (iii) a large $\delta_r(u_i)$ indicates security with respect to Attacker 3. For these reasons, it is useful to derive strategies for increasing δ_r that can be used in large-scale systems. This is the problem we address next.

Remark 6.8. *Increasing δ_r does not generally imply that we increase δ. However, the placement of new sensors cannot decrease δ (Proposition 6.2), so we definitely do not decrease this index. In fact, we illustrate in Section 6.5 that increasing δ_r may indirectly increase δ.*

6.4.3 Increasing the robust security index δ_r

We now derive a sensor allocation strategy for increasing δ_r. We assume that each of the newly deployed sensors measures only a single state, which is a commonly adopted assumption in sensor placement problems for large-scale systems [202].

Let u_i be an actuator for which we want to increase the robust security index. Consider the extended graph $\mathcal{G}_t$, and let x_{j_n} be a state for which there exists a directed path $u_i, x_{j_1}, \ldots, x_{j_n}$ in which the states $x_{j_1}, \ldots, x_{j_n}$ are not adjacent to the actuators $\mathcal{U} \setminus u_i$. We denote the set of all such states by X_i (see Figure 6.3 for an illustration). We now show that if we place a new sensor to measure a state from X_i, then we are guaranteed to increase $\delta_r(u_i)$. Moreover, if every state that is directly controlled by an actuator is also directly measured by a sensor, then placing a new sensor to measure a state from X_i is the only way to increase $\delta_r(u_i)$.

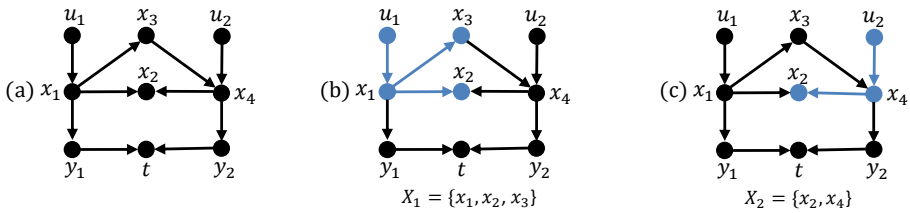

Figure 6.3: (a) An example of the extended graph $\mathcal{G}_t$. (b) The set X_1 of the actuator u_1. For example, $x_2 \in X_1$ because of the directed path u_1, x_1, x_2 (this path does not contain the states adjacent to u_2). Since x_4 is adjacent to u_2, we have $x_4 \notin X_1$. (b) The set X_2 of the actuator u_2.

Theorem 6.3. *Let u_i be an actuator with $\delta_r(u_i) \neq +\infty$, and let the set X_i be defined as above. The following statements holds:*

(i) If a sensor y_j is placed to measure a state from X_i and if $\delta'_r(u_i)$ is the robust index after the placement, then $\delta'_r(u_i) = +\infty$ if y_j is a protected sensor, and $\delta'_r(u_i) = \delta_r(u_j) + 1$ if y_j is an unprotected sensor.

(ii) If every state directly controlled by an actuator is directly measured by a sensor, then $\delta_r(u_i)$ is increased if and only if a new sensor is placed to measure a state from X_i.

Proof. We refer the reader to Appendix 6.F. ■

The sets $X_1, \ldots, X_{n_u}$ have two important properties. Firstly, they are not affected by the placement of new sensors. Thus, if we place n sensors to measure states from X_i, then $\delta_r(u_i)$ increases by n. Secondly, if we remove from $\mathcal{G}_t$ all the states that are adjacent to an actuator from $U \setminus u_i$, then X_i contains all the states to which u_i is connected with a directed path. Hence, X_i can be computed using the depth first search algorithm [203].

In what follows, we use the sets $X_1, \ldots, X_{n_u}$ to formulate a sensor allocation problem. In this chapter, we focus on allocating unprotected sensors. The goal is to place these sensors to increase δ_r for every actuator u_i by at least $k_i \in \mathbb{Z}_{\geq 0}$. We assume unprotected sensors to be inexpensive, so we do not have a sharp constraint on the number of sensors we should place. Yet, we still want to place the minimum number of them to achieve the desired benefit.

Let the set of unprotected sensors be $\mathcal{Y}_s = \{y_1, \ldots, y_{n_s}\}$, and let x_{y_i} be the state measured by $y_i \in \mathcal{Y}_s$. For every actuator u_i, we define a gain function

$$g_i(Y_p) = \min\{\textstyle\sum_{y_j \in Y_p} |x_{y_j} \cap X_i|,\ k_i\},$$

where $Y_p \subseteq \mathcal{Y}_s$ is the set of newly placed sensors. This function equals k_i, if at least k_i sensors from Y_p measure the states from X_i. We then have from Theorem 6.3

that $\delta_r(u_i)$ increases by at least (or exactly) k_i. Thus, the problem we want to solve can be written as follows:

$$\begin{aligned} &\underset{Y_p}{\text{minimize}} && |Y_p| \\ &\text{subject to} && G(Y_p) = \sum_{u_i \in \mathcal{U}} k_i, \end{aligned} \tag{6.6}$$

where $G(Y_p) = \sum_{u_i \in \mathcal{U}} g_i(Y_p)$. The objective function we are minimizing is the number of deployed sensors. Additionally, if the constraint is satisfied, then the robust indices of all the actuators are increased by the desired values.

We now show that this problem has the same submodular structure as the security measure allocation problem. Hence, its suboptimal solution with performance guarantees can be computed efficiently using Algorithm 3.1.

Proposition 6.8. *Let m^* be the optimal value of the problem (6.6), m_G be the value found by Algorithm 3.1, and $H(n) = \sum_{i=1}^{n} i^{-1}$. The following then holds:*

$$m_G \leq H\big(\max_{y_j \in \mathcal{Y}_s} G(y_j)\big) m^*. \tag{6.7}$$

Proof. It suffices to show that G is submodular, nondecreasing, and integer-valued. The problem (6.6) is then an instance of Problem 3.1, and the claim of the proposition immediately follows from Lemma 3.3. Let $f(Y_p) = \sum_{y_j \in Y_p} |x_{y_j} \cap X_i|$. Since $|x_{y_j} \cap X_i|$ is a constant, $f(Y_p)$ is a linear set function. Since linear set functions are submodular [152, Section 2], it follows that $f(Y_p)$ is submodular. Furthermore, $f(Y_p)$ is a nondecreasing function, since it represents a sum of nonnegative numbers. Next, note that $g_i(Y_p) = \min\{f(Y_p), k_i\}$, so it follows from Lemma 3.2 that g_i is submodular and nondecreasing. The function g_i is also integer valued, since f is integer valued and k_i is an integer. From the previous discussion and Lemma 3.1, it follows that G is submodular, nondecreasing, and integer valued. ■

The problem (6.6) can also be formulated as the following integer linear program:

$$\begin{aligned} &\underset{z \in \{0,1\}^{n_s}}{\text{minimize}} && \sum_{j=1}^{n_s} z_j \\ &\text{subject to} && \sum_{j=1}^{n_s} \mathbb{1}_{[x_{y_j} \in X_i]}\ z_j \geq k_i, \qquad \forall i \in \{1, \ldots, n_u\}. \end{aligned}$$

Here, z models the unprotected sensors that we intend to place. If $z_j = 1$ (resp. $z_j = 0$), then the sensor $y_j \in \mathcal{Y}_s$ is placed (resp. not placed). The objective function reflects that we want to place the minimum number of unprotected sensors, and the constraint guarantees that the robust security index of every actuator is sufficiently improved. Thus, one can also use integer linear program solvers to tackle (6.6).

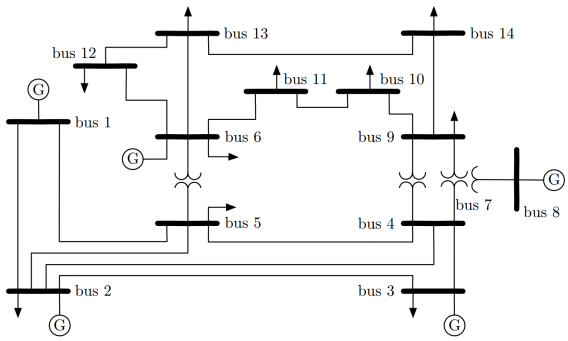

Figure 6.4: The IEEE 14 bus system (the figure is courtesy of Pasqualetti *et al.* [59]).

6.5 Illustrative examples

We now discuss the theoretical developments on illustrative numerical examples.

6.5.1 Model 1: Power grid

We consider the IEEE 14 bus system shown in Figure 6.4. We model the system using linearized swing equations where the generators are represented by two states (rotor angle ϕ_i and frequency ω_i) and load buses with one state (voltage angle θ_i) [204]. The parameters given in [205] are used. The system is controlled using five generators located at buses $\{1, 2, 3, 6, 8\}$. We assume that the operator has access to phasor measurement units providing the measurements of $\{\theta_1, \theta_3, \theta_5, \theta_7, \theta_9, \theta_{11}, \theta_{13}\}$. We also assume that every generator and every measurement can be compromised by the attacker, as well as some of the loads [206]. Particularly, the loads at buses $\{2, 5, 9, 14\}$ are assumed to have considerable effect to the network, and are modeled as additional actuators.

We consider the following system realizations: (i) normal operation, as shown in Figure 6.4; (ii) the power line between Buses 4 and 7 switched-off; (iii) a microgrid consisting of Bus 3 and Generator 3 detaches from the grid; and (iv) the measurement of θ_1 stops being available.

6.5.2 Example 1: Robustness of δ and δ_r

We first investigate the robustness of δ and δ_r with respect to system variations. To do this, we compute the values of δ and δ_r of all the generators in the aforementioned four realizations of the system, and plot the results in Figure 6.5.

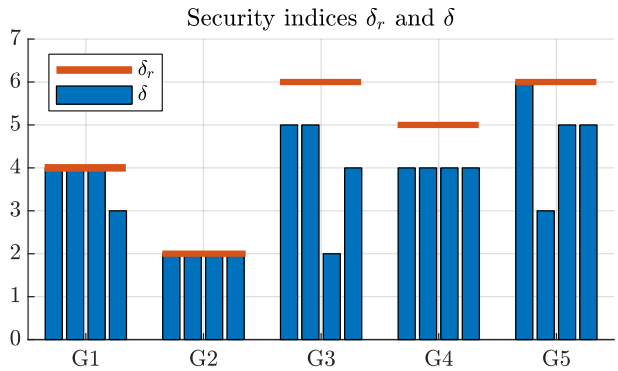

Figure 6.5: The values of the security index δ (blue bars) and the robust security index δ_r (red lines) of Generators 1-5 for different realizations of the system.

The results confirm that δ changes with respect to system realizations. Thus, if the operator decides to use δ as a security index, it is not sufficient to consider only one realization. For example, Generator 3 that appears to be the second most secure in the first realization, becomes one of the two most vulnerable in the third.

A less evident observation is that the use of δ can lead to a considerable security allocation cost. Particularly, we see that the minimum value of δ for all the generators is quite similar (except for maybe Generator 4). Therefore, ensuring that each generator has a sufficiently large security index δ for every realization of the system may require a large security investment.

Evidently, the values of δ_r do not depend on the realization. Therefore, having a small value of $\delta_r(u_i)$ implies that actuator u_i is vulnerable in any system realization. For example, since $\delta_r(G_2) = 2$, Attackers 1 and 2 can attack Generator 2 in any realization of the system by compromising only two components.

However, as it can be seen, δ_r is not a tight upper bound on δ. Thus, large δ_r does not necessarily imply security, which is the main drawback of δ_r. For instance, note that $\delta(G_3) = 2$ in the third realization. Hence, Attacker 1 can conduct a perfectly undetectable attack against Generator 3 in this realization by compromising two components, although $\delta_r(G_3) = 6$.

6.5.3 Example 2: Increasing δ and δ_r

Next, we investigate if by increasing δ_r we also increase δ. We focus on Generators 1 and 2, since these generators have the lowest values of δ_r. Based on the discussion from Section 6.4.3, we obtain that suitable locations for placing additional sensors are $X_1 = \{\phi_1, \omega_1, \theta_1\}$ for Generator 1, and $X_2 = \{\phi_2, \omega_2\}$ for Generator 2.

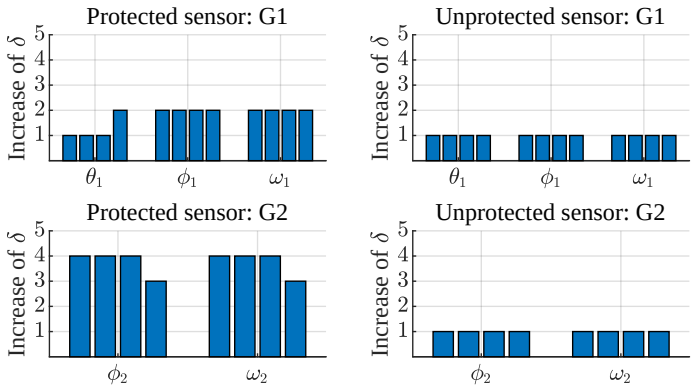

Figure 6.6: Increase of the security index δ for Generator 1 and Generator 2.

We first investigate how the placement of one protected sensor at a location from X_1 influences δ. While placing the protected sensor at any location from X_1 increases $\delta_r(G_1)$ to $+\infty$, it can be seen from Figure 6.6 that $\delta(G_1)$ does not increase to $+\infty$ in any of the four realizations. Yet, the increase of $\delta(G_1)$ by more than one is achieved in majority of the cases, which is impossible to achieve by placing a single unprotected sensor (Proposition 6.2).

The experiment is also conducted for Generator 2. Similarly, $\delta(G_2)$ does not increase to $+\infty$ in any of the four realizations. However, the placement of one protected sensor leads to increase of $\delta(G_2)$ by at least three for all the locations from X_2 and all the realizations we consider.

We also consider placing one unprotected sensor at locations from X_1, which increases $\delta_r(G_1)$ by one. Interestingly, as seen in Figure 6.6, the placement of one unprotected sensor at any of the locations from X_1 increases $\delta(G_1)$ in all four realizations. The same holds for X_2 and $\delta(G_2)$.

Overall, the experiment illustrates that by increasing δ_r, we may also indirectly increase δ. However, the experiment with the protected sensor demonstrates that we do not achieve the same level of improvement. This again shows that protecting the system against the advanced Attacker 1 may require much more resources than protecting it against less advanced attackers such as Attacker 3.

6.5.4 Example 3: Increasing δ_r in large-scale systems

We now demonstrate that the robust security indices δ_r can be computed and increased efficiently in large-scale systems. For this purpose, we use the IEEE 2383 bus system. This large-scale system has 3037 states and 327 generators. We model

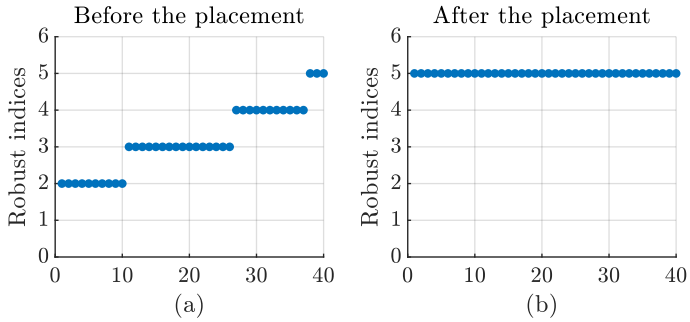

Figure 6.7: The values of the lowest forty robust security indices before and after the placement of unprotected sensors.

the system in the same way as the IEEE 14 bus, randomly select 40 percent of the states to be measurable, and 10 percent of the load buses to be attackable.

Next, we compute the robust indices of all the generators, and plot the lowest 40 robust security indices in Figure 6.7.(a). We emphasize that it takes only 114.03 seconds to compute the robust indices of all the generators on Intel Core i7-8650U computer. This confirms that the robust security indices can be computed efficiently in large-scale systems. As one can see, there is a large number of generators whose robust security indices are equal to 2, 3, or 4. Hence, these generators are vulnerable in any system realization.

Therefore, we consider the problem of allocating unprotected sensors to increase all the robust indices to at least 5. For this purpose, we first compute suitable locations for placing additional sensors according to the discussion from Section 6.4.3, and then solve the problem (6.6) using Algorithm 3.1. As we can see from Figure 6.7.(b), the robust indices are successfully increased after the placement. Additionally, this process takes only 0.57 seconds.

6.5.5 Model 2: Vehicular system

Consider the system consisting of two autonomous vehicles shown in Figure 6.8. Each vehicle is modeled by a single state representing its position relative to some moving reference frame. The operator can control both vehicles through signals u_1 and u_2, and knows the position of the second vehicle $y = x_2$. The operator's goal is to keep the distance between the vehicles equal to ten.

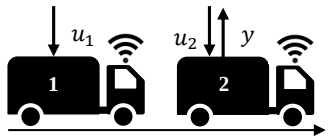

Figure 6.8: The platoon consisting of two autonomous vehicles. Each vehicle can be controlled by the operator through the signals u_1 and u_2. The operator also knows the position y of the second vehicle.

To study this formation control problem, we use the model from [136]:

$$x(k+1) = \begin{bmatrix} 1-2\alpha_1 & \alpha_1 \\ \alpha_2 & 1-2\alpha_2 \end{bmatrix} x(k) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u(k),$$
$$y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(k),$$

where $\alpha_1 = \alpha_2 = 0.1$. We initially assume that $x(0) = [0 \;\; 10]^T$ and $u(k) = [-1 \;\; 2]^T$ for any $k \in \mathbb{Z}_{\geq 0}$, so that desired behavior is achieved prior to attacks.

6.5.6 Example 4: Properties of Attacker 1 and Attacker 2

We now illustrate some properties of Attackers 1 and 2[1]. Both attackers control u_1 and y, and have the goal to disrupt the platoon formation without the operator noticing. In the following, we discuss in which situations the attackers can achieve this objective. By Δy_1 (resp. Δy_2), we denote the difference between the measurement expected in the normal operation and the received measurement in the case of the first (resp. second) attacker. If Attacker 1 (resp. Attacker 2) conducts a perfectly undetectable attack, then $\Delta y_1 \equiv 0$ (resp. $\Delta y_2 \equiv 0$) holds.

Case 1: The first case illustrates that both of the attackers can conduct a perfectly undetectable attack when the system is in a steady state. Attacker 1 constructs the attack as follows:

$$a_1(k) = -k, \qquad a_3(k+2) = 1.6a_3(k+1) - 0.63a_3(k) - 0.1a_1(k), \tag{6.8}$$

which is according to the strategy introduced in the proof of Proposition 6.1. Attacker 2 applies the attack signals

$$a_1(k) = -k, \qquad a_3(k) = -x_2(k) + y(0), \tag{6.9}$$

which is according to the strategy introduced in the proof of Proposition 6.6. As we can see from Figure 6.9, Case 1, both of the attackers remain perfectly undetectable.

[1]The properties of Attacker 2 we outline next are the same as for Attacker 3, which is the reason why we do not explicitly consider Attacker 3 in this example.

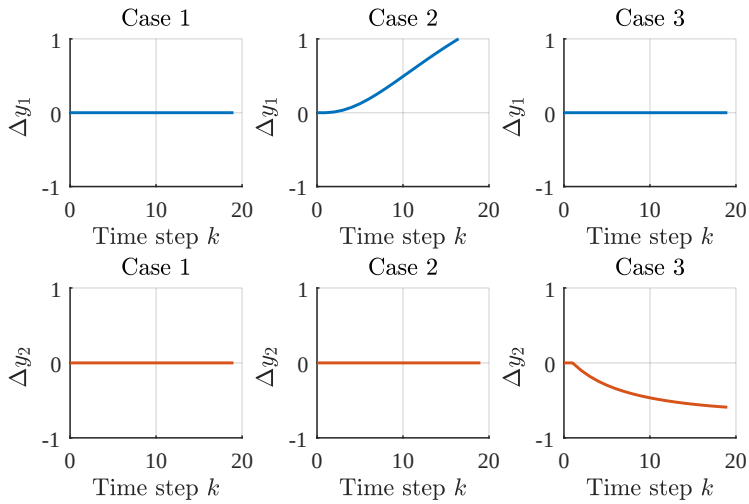

Figure 6.9: Differences Δy_1 and Δy_2 of the expected and attacked sensor measurements in three different cases. If Attacker 1 (resp. Attacker 2) conducts a perfectly undetectable attack, then $\Delta y_1 \equiv 0$ (resp. $\Delta y_2 \equiv 0$) holds.

Case 2: The second case illustrates the fragility of Attacker 1 with respect to modeling errors. Particularly, assume that Attacker 1 believes that $\alpha_2' = 0.11$. In that case, he/she constructs the attack as follows:

$$a_1(k) = -k, \qquad a_3(k+2) = 1.58a_3(k+1) - 0.613a_3(k) - 0.11a_1(k).$$

Attacker 2 applies the same signals as in the previous case. From Figure 6.9, Case 2, we can see that Attacker 1 is revealed, while Attacker 2 remains undetected. Generally, Attacker 2 can also be vulnerable to modeling errors. However, the fact that this attacker uses only a fraction of the model (in this case none), lowers his/her chances to be detected because of modeling errors.

Case 3: Finally, assume the scenario where the operator increases u_2 by 0.1 at $k = 2$. The attackers apply the attacks from Case 1. From Figure 6.9, Case 3, we see that Attacker 2 is revealed. This illustrates that the steady state assumption is generally required for Attacker 2 to remain perfectly undetectable. Namely, Attacker 2 does not know neither u_2 nor the equation for x_2. Hence, when y starts changing, he/she cannot distinguish if this is because of the attack or a change in u_2. We also see that Attacker 1 remains undetected. The reason is that the attack (6.8) can be computed prior to the attack and implemented in a feedforward manner. This makes the attack decoupled from $x(0)$ and u.

6.6 Summary

This chapter introduced the actuator security indices δ and δ_r. A method for computing δ was derived, and it was shown that δ can potentially be increased (resp. decreased) by placing additional sensors (resp. actuators). We then showed that δ may not be an appropriate index for large-scale systems, since it is NP-hard to compute, vulnerable to system variations, and based on the assumption that the attacker knows the entire system model. In contrast, the robust index δ_r can be computed efficiently, can characterize actuators vulnerable in any system realization, and can be related to both the full and limited model knowledge attackers. Additionally, a sensor placement problem for increasing δ_r was proposed, and it was shown that a suboptimal solution with performance guarantees of this problem can be computed efficiently. Finally, the properties of δ and δ_r, as well as some of the theoretical results, were clarified by means of numerical examples of power grids and autonomous vehicles. We now move to the next chapter, where we study a sensor placement game based on actuator security indices.

Appendix to Chapter 6

6.A Proof of Proposition 6.1

We first introduce an auxiliary lemma that we use in the proof.

Lemma 6.1. *[136, Theorem 1] [37, Theorem 7] A perfectly undetectable attack conducted with components $I_a \subseteq I$ exists if and only if* $\operatorname{nrank}[\,G^{(I_a)}\,] < |I_a|$.

Proof of Proposition 6.1: ($\Rightarrow$) Let $\mathcal{A}$ be the $\mathcal{Z}$-transform of an attack a. We show that if there exists a perfectly undetectable attack $\mathcal{A}$ with $\mathcal{A}_i \neq 0$, then the condition (6.2) has to hold. We split the proof into two cases.

Case 1: $\operatorname{nrank}[\,G^{(I_a \setminus i)}\,] = |I_a| - 1$. Since undetectable attacks are possible to conduct using the components I_a, then it follows from Lemma 6.1 that

$$\operatorname{nrank}[\,G^{(I_a)}\,] < |I_a|. \tag{6.10}$$

In addition, we have

$$\operatorname{nrank}[\,G^{(I_a)}\,] \geq \operatorname{nrank}[\,G^{(I_a \setminus i)}\,] = |I_a| - 1. \tag{6.11}$$

From (6.10) and (6.11), it follows that $\operatorname{nrank}[\,G^{(I_a)}\,] = |I_a| - 1$. Therefore, we conclude that $\operatorname{nrank}[\,G^{(I_a)}\,] = \operatorname{nrank}[\,G^{(I_a \setminus i)}\,]$.

Case 2: $\operatorname{nrank}[\,G^{(I_a \setminus i)}\,] = r < |I_a| - 1$. Let $z' \in \mathbb{C}$ satisfy $\operatorname{rank}[\,G^{(I_a \setminus i)}(z')\,] = r$, and let I_b be a set that contains indices of any r linearly independent columns of

$G^{(I_a \setminus i)}(z')$. Since $G^{(I_b)}$ has r columns and $\text{nrank}[\, G^{(I_b)} \,]$ cannot be larger than the number of the columns of $G^{(I_b)}$, we have

$$\text{nrank}[\, G^{(I_b)} \,] \leq r. \tag{6.12}$$

From the definition of the normal rank, it follows that

$$\text{nrank}[\, G^{(I_b)} \,] = \max_{z \in \mathbb{C}} \text{rank}[\, G^{(I_b)}(z) \,] \geq \text{rank}[\, G^{(I_b)}(z') \,] = r. \tag{6.13}$$

From (6.12) and (6.13), we conclude that the following equality holds:

$$\text{nrank}[\, G^{(I_a \setminus i)} \,] = \text{nrank}[\, G^{(I_b)} \,] = r. \tag{6.14}$$

Next, note that $\text{nrank}[\, G^{(I_b)} \; G^{(j)} \,] = r$ for any $j \in I_a \setminus i$. Otherwise, we would have $\text{nrank}[\, G^{(I_a \setminus i)} \,] > r$. Hence, we can find rational matrices P and $Q \neq 0$ that satisfy $G^{(I_b)} P + G^{(j)} Q = 0$ [207, p. 31]. Thus, the columns of $G^{(I_b)}$ span all the columns of $G^{(I_a \setminus i)}$, and we can find $\mathcal{A}'$ such that

$$G^{(I_a \setminus i)} \mathcal{A}^{(I_a \setminus i)} = G^{(I_b)} \mathcal{A}', \tag{6.15}$$

where $\mathcal{A}^{(I_a \setminus i)}$ is the vector that contains the elements of $\mathcal{A}$ from the set $I_a \setminus i$. From (6.15) and $G\mathcal{A} = 0$ ($\mathcal{A}$ is a perfectly undetectable attack), we have

$$G\mathcal{A} = G^{(I_a \setminus i)} \mathcal{A}^{(I_a \setminus i)} + G^{(i)} \mathcal{A}_i \overset{(6.15)}{=} G^{(I_b)} \mathcal{A}' + G^{(i)} \mathcal{A}_i \overset{G\mathcal{A}=0}{=} 0.$$

This implies that $[\mathcal{A}'^T \; \mathcal{A}_i]^T$ is a perfectly undetectable attack against $[G^{(I_b)} \; G^{(i)}]$ with $\mathcal{A}_i \neq 0$. From this fact and $\text{nrank}[\, G^{(I_b)} \,] = |I_b|$, it follows from Case 1 that the condition (6.2) holds for the set of components $I_b \cup i$. Thus, we have

$$\text{nrank}[G^{(I_b)} \; G^{(i)}] \overset{\text{Case 1}}{=} \text{nrank}[\, G^{(I_b)} \,] \overset{(6.14)}{=} \text{nrank}[\, G^{(I_a \setminus i)} \,]. \tag{6.16}$$

Since $G^{(I_b)}$ spans the columns of $G^{(I_a \setminus i)}$, we have

$$\text{nrank}[\, G^{(I_b)} \, G^{(i)} \,] = \text{nrank}[\, G^{(I_a \setminus i)} \, G^{(i)} \,] = \text{nrank}[\, G^{(I_a)} \,]. \tag{6.17}$$

From (6.16) and (6.17), we conclude that $\text{nrank}[\, G^{(I_a)} \,] = \text{nrank}[\, G^{(I_a \setminus i)} \,]$. Thus, the condition (6.2) holds in this case as well.

($\Leftarrow$) If (6.2) holds, then the column $G^{(i)}$ has to be spanned with the columns of $G^{(I_a \setminus i)}$. Therefore, there exist real rational functions P and $Q \neq 0$ of appropriate dimensions that satisfy

$$G^{(I_a \setminus i)} P + G^{(i)} Q = 0.$$

Thus, an arbitrary attack $\mathcal{A}_i$ can be masked by applying $\mathcal{A}^{(I_a \setminus i)} = P\mathcal{A}_i / Q$ on the remaining attacked components.

6.B Proof of Theorem 6.1

To prove NP-hardness of Problem 6.1, it suffices to show that every instance of an NP-hard problem can be mapped into an instance of Problem 6.1. For this purpose, we use the sparse recovery problem:

$$\begin{aligned} \underset{d}{\text{minimize}} \quad & \|d\|_0 \\ \text{subject to} \quad & Fd = z, \end{aligned} \tag{6.18}$$

where $F \in \mathbb{R}^{p\times m}$ and $z \in \mathbb{R}^p$ are given. This problem is known to be NP-hard [208].

Let F and z be arbitrarily selected. Set $A = 0_{m\times m}$, $B = I_m$, $C = [-z\ F]$, $D_a = 0_{p\times m}$, and $u_i = u_1$. Then $x(k+1) = a(k)$ and $y(k) = Ca(k-1)$. Hence, Problem 6.1 reduces to

$$\begin{aligned} \underset{a}{\text{minimize}} \quad & \|a\|_0 \\ \text{subject to} \quad & Ca(k) = 0_m, \\ & a_1 \not\equiv 0. \end{aligned} \tag{6.19}$$

To solve (6.19) for all k, it suffices to solve it for a single k. Thus, (6.19) reduces to

$$\begin{aligned} \underset{a(0)}{\text{minimize}} \quad & \|a(0)\|_0 \\ \text{subject to} \quad & Ca(0) = 0_m,\ a_1(0) = 1, \end{aligned}$$

where the substitution of $a_1(0) \neq 0$ with $a_1(0) = 1$ is without loss of generality. Let $a(0) = [1\ d^T]^T$. Then minimizing $\|a(0)\|_0$ is equivalent to minimizing $\|d\|_0$, which is the objective function of (6.18) (Observation 1). Moreover, we also have

$$Ca(0) = [-z\ F]a(0) = -z + Fd.$$

Thus, the constraint $Ca(0) = 0$ is equivalent to the constraint $Fd = z$ from the problem (6.18)(Observation 2). From Observations 1 and 2, it directly follows that every instance of the NP-hard problem (6.18) can be mapped into Problem 6.1, which concludes the proof.

6.C Proof of Theorem 6.2

($\Leftarrow$) Let $X_a \cup Y_a$ be a vertex separator of u_i and t in the extended graph $\mathcal{G}_t$. To prove the claim, we introduce an attack strategy that uses the components U_a and Y_a. We then prove that this strategy is actively using u_i and remains perfectly undetectable in any system realization from $\mathcal{R}$.

For actuator u_i, the attacker injects a signal $a_i \not\equiv 0$, which ensures that u_i is used in the attack actively. For other actuators $u_j \in U_a \setminus u_i$, the attack is given by

$$a_j(k) = -\frac{A(p,:)}{B(p,j)}x(k). \tag{6.20}$$

Here, $A(p,:)$ is the row of A corresponding to the state x_p adjacent to u_j, and $B(p,j)$ is the non-zero element of B multiplying u_j ($B(p,j) \neq 0$ in any system realization due to Assumption 6.2). For every $y_l \in Y_a$, the attack is given by

$$a_{n_u+l}(k) = -C(l,:)x(k). \tag{6.21}$$

For the attacker with the full model knowledge, this strategy can be constructed for any system realization. Firstly, the attacker knows $A(p,:), B(p,j), C(l,:)$. Secondly, the attacker can predict the value of $x(k)$ for any $k \in \mathbb{Z}_{\geq 0}$ based on the model knowledge and the attack signals he/she injects in the system. We now prove that this strategy is perfectly undetectable, that is, $y \equiv 0$.

Consider first the attacked sensors. For any $y_l \in Y_a$ and $k \in \mathbb{Z}_{\geq 0}$, we have

$$y_l(k) = C(l,:)x(k) + a_{n_u+l}(k) \overset{(6.21)}{=} 0.$$

Thus, the attacked sensor measurements are equal to zero.

Consider now the states X_a. Let $x_p \in X_a$ and $u_j \in U_a \setminus u_i$. Then

$$x_p(k+1) = A(p,:)x(k) + B(p,j)a_j(k) \overset{(6.20)}{=} 0.$$

Thus, the attack cannot be detected by measuring the states X_a.

Let X_b be the set of all the states for which there exists a directed path from u_i that does not contain the states from X_a. These states cannot be measured using the non-attacked sensors. That would imply that there exists a directed path between u_i and t not intersected by $X_a \cup Y_a$, which is in contradiction with the assumption that $X_a \cup Y_a$ is a vertex separator of u_i and t.

Finally, let $X_c = \mathcal{X} \setminus (X_b \cup X_a)$. Note that the directed edges (x_b, x_c), $x_b \in X_b$, $x_c \in X_c$, cannot exist. That would imply that there exists a directed path from u_i to x_c that does not contain the states from X_a, so x_c would belong to X_b. Thus, the states from X_c cannot be directly influenced by the states from X_b. Since $x(0) = 0_{n_x}$, $u \equiv 0$, and the states X_a are equal to zero, we conclude that the states X_c are also equal to zero during the attack. Thus, the attack cannot be detected by measuring the states X_c.

From the previous four paragraphs, it follows that both the attacked and the non-attacked sensor measurements are equal to zero. Therefore, the proposed attack strategy is perfectly undetectable, which completes the first part of the proof.

($\Rightarrow$) The proof is by contradiction. Assume that $X_a \cup Y_a$ is not a vertex separator of u_i and t in $\mathcal{G}_t$. Then there exists at least one simple directed path $u_i, x_{i_0}, \ldots, x_{i_n}, y_l, t$ (Path 1) not intersected by $X_a \cup Y_a$. We show that this implies existence of at least one realization $(A,B,C) \in \mathcal{R}$ in which a perfectly undetectable attack against u_i cannot be conducted. Particularly, let us consider any feasible realizations of A and C with the following properties:

(i) for x_{i_0} from Path 1, $A(i_0,:) = 0^T_{n_x}$ (x_{i_0} is not influenced by other states);

(ii) for any $x_{i_k} \neq x_{i_0}$ from Path 1, we have $A(i_k, j) \neq 0$ if $j = i_{k-1}$, and $A(i_k, j) = 0$ otherwise (ensures that the only state that influences x_{i_k} is $x_{i_{k-1}}$);

(iii) we have $C(l, j) \neq 0$ if $j = i_n$, and $C(l, j) = 0$ otherwise (ensures that $y_l(k) \neq 0$ every time $x_{i_n}(k) \neq 0$).

Let $a_i \not\equiv 0$ be an arbitrary attack signal against u_i, and let k_0 be the first time instant for which $a_i(k_0) \neq 0$. Since a_i is the only attack signal that can directly influence x_{i_0} (Assumption 6.2), we have

$$x_{i_0}(k_0 + 1) = A(i_0,:)x(k_0) + B(i_0, i)a_i(k_0).$$

Since $A(i_0,:) = 0^T_{n_x}$ (by the construction) and $B(i_0, i) \neq 0$ (Assumption 6.2), we have $x_{i_0}(k_0+1) \neq 0$. Next, note that A is such that the only state that influences x_{i_1} is x_{i_0}. Moreover, since x_{i_1} cannot be directly influenced by the attacked actuators ($x_{i_1} \notin X_a$), it follows that

$$x_{i_1}(k_0 + 2) = A(i_1, i_0)x_{i_0}(k_0 + 1) \neq 0.$$

By applying the similar reasoning to all other states from Path 1, it can be shown that $x_{i_n}(k_0 + n + 1) \neq 0$. From the way we constructed C, it immediately follows that $y_l(k_0 + n + 1) \neq 0$. Therefore, the attack is revealed. Since a_i was arbitrarily selected, perfectly undetectable attacks with u_i actively used do not exist in this realization. This contradicts the initial assumption and establishes the claim.

6.D Proof of Proposition 6.4

Case 1: The problem (6.5) is solvable. Let (U_a, Y_a) be a solution of the problem (6.5) and $X_a \cup Y_a$ be a corresponding vertex separator. Let $E_c \subseteq \mathcal{E}_i$ be constructed as follows: (i) for every $x_k \in X_a$, we add $(x_{k_{in}}, x_{k_{out}})$ to E_c; and (ii) for every $y_j \in Y_a$, we add (y_j, t) to E_c. It follows from the construction of $\mathcal{G}_i$ that the edges added to E_c have the cost

$$\delta_c = |\mathcal{U}_a \setminus i| + |\mathcal{Y}_a| = \delta_r(u_i) - 1.$$

We now show that E_c is an edge separator of u_i and t in $\mathcal{G}_i$ (Claim 1) of the minimum cost (Claim 2). This implies that $\delta_r(u_i) = \delta_c + 1 = \delta^* + 1$.

Claim 1. The proof is by contradiction. Assume that E_c is not an edge separator. Then there exists a simple directed path

$$u_i, x_{j_1}, \ldots, x_{j_n}, y_l, t \qquad \text{(Path 1)}$$

in $\mathcal{G}_i$ that is not intersected by E_c. By the construction of $\mathcal{G}_i$, it follows that there exists a simple directed path

$$u_i, x_{k_1}, \ldots, x_{k_m}, y_l, t \qquad \text{(Path 2)}$$

in $\mathcal{G}_t$ obtained from Path 1 by replacing every pair $x_{p_{\text{in}}}, x_{p_{\text{out}}}$ that corresponds to x_p of Type 1 by x_p. Path 2 has to be intersected with $X_a \cup Y_a$, so either there exists $x_p \in X_a$ that belongs to Path 2 or $y_l \in Y_a$. Yet, then either $(x_{p_{in}}, x_{p_{out}})$ or (y_l, t) belongs to E_c. This implies that Path 1 has to be intersected by E_c. This contradicts the existence of Path 1, so Claim 1 holds.

Claim 2. The proof is again by contradiction. Assume there exists an edge separator E'_c with a cost $\delta' < \delta_c$. Let U'_a and Y'_a be constructed as follows. For each $(x_{k_{in}}, x_{k_{out}}) \in E_c$, we add u_j to U'_a, where u_j is adjacent to x_k. We also add u_i to U'_a. For each $(y_l, t) \in E_c$, we add y_l to Y'_a. Note that E'_c cannot contain edges of other types, because their weight is $+\infty$, which would imply $\delta' > \delta_c$.

We now prove that (U'_a, Y'_a) is a feasible point of the problem (6.5). Assume this is not the case. It then follows that there exists a simple directed path

$$u_i, x_{k_1}, \ldots, x_{k_m}, y_l, t \qquad \text{(Path 1')}$$

in $\mathcal{G}_t$ in which the states $x_{k_1}, \ldots, x_{k_m}$ are not adjacent to $U'_a \setminus u_i$ and $y_l \notin Y'_a$. This implies that there exists a simple directed path in $\mathcal{G}_i$ obtained from Path 1' by replacing each node x_p of Type 1 from this path by the pair $x_{p_{\text{in}}}, x_{p_{\text{out}}}$. By the construction of U'_a, Y'_a, and $\mathcal{G}_i$, this path cannot be intersected by E'_c. This contradicts the assumption that E'_c is an edge separator. Hence, (U'_a, Y'_a) has to be a feasible point of the problem (6.5).

However, then (U_a, Y_a) is not a solution of the problem (6.5). Namely, by the construction of U'_a and Y'_a, we have

$$|U'_a \cup Y'_a| = \delta' + 1 < \delta_c + 1 = |U_a \cup Y_a|.$$

Thus, E'_c cannot exist. This implies that E_c is an edge separator of the minimum cost, so Claim 2 holds.

Case 2: The problem (6.5) is not solvable. This situation occurs when there exists a simple directed path

$$u_i, x_{j_1}, \ldots, x_{j_n}, y_l, t$$

in $\mathcal{G}_t$ that consists of only Type 2 states and a protected measurement y_l. By the construction of $\mathcal{G}_i$, this path exists also in $\mathcal{G}_i$, and the weights of its edges are $+\infty$. Any edge separator needs to cut this path, which implies $\delta^* = +\infty$.

6.E Proof of Proposition 6.6

($\Rightarrow$) The proof is by contradiction. If $X_a \cup Y_a$ is not a vertex separator of u_i and t in $\mathcal{G}_t$, then we know from the proof of Theorem 6.2 that we can find a realization from

$\mathcal{R}$ in which it is impossible to conduct a perfectly undetectable attack against u_i. Thus, $X_a \cup Y_a$ has to be a vertex separator of u_i and t.

($\Leftarrow$) If $X_a \cup Y_a$ is a vertex separator of u_i and t, then the attacker can conduct a perfectly undetectable attack against u_i using the following strategy:

(i) For the targeted actuator u_i, the attacker injects an arbitrary signal $a_i \not\equiv 0$.

(ii) For $u_j \in U_a \setminus u_i$, the attack is given by (6.20).

(iii) For $y_l \in Y_a$, the attacker selects $a_{l+n_u}(k)$ to maintain $y_l \equiv 0$.

Attacker 2 can construct this attack. Firstly, for any $u_j \in U_a \setminus u_i$, we have

$$a_j(k) = -\frac{A(p,:)}{B(p,j)}x(k) = -\frac{1}{B(p,j)}\sum_{x_r \in \mathcal{N}_{x_p}^{\text{in}}} A(p,r)x_r(k),$$

where x_p is the state adjacent to u_j. Attacker 2 can construct this signal because he/she knows $A(p,:)$, $B(p,:)$, and $x_r(k)$ for any of the in-neighbors of x_p.

Secondly, Attacker 2 can set the measurements and control actions corresponding to the attacked sensors and actuators to an arbitrary value. Hence, he/she can maintain $y_l \equiv 0$ for any $y_l \in Y_a$. The proof that $y \equiv 0$ can then be found in the proof of Theorem 6.2.

6.F Proof of Theorem 6.3.

Proof of (i): By placing y_j to measure any of the states from X_i, we introduce at least one directed path $u_i, \ldots, y_j, t$ from u_i to t, which does not contain states adjacent to $\mathcal{U} \setminus u_i$. Thus, the only way to eliminate this path is by adding y_j to a new vertex separator. If y_j is protected, then the path is impossible to eliminate. Hence, $\delta_r'(u_i) = +\infty$ holds. Otherwise, the attacker has to attack y_j to eliminate the path. Thus, $\delta_r'(u_i) = \delta_r(u_i) + 1$ holds in this case.

Proof of (ii): Let (U_a, Y_a) be a solution of the problem (6.5). We first form another solution (U_a', Y_a') of (6.5). The set Y_a' is formed by removing from Y_a any y_s which directly measures $x_k \in \mathcal{X}$ that is adjacent to $u_l \in \mathcal{U} \setminus u_i$. As a substitute of y_s, we add u_l to U_a'. We then add all the actuators U_a to U_a'. This ensures that for all the states that are both directly controlled by an actuator and directly measured by a sensor, we always select an actuator to belong to a solution of the problem (6.5) rather than a sensor. Let X_a' be defined as in (6.4) based on U_a'.

Assume that a new sensor is placed on a location $x_l \notin X_i$. If there are no directed paths from u_i to x_l, then (U_a', Y_a') is still a solution of the problem (6.5). Thus, $\delta_r(u_i)$ is not increased in this case.

Assume now that there exists at least one simple directed path from u_i to x_l. Let us select arbitrarily one of these paths $u_i, \ldots, x_l$ (Path 1). Since $x_l \notin X_i$, there has to exist at least one state x_p from Path 1 such that $(u_k, x_p) \in \mathcal{E}_{ux}$. Then we have the following three possibilities.

(i) $X'_a \cup Y'_a$ is a vertex separator of u_i and x_p;

(ii) $x_p \in X'_a$;

(iii) $X'_a \cup Y'_a$ is not a vertex separator of u_i and x_p.

In cases (i) and (ii), $X'_a \cup Y'_a$ intersects Path 1. We now show by contradiction that (iii) cannot hold. Suppose that (iii) holds. Since every state directly controlled by an actuator is directly measured by a sensor, we conclude that there exists a directed path between u_i and t passing through x_p that is not intersected by $X'_a \cup Y'_a$. This implies that (U'_a, Y'_a) is not a solution of (6.5), which leads to a contradiction.

From the previous paragraph, it follows that Path 1 has to be intersected by $X'_a \cup Y'_a$. Since Path 1 was arbitrarily selected, we conclude that directed paths from u_i to x_l that are not intersected by $X'_a \cup Y'_a$ do not exist. Therefore, $\delta_r(u_i)$ cannot be increased by placing sensors at locations outside X_i.

Chapter 7

Allocation of protected sensors

This chapter studies an operator-attacker game based on actuator security indices. In this game, the operator seeks to allocate a limited number of protected sensors to improve actuator security indices, while the attacker seeks to select an actuator with a small value of the security index to attack. We also assume that the attacker uses the extended replay strategy, which is inspired by the Stuxnet attack. The purpose of studying this game is to compute a mixed monitoring strategy that lies in a NE. Such a strategy can be computed by solving a linear program. However, this program is challenging to solve for large-scale systems, since the size of the program grows combinatorially with the number of protected sensors that the operator seeks to allocate. Therefore, the question we pursue is how to compute a NE monitoring strategy, or a good approximation of this strategy, in a tractable manner.

To answer this question, we first express the payoff function of the game analytically. Using this expression, we derive an approximate NE (ϵ-NE). We then present cases when the ϵ-NE becomes exact, and outline some game-theoretic intuition behind this equilibrium. We also discuss ways to further improve the monitoring strategy from the ϵ-NE by deploying additional sensors, focusing on the most vulnerable actuators, and using numerical CGP. Finally, we conduct experiments on a benchmark of a large-scale power grid, and show that the tools we propose allow us to construct NE monitoring strategies in a tractable manner.

The chapter is organized as follows. Section 7.1 introduces the problem. Section 7.2 derives an analytical expression for the payoff function. Section 7.3 presents and discusses the aforementioned ϵ-NE of the game. Section 7.4 explains how the monitoring strategy from this ϵ-NE can be improved. Section 7.5 contains numerical experiments. Section 7.6 concludes the chapter. The appendix contains lengthy proofs, and the definition of the security index we use in this chapter.

7.1 Model setup and problem formulation

This section introduces the system model, the game, and the problem of computing a NE monitoring strategy.

7.1.1 System model

We model the control system by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ba_u(k), \\ y(k) &= Cx(k) + a_y(k), \end{aligned} \tag{7.1}$$

where $x(k) \in \mathbb{R}^{n_x}$ are the system states, $u(k) \in \mathbb{R}^{n_u}$ are the control actions applied through the actuators, $y(k) \in \mathbb{R}^{n_y}$ are the sensor measurements, $a_u(k) \in \mathbb{R}^{n_u}$ are the actuator attacks, and $a_y(k) \in \mathbb{R}^{n_y}$ are the sensor attacks. The elements of a_u (resp. a_y) that correspond to the attacked actuators (resp. sensors) can take any value. The remaining elements of a_u and a_y are equal to zero for any k. We define by $\mathcal{X} = \{x_1, \ldots, x_{n_x}\}$ the set of states, $\mathcal{U} = \{u_1, \ldots, u_{n_u}\}$ the set of actuators, and $\mathcal{Y} = \{y_1, \ldots, y_{n_y}\}$ the set of sensors. All the sensors from $\mathcal{Y}$ and all the actuators from $\mathcal{U}$ can be compromised by the attacker. We adopt the following assumptions.

Assumption 7.1. *We assume that* $x(0) = 0_{n_x}$ *and* $u \equiv 0$.

Assumption 7.1 is introduced because we focus on an attack strategy that exploits the fact that the system is in a steady state. The steady state $x(0) = 0_{n_x}$ and $u \equiv 0$ is assumed for the sake of conciseness, and can be replaced with any other steady state where x and u are constant in absence of attacks. Since control algorithms often have a goal to maintain the system in a steady state, we believe that this assumption is without a significant loss of generality.

Assumption 7.2. *The following statements hold: (i)* $B = [e_{i_1} \ldots e_{i_{n_u}}]$ *and* $C = [e_{j_1} \ldots e_{j_{n_y}}]^T$*; (ii)* $\operatorname{rank}[B] = n_u$*; and (iii) every state directly controlled by an actuator is directly measured by a sensor.*

Assumption 7.2.(i) is commonly adopted in placement problems [199, 200, 202]. It implies that every actuator (resp. sensor) directly controls (resp. measures) only one state. Same as in the previous chapter, Assumption 7.2.(ii) excludes the cases where attacks trivially cancel each-other, and the cases where an actuator does not affect the system. Assumption 7.2.(iii) enables some derivations. This assumption can be justified by the fact that modern day control systems are highly sensed [128]. Additionally, states that are directly controlled by an actuator can be measured to compute a control signal for that actuator [201].

Remark 7.1. *To keep the model simple, we assume the matrices A, B, C to be fixed. Nevertheless, we show that monitoring strategies derived in this chapter are robust with respect to changes that can occur in these matrices.*

7.1.2 Game model

To derive a monitoring strategy, we consider a two-player zero-sum game. We now introduce the components of this game, and some game-theoretic terminology.

Players, pure strategies, and the payoff

Player 1 (P1) is the operator, whose goal is to improve security of the actuators by allocating n protected sensors at a subset of states $X \subseteq \mathcal{X}$. Hence, the set of pure strategies $\mathcal{A}_1$ of P1 is given by $\mathcal{A}_1 = \{X \subseteq \mathcal{X} \mid |X| \leq n\}$.

Player 2 (P2) is the attacker, who seeks to attack an actuator and remain perfectly undetectable (see Definition 6.1). P2 also seeks to compromise the minimum number of components to achieve these goals. To simplify the analysis, we are not interested in the exact set of components P2 compromises, but only their number. This number is determined by a security index, which we define later in this section. In that case, we can define the set of pure strategies of P2 by $\mathcal{A}_2 = \mathcal{U}$. That is, P2 only selects the target of an attack.

Next, P2 is assumed to follow the extended replay strategy. Let U_a be attacked actuators, Y_a be attacked sensors, and $u_i \in U_a$ be the target of an attack. The extended replay strategy can be defined as follows:

(i) a_{u_i} is an arbitrarily selected nonzero attack;

(ii) $a_{u_j}(k) = -A(l,:)x(k)$ for every actuator $u_j \in U_a \setminus u_i$, where l is the index of the state x_l directly controlled by u_j;

(iii) $a_{y_j}(k) = -C(j,:)x(k)$ for every sensor $y_j \in Y_a$.

In words, P2 first collects the information on the steady state values of the states directly controlled by the attacked actuators $U_a \setminus u_i$ and the attacked sensor measurements Y_a. He/she then tries to cover an attack against the targeted actuator u_i by keeping the aforementioned states and measurements at their steady state values. An illustrative example is provided in Figure 7.1.

Remark 7.2. *The extended replay strategy can be constructed with limited model knowledge. In fact, Attacker 2 from the previous chapter is able to construct this strategy. Particularly, to construct a_{y_j} corresponding to $y_j \in Y_a$, P2 needs only a*

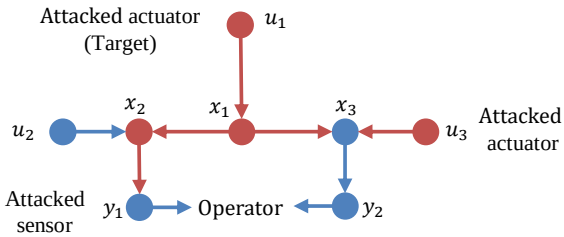

Figure 7.1: An illustration of the extended replay attack. The states, control actions, and measurements that remain in the steady state during the attack are indicated with blue. In this case, P2 covers the attack against u_1 by keeping the value of y_1 and x_3 in the steady state.

steady state value of y_j. An attack a_{u_j} corresponding to $u_j \in U_a$ can be written as

$$a_{u_j}(k) = -A(l,:)x(k) = -\sum_{x_r \in \mathcal{N}_l^{in}} A(l,r)x_r(k),$$

where $\mathcal{N}_l^{in} = \{x_r \in \mathcal{X} : A(l,r) \neq 0\}$ is the in-neighborhood of x_l. Hence, P2 can construct a_{u_j} based on the local model knowledge and the measurements of local states. Moreover, this strategy can be lucrative for P2 even if he/she possesses the full model knowledge. The next section shows that resources that enable a perfectly undetectable extended replay attack against u_i can be found efficiently in large-scale systems. These properties make the extended replay strategy a serious threat.

Remark 7.3. *In a traditional replay strategy [84], P2 tries to cover an attack by sending the steady state measurements from attacked sensors to P1. The extended replay strategy generalizes the traditional replay strategy, since P2 also uses actuators to maintain the measurements in the steady state. In that way, P2 may decrease the number of components he/she needs to compromise to cover an attack.*

To define the payoff, we introduce a security index δ_{ER}. Particularly, $\delta_{\text{ER}}(X, u_i)$ is equal to the minimum number of components that the attacker has to compromise to conduct a perfectly undetectable extended replay attack against u_i. Following the convention from Chapter 6, we adopt $\delta_{\text{ER}}(X, u_i) = +\infty$ if the attacker cannot gather components that allow him/her to conduct a perfectly undetectable extended replay attack against u_i. The security index δ_{ER} is related to both security indices introduced in the previous chapter. Namely, the problem of computing δ_{ER} can be obtained by adding to the problem of computing δ an additional constraint that imposes the extended replay strategy (see Appendix 7.A). Additionally, Section 7.2 shows that δ_{ER} can be computed and improved in the same way as δ_r.

P1 seeks to make perfectly undetectable extended replay attacks against actuators harder to conduct, which corresponds to maximizing $\delta_{\text{ER}}(X, u_i)$. P2 seeks to con-

duct a perfectly undetectable extended replay attack against an actuator with the minimum effort, which corresponds to minimizing $\delta_{\mathrm{ER}}(X, u_i)$. Thus, we define the payoff as the scaled value of the security index:

$$f(X, u_i) = \varphi\left(\delta_{\mathrm{ER}}(X, u_i)\right).$$

Here, $\varphi : [1, +\infty] \to (0, 1]$ is a known scaling function that is nondecreasing on the interval $[1, +\infty]$. Additionally, we assume that $\varphi(x) = 1$ if and only if $x = +\infty$.

Remark 7.4. *The security index δ_{ER} is not used as the payoff function since it can be equal to $+\infty$. In that case, the expected payoff we introduce in the following would be ill defined.*

Remark 7.5. *The analysis from this chapter is valid for any scaling function with the above-mentioned properties. A concrete example of φ is provided in Section 7.5.*

Mixed strategies

Each player may use mixed strategies, which are probability distributions over the set of pure strategies of that player. We define mixed strategies by

$$\sigma_1 \in \Delta_1, \;\; \Delta_1 = \left\{\sigma_1 \in [0,1]^{|\mathcal{A}_1|} \;\middle|\; \textstyle\sum_{X \in \mathcal{A}_1} \sigma_1(X) = 1\right\},$$
$$\sigma_2 \in \Delta_2, \;\; \Delta_2 = \left\{\sigma_2 \in [0,1]^{|\mathcal{A}_2|} \;\middle|\; \textstyle\sum_{u_i \in \mathcal{A}_2} \sigma_2(u_i) = 1\right\},$$

where σ_1 (resp. σ_2) is a mixed strategy of P1 (resp. P2), and $\sigma_1(X)$ (resp. $\sigma_2(u_i)$) is the probability the strategy X (resp. u_i) is taken.

The expected payoff is given by

$$F(\sigma_1, \sigma_2) = \sum_{X \in \mathcal{A}_1} \sum_{u_i \in \mathcal{A}_2} \sigma_1(X)\sigma_2(u_i) f(X, u_i).$$

We use $F(X, \sigma_2)$ (resp. $F(\sigma_1, u_i)$) to denote the payoff when P1 (resp. P2) plays a pure strategy X (resp. u_i).

Remark 7.6. *One interpretation of σ_1 is that it provides a randomized monitoring strategy. For example, in a day-to-day play in which both players play myopically, P1 selects a sensor placement according to sampling from the probability distribution σ_1. The similar observation holds for σ_2.*

Equilibrium concepts

We focus on NE and ϵ-NE. A strategy profile (σ_1^*, σ_2^*) is a NE if

$$F(\sigma_1^*, \sigma_2) \geq F(\sigma_1^*, \sigma_2^*) \geq F(\sigma_1, \sigma_2^*)$$

holds for all $(\sigma_1, \sigma_2) \in \Delta_1 \times \Delta_2$. Hence, if P2 plays σ_2^*, then P1 cannot perform better than by playing σ_1^*. Similar observation holds for σ_2^*. Other favorable properties of NE strategies are summarized in Section 3.5.

Let $\epsilon \in \mathbb{R}_{\geq 0}$. A strategy profile $(\sigma_1^\epsilon, \sigma_2^\epsilon)$ is an ϵ-NE if

$$F(\sigma_1^\epsilon, \sigma_2) + \epsilon \geq F(\sigma_1^\epsilon, \sigma_2^\epsilon) \geq F(\sigma_1, \sigma_2^\epsilon) - \epsilon$$

holds for all $(\sigma_1, \sigma_2) \in \Delta_1 \times \Delta_2$. Thus, if P2 plays σ_2^ϵ, then P1 may be able to increase the payoff by deviating from σ_1^ϵ. However, not more than ϵ. Thus, σ_1^ϵ is a good approximation of σ_1^* if ϵ is small. Similar observation holds for σ_2^ϵ.

7.1.3 Computing a NE monitoring strategy

We aim computing a NE monitoring strategy, or a good approximation of this strategy, in a tractable manner. Since our game is a finite zero-sum game, a NE monitoring strategy can be computed by solving the following linear program [187]:

Problem 7.1. *Computing a NE monitoring strategy*

$$\begin{aligned} &\underset{z \in \mathbb{R}, \sigma_1 \in \Delta_1}{\text{maximize}} && z \\ &\text{subject to} && F(\sigma_1, u_i) \geq z, \ \forall u_i \in \mathcal{U}. \end{aligned}$$

Unfortunately, Problem 7.1 is challenging to construct and solve when the system is of a large scale. Namely, since the cardinality of $\mathcal{A}_1$ grows combinatorially with respect to n, so does the number of variables of Problem 7.1. Thus, another approach is needed to compute or approximate a NE monitoring strategy in this case.

Sections 7.3 and 7.4 are discussing one possible approach. Particularly, Section 7.3 introduces an ϵ-NE monitoring strategy that can be constructed in a tractable manner, and Section 7.4 discusses the ways to improve this monitoring strategy. Before we move to these sections, we derive the expression for the payoff function based on which we define the above-mentioned ϵ-NE monitoring strategy.

7.2 An analytic expression for the payoff function

To derive the expression for the payoff, we introduce the extended graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E})$ representing the system matrices A, B, C. Same as in the previous chapter, the set of nodes is $\mathcal{V} = \mathcal{X} \cup \mathcal{U} \cup \mathcal{Y} \cup t$, where t models P1. The set of edges is given by $\mathcal{E} = \mathcal{E}_{ux} \cup \mathcal{E}_{xx} \cup \mathcal{E}_{xy} \cup \mathcal{E}_{yt}$, where $\mathcal{E}_{ux} = \{(u_j, x_i) : B(i,j) \neq 0\}$ are the edges from the actuators to the states, $\mathcal{E}_{xx} = \{(x_j, x_i) : A(i,j) \neq 0\}$ are the edges between the states, $\mathcal{E}_{xy} = \{(x_j, y_i) : C(i,j) \neq 0\}$ are the edges from the states to the unprotected sensors, and $\mathcal{E}_{yt} = \{(y_j, t) : y \in \mathcal{Y}\}$ are the edges from the unprotected sensors to t. We also need the following assumption.

Assumption 7.3. *Let A' be obtained from A by setting an arbitrary set of elements of A to zero, and $\mathcal{G}'_t$ be the extended graph defined based on A', B, C. If there exists a directed path from an actuator u_i to a state x_j in $\mathcal{G}'_t$, then the transfer function from u_i to x_j is nonzero.*

We argue that Assumption 7.3 is mild. Let us collect all the nonzero elements of A' in the vector $\lambda \in \mathbb{R}^{n_\lambda}$. If there exists a directed path between u_i and x_j in $\mathcal{G}'_t$, then the transfer function from u_i to x_j is nonzero for almost all vectors $\lambda \in \mathbb{R}^{n_\lambda}$ [209]. That is, the vectors $\lambda \in \mathbb{R}^{n_\lambda}$ for which the transfer function from u_i to x_j equals zero form a set of Lebesgue measure zero.

Next, for every $u_i \in \mathcal{U}$, we define m_i as the size of a minimum vertex separator of u_i and t consisting of: (i) a subset of unprotected sensors; and (ii) a subset of states that are directly controlled by the actuators $\mathcal{U} \setminus u_i$. We also define the set X_i that contains every state x_{i_n} with the following properties: (i) there exists a directed path $u_i, x_{i_0}, x_{i_1}, \ldots, x_{i_n}$; and (ii) the states $x_{i_0}, x_{i_1}, \ldots, x_{i_n}$ are not directly controlled by the actuators $\mathcal{U} \setminus u_i$. The payoff can then be expressed as follows.

Lemma 7.1. *Let Assumptions 7.1–7.3 hold, and let us define $\varphi_i = \varphi(m_i + 1)$. If P2 uses the extended replay attack strategy, then*

$$f(X, u_i) = \begin{cases} \varphi_i, & \text{if } X_i \cap X = \emptyset, \\ 1, & \text{if } X_i \cap X \neq \emptyset. \end{cases} \tag{7.2}$$

Proof. We refer the reader to Appendix 7.B. ■

Put differently, the scaled security index of every actuator u_i prior to the placement of protected sensors equals to φ_i. If P1 measures any state from X_i with a protected sensor, then the scaled index of u_i increases to one. This implies that every extended replay attack against u_i can be detected by the protected sensors. If P1 does not measure states from X_i with protected sensors, then the scaled index of u_i remains equal to φ_i. In the next section, we use this convenient form of the payoff to derive an ϵ-NE. Before we move to the next section, we introduce some remarks.

Remark 7.7. *The payoff* (7.2) *is similar to the one considered in [144]. Yet, an important difference is that the payoff from [144] does not model the fact that network components may be of different importance to the players, while in our case it does. The next section shows that this significantly affects the player's strategies.*

Remark 7.8. *The proof of Lemma 7.1 shows that computing the security index δ_{ER} reduces to computing a minimum vertex separator of u_i and t consisting of a subset of unprotected sensors and a subset of states directly controlled by the actuators $\mathcal{U} \setminus u_i$. Theorem 6.2 establishes the same claim for the robust security index δ_r. Additionally, Lemma 7.1 shows that the index δ_{ER} can be increased in the same way as the robust security index δ_r.*

Remark 7.9. *Both m_i and X_i can be computed efficiently. The problem of computing m_i can be reduced to the minimum cut problem and solved in polynomial time (Section 6.4.1), and X_i can be computed using a breadth first search on $\mathcal{G}_t$ (Section 6.4.3). This implies that: (i) the payoff can be efficiently constructed; and (ii) the attacker that knows the extended graph $\mathcal{G}_t$ can efficiently localize components that enable a perfectly undetectable extended replay attack against u_i.*

7.3 Game analysis

This section introduces an ϵ-NE of the game and discusses its properties. We begin by introducing some preliminaries.

7.3.1 Preliminaries

Firstly, we define a set $U_i = \{u_j \in \mathcal{U} \mid X_j \ni x_i\}$ associated to every state $x_i \in \mathcal{X}$, which we refer to as the monitoring set of x_i. The set U_i contains actuators whose scaled indices become equal to one when we measure x_i with a protected sensor. We denote by $\bar{\varphi}_i$ the minimum scaled security index among the actuators from U_i, and by $U_X = \cup_{x_i \in X} U_i$ the actuators whose security indices are improved by measuring states $X \subseteq \mathcal{X}$ with protected sensors.

Secondly, we introduce set packings and set covers.

Definition 7.1. *A subset of actuators $U \subseteq \mathcal{U}$ is a set packing if $|U \cap U_i| \leq 1$ holds for every $x_i \in \mathcal{X}$. A set packing U is maximal, if adding any actuator from $\mathcal{U} \setminus U$ to U results in a set that is not a set packing. A maximum set packing is a set packing of the maximum cardinality.*

Definition 7.2. *A subset of states $X \subseteq \mathcal{X}$ is a set cover if $U_X = \mathcal{U}$ holds. A set cover X is minimal, if removing any state from X results in a set that is not a set cover. A minimum set cover is a set cover of the minimum cardinality.*

Set packings are of interest for P2. Namely, each of the actuators from a set packing needs a separate protected sensor to improve its index. Thus, P2 can make it challenging for P1 to detect an attack by randomizing the targeted actuators over a set packing. Set covers are of interest for P1. If P1 can form a set cover using n protected sensors, then he/she can improve the indices of all the actuators. As discussed later in this section, the game is then easy to solve. Since we focus on large-scale control systems, a more interesting and relevant situation is one in which P1 cannot improve the indices of all the actuators simultaneously. Hence, unless otherwise stated, we assume that $n < |X|$ holds for any set cover $X \subseteq \mathcal{X}$.

Finally, to characterize monitoring strategies in a convenient manner, we introduce the marginal probability:

$$\rho_{\sigma_1}(x_i) = \sum_{X \in \mathcal{A}_1} \sigma_1(X) \mathbb{1}_{[x_i \in X]}. \tag{7.3}$$

This is the probability that a protected sensor measures x_i when P1 plays σ_1. We point out that σ_1 can be recovered from ρ_{σ_1} systematically by solving a sequence of linear programs (see [144, Section EC.4.]).

7.3.2 An approximate ϵ-NE of the game

We first introduce P1's strategy. Let $X^* = \{x_{i_1}, \ldots, x_{i_{n^*}}\}$ be a minimal set cover, $\bar{\varphi}_{i_1}, \ldots, \bar{\varphi}_{i_{n^*}}$ be the scaled indices associated with the elements of X^*, and $\alpha_1 \leq \ldots \leq \alpha_{n^*}$ be a sorted sequence of these indices. Next, we define the set

$$Z_x(n, X^*) = \left\{ i \in \{1, \ldots, n^*\} \,\middle|\, \alpha_i \leq 1 - \frac{i-n}{\sum_{j=1}^{i} (1-\alpha_j)^{-1}} \right\},$$

which we use to determine the states from X^* that are to be measured by the protected sensors. The following lemma introduces two properties of this set, which we later use in some of the proofs.

Lemma 7.2. *The following statements hold:*

(i) Let $n_1, n_2 \in \mathbb{N}$ and $n_1 < n_2 \leq n^*$. If p_1 is the largest element of $Z_x(n_1, X^*)$ and p_2 is the largest element of $Z_x(n_2, X^*)$, then $p_1 \leq p_2$ holds.

(ii) If p is the largest element of $Z_x(n, X^*)$ and $p \leq n^*$, then $\alpha_p < \alpha_{p+1}$.

Proof. We refer the reader to Appendix 7.C. ■

Let p be the largest element of $Z_x(n, X^*)$. A monitoring strategy σ_1^ϵ of P1 can be characterized as follows:

$$\rho_{\sigma_1^\epsilon}(x_i) = \begin{cases} 1 - \frac{p-n}{S_x(1-\bar{\varphi}_i)}, & \text{if } x_i \in X_p^*, \\ 0, & \text{if } x_i \notin X_p^*, \end{cases} \tag{7.4}$$

where $X_p^* = \{x_i \in X^* \mid \bar{\varphi}_i \leq \alpha_p \}$ and $S_x = \sum_{i=1}^{p} (1-\alpha_i)^{-1}$. Thus, P1 measures only the states X_p^* using the protected sensors, with the marginal probabilities (7.4). Note that Lemma 7.2.(ii) implies that X_p^* contains exactly p elements.

Remark 7.10. *The monitoring strategy σ_1^ϵ is constructed based on the extended graph $\mathcal{G}_t$. This makes the strategy robust with respect to changes in A, B, C.*

We now introduce P2's strategy. Let $U^* = \{u_{j_1}, \ldots, u_{j_{m^*}}\}$ be a maximal set packing, $\varphi_{j_1}, \ldots, \varphi_{j_{m^*}}$ be the scaled indices associated with the elements of U^*, and $\beta_1 \leq \ldots \leq \beta_{m^*}$ be a sorted sequence of these indices. Let us define the set

$$Z_u(n, U^*) = \left\{ i \in \{1, \ldots, m^*\} \,\middle|\, \beta_i \leq 1 - \frac{i-n}{\sum_{j=1}^{i}(1-\beta_j)^{-1}} \right\},$$

which we use to determine the actuators attacked by P2. We remark that the properties of the set $Z_x(n, X^*)$ introduced in Lemma 7.2 also hold for $Z_u(n, U^*)$.

Let q be the largest element of $Z_u(n, U^*)$, $U_q^* = \{u_i \in U^* \mid \varphi_i \leq \beta_q\}$, and $S_u = \sum_{i=1}^{q}(1-\beta_i)^{-1}$. The strategy σ_2^ϵ of P2 is then given by

$$\sigma_2^\epsilon(u_i) = \begin{cases} \frac{1}{S_u(1-\varphi_i)}, & \text{if } u_i \in U_q^*, \\ 0, & \text{if } u_i \notin U_q^*. \end{cases} \tag{7.5}$$

Hence, the strategy of P2 consists of targeting the actuators from U_q^* with the probabilities given by (7.5).

We now show that σ_1^ϵ and σ_2^ϵ form an ϵ-NE of the game, and derive an upper bound for ϵ. We then present several cases when this ϵ-NE becomes an exact NE, outline some game-theoretic intuition, and discuss the general case of the game.

Theorem 7.1. *The following statements hold:*

(i) There exists a strategy profile $(\sigma_1^\epsilon, \sigma_2^\epsilon)$ that satisfies (7.4)–(7.5).

(ii) Any profile $(\sigma_1^\epsilon, \sigma_2^\epsilon)$ that satisfies (7.4)–(7.5) is an ϵ-NE of the game, where

$$\epsilon \leq \frac{p-n}{S_x} - \max\left\{0, \frac{q-n}{S_u}\right\}. \tag{7.6}$$

(iii) $F(\sigma_1^\epsilon, \sigma_2) \geq F_{LB}(n, X^*)$ holds for any $\sigma_2 \in \Delta_2$, where

$$F_{LB}(n, X^*) = 1 - \frac{p-n}{S_x}. \tag{7.7}$$

Proof. We refer the reader to Appendix 7.D. ■

Non-overlapping monitoring sets

Consider the case where the monitoring sets do not overlap. That is, $U_i \cap U_j = \emptyset$ holds for any two states $x_i, x_j \in \mathcal{X}$. We show in the following that this is one of the cases where the previously introduced ϵ-NE becomes exact. We also remark that an example of the system where the monitoring sets do not overlap is a transportation system considered in [37], where $B = I_{n_x}$.

Lemma 7.3. *If $B = I_{n_x}$, then $U_i \cap U_j = \emptyset$ holds for any two states $x_i, x_j \in \mathcal{X}$.*

Proof. Recall that the set X_i contains every state x_{i_n} for which there exists a path $u_i, x_i, x_{i_1}, \ldots, x_{i_n}$ in which the states $x_i, x_{i_1}, \ldots, x_{i_n}$ are not adjacent to actuators $\mathcal{U} \setminus u_i$. Note that the state x_i adjacent to u_i always satisfies this condition. Additionally, since $B = I_{n_x}$, then every state from $\mathcal{X} \setminus x_i$ is adjacent to an actuator from $\mathcal{U} \setminus u_i$. Hence, $X_i = \{x_i\}$ for every u_i. From the latter, it follows that $U_i = \{u_i\}$ for every x_i, and we conclude that the claim holds. ■

Let us define a set $\tilde{X}^* = \{x_{i_1}, \ldots, x_{i_r}\}$ that contains every state $x_i \in \mathcal{X}$ for which U_i is not empty. Since X^* contains all the states whose monitoring sets are not empty, X^* is a set cover. Let us now define $\tilde{U}^* = \{u_1^*, \ldots, u_r^*\}$, where u_j^* is an actuator from a monitoring set U_{i_j} with the scaled index $\bar{\varphi}_{i_j}$. Note that $\tilde{U}^*$ is a maximum set packing, since it contains a single actuator from each of the nonempty monitoring sets. In what follows, we show that the strategies σ_1^ϵ and σ_2^ϵ constructed based on $\tilde{X}^*$ and $\tilde{U}^*$ form a NE.

Corollary 7.1. *Assume that $U_i \cap U_j = \emptyset$ holds for any two states $x_i, x_j \in \mathcal{X}$, and let $\tilde{X}^*$ and $\tilde{U}^*$ be defined as above. Then any strategy profile $(\sigma_1^\epsilon, \sigma_2^\epsilon)$ constructed based on $\tilde{X}^*$, $\tilde{U}^*$, and (7.4)–(7.5) is a NE.*

Proof. Since the actuators from $\tilde{U}^*$ are chosen such as to have the scaled indices $\bar{\varphi}_{i_1}, \ldots, \bar{\varphi}_{i_r}$, we have $\alpha_1 = \beta_1, \ldots, \alpha_r = \beta_r$. Thus, $Z_x(n, \tilde{X}^*) = Z_u(n, \tilde{U}^*)$, and we conclude that $p = q$ and $S_u = S_x$. From the latter and (7.6), we have $\epsilon = 0$. ■

We now discuss Corollary 7.1. Firstly, it follows from (7.4) that P1 measures with the protected sensors only the states from X_p^*. If $p < r$, then the indices of the actuators from $\mathcal{U} \setminus U_{X_p^*}$ are never improved. This is in contrast with Proposition 3 from the related game [144], where it was shown that P1 monitors all of the network components with a nonzero probability in any NE.

Secondly, the proof of Corollary 7.1 shows that $p = q$. Hence, P2 targets the actuators with the p smallest indices from $\tilde{U}^*$. By the construction of $\tilde{U}^*$, these actuators belong to $U_{X_p^*}$. Thus, P2 does not target the actuators $\mathcal{U} \setminus U_{X_p^*}$ whose indices are not improved. Namely, it can be shown that these actuators have the scaled indices lower than the value of the game. Hence, P2 gains more by targeting the actuators $U_{X_p^*}$, even though he/she may get detected by P1.

Finally, it follows from (7.5) that P2 targets the actuators with small indices with low probabilities. On the first glance, this may appear counterintuitive. Yet, we observe from (7.4) that P1 improves actuator indices that are initially small with high probabilities. Hence, P2 targets corresponding actuators with low probabilities to decrease the probability of being detected.

Homogeneous security indices

Assume that the scaled indices are homogeneous, that is, $\varphi_1 = \ldots = \varphi_{n_u} = \tilde{\varphi}$. We show that in this case, the strategies σ_1^ϵ and σ_2^ϵ reduce to those proposed in [144]. We first establish the following auxiliary lemma.

Lemma 7.4. *If $\varphi_1 = \ldots = \varphi_{n_u} = \tilde{\varphi}$ is satisfied, then $p = n^*$, $q = m^*$, $S_x = n^*(1-\tilde{\varphi})^{-1}$, and $S_u = m^*(1-\tilde{\varphi})^{-1}$ hold.*

Proof. We first observe that $n^* \in Z_x(n, X^*)$, since

$$\alpha_{n^*} = \tilde{\varphi} \overset{\tilde{\varphi}\leq 1}{\leq} \tilde{\varphi} + (1-\tilde{\varphi})\frac{n}{n^*} = 1 - \frac{n^* - n}{\sum_{j=1}^{n^*}(1-\tilde{\varphi})^{-1}} = 1 - \frac{n^* - n}{\sum_{j=1}^{n^*}(1-\alpha_j)^{-1}}.$$

Next, note that the largest element of $Z_x(n, X^*)$ cannot be larger than n^*. Therefore, it follows that $p = n^*$. From the latter and the fact that the indices are homogeneous, we have $S_x = n^*(1-\tilde{\varphi})^{-1}$. The same procedure can be used to establish $q = m^*$ and $S_u = m^*(1-\tilde{\varphi})^{-1}$. ■

From (7.4), (7.5), and Lemma 7.4, the strategies σ_1^ϵ and σ_2^ϵ reduce to

$$\rho_{\sigma_1^\epsilon}(x_i) = \begin{cases} \frac{n}{n^*}, & \text{if } x_i \in X^*, \\ 0, & \text{if } x_i \notin X^*, \end{cases} \qquad \sigma_2^\epsilon(u_i) = \begin{cases} \frac{1}{m^*}, & \text{if } u_i \in U^*, \\ 0, & \text{if } u_i \notin U^*. \end{cases} \tag{7.8}$$

In words, P1 measures every state from X^* with the probability n/n^*, while P2 targets every actuator from U^* with the probability $1/m^*$. Interestingly, these are the strategies introduced in the related game [144]. The following corollary investigates performance of these strategies.

Corollary 7.2. *If $\varphi_1 = \ldots = \varphi_{n_u} = \tilde{\varphi}$, then any strategy profile $(\sigma_1^\epsilon, \sigma_2^\epsilon)$ that satisfies* (7.4)–(7.5) *is an ϵ-NE, where*

$$\epsilon \leq (1-\tilde{\varphi})\frac{n^* - n}{n^*} - (1-\tilde{\varphi})\max\left\{0, \frac{m^* - n}{m^*}\right\}. \tag{7.9}$$

Proof. The inequality (7.9) follows directly from (7.6) and Lemma 7.8. ■

Corollary 7.2 has the following consequences. Firstly, it follows from (7.9) that $(\sigma_1^\epsilon, \sigma_2^\epsilon)$ is a NE when $n^* = m^*$. It turns out that n^* and m^* can be equal or close to each other when X^* is a minimum set cover and U^* is a maximum set packing [144]. Hence, the strategies can form a NE or a good ϵ-NE in this case.

Secondly, Corollary 7.2 shows that having heterogeneous security indices associated to the actuators adds an additional layer of complexity to the game. As it can be seen from (7.6), values of the scaled security indices affect ϵ in a non-trivial way in the heterogeneous case. Hence, $(\sigma_1^\epsilon, \sigma_2^\epsilon)$ is generally not a NE even if $n^* = m^*$.

Finally, observe from (7.8) that the set of states measured by the protected sensors is always X^*. Since X^* is a set cover, security indices of all the actuators are improved with nonzero probability regardless of n. Thus, it follows that neglecting security indices may result in excessive spending of the security budget. Namely, if P1 selects the scaling function such as to make all the actuators equally important, then he/she needs to spread the resources to improve the indices of all the actuators. To ensure that the security index of each actuator is improved with sufficiently high probability, P1 needs to make n/n^* sufficiently large by increasing n.

P1 can form a set cover

If $n = n^*$, then it directly follows that $n^* \in Z_x(n, X^*)$ and $p = n^* = n$. From the latter and (7.4), it follows that the strategy σ_1^ϵ reduces to measuring every state of a set cover X^* with probability one. In words, P1 plays a pure strategy X^* in this case. Additionally, from $p = n$ and (7.7), we have $F_{\mathrm{LB}}(n, X^*) = 1$. It then directly follows that $(\sigma_1^\epsilon, \sigma_2^\epsilon)$ is a NE.

Corollary 7.3. *If $n = n^*$, then $(\sigma_1^\epsilon, \sigma_2^\epsilon)$ is a NE.*

Proof. Since $F_{\mathrm{LB}}(n, X^*) = 1$ and the maximum value of the expected payoff is one, P1 cannot increase its payoff. Additionally, if P1 plays σ_1^ϵ, then P2 cannot decrease its payoff by deviating from σ_2^ϵ. Thus, the claim holds. ■

General case

Besides the above-mentioned cases, $(\sigma_1^\epsilon, \sigma_2^\epsilon)$ is a NE every time $(p - n)/S_x$ equals $\max\{0, q - n\}/S_u$. Additionally, the monitoring strategy σ_1^ϵ guarantees that the payoff is at least equal to $F_{\mathrm{LB}}(n, X^*)$. If $F_{\mathrm{LB}}(n, X^*)$ is sufficiently large, then P1 can adopt σ_1^ϵ in spite of its suboptimality. If $F_{\mathrm{LB}}(n, X^*)$ is small, then P1 can try to improve σ_1^ϵ and $F_{\mathrm{LB}}(n, X^*)$ in several ways. This problem is discussed next.

7.4 Improving the monitoring strategy σ_1^ϵ

This section discusses three approaches for improving the monitoring strategy σ_1^ϵ.

7.4.1 Approach 1: Increasing the number of protected sensors

The following proposition shows that $F_{\mathrm{LB}}(n, X^*)$ is increasing with n. Hence, a simple way to improve the worst-case guarantees on the payoff of σ_1^ϵ is by increasing a number of protected sensors. Particularly, if $n = n^*$, then $F_{\mathrm{LB}}(n, X^*) = 1$, in which case σ_1^ϵ becomes a NE monitoring strategy.

Proposition 7.1. *Let $n_1, n_2 \in \mathbb{N}$. If $n_1 < n_2 \leq n^*$, then $F_{LB}(n_1, X^*) < F_{LB}(n_2, X^*)$.*

Proof. Let p_1 (resp. p_2) be the largest element of $Z_x(n_1, X^*)$ (resp. $Z_x(n_2, X^*)$). Recall from Lemma 7.2 that $p_1 \leq p_2$ holds. If $p_1 < p_2$, then

$$F_{\mathrm{LB}}(n_1, X^*) \overset{(7.22)}{<} \alpha_{p_1+1} \overset{p_1<p_2}{\leq} \alpha_{p_2} \overset{(*)}{\leq} 1 - \frac{p_2 - n_2}{\sum_{i=1}^{p_2}(1-\alpha_i)^{-1}} \overset{(7.7)}{=} F_{\mathrm{LB}}(n_2, X^*),$$

where (*) follows from the fact that p_2 belongs to $Z_x(n_2, X^*)$. If $p_2 = p_1 = p$, then

$$F_{\mathrm{LB}}(n_1, X^*) \overset{(7.7)}{=} 1 - \frac{p - n_1}{\sum_{i=1}^{p}(1-\alpha_i)^{-1}} \overset{n_1<n_2}{<} 1 - \frac{p - n_2}{\sum_{i=1}^{p}(1-\alpha_i)^{-1}} = F_{\mathrm{LB}}(n_2, X^*),$$

which concludes the proof. ■

7.4.2 Approach 2: Focusing on the most vulnerable actuators

We now discuss an approach that can be used when P1 is low in resources. Let us denote by $U_{\min}$ the set of actuators that have the scaled index $\varphi_{\min}$ smaller than all the other actuators. Let $\bar{X}^*$ be a minimum set cover of the actuators $U_{\min}$, $\bar{U}^*$ be a maximum set packing of $U_{\min}$, $\bar{n}^* = |\bar{X}^*|$, and $\bar{m}^* = |\bar{U}^*|$. Consider the strategies characterized by

$$\rho_{\bar{\sigma}_1^\epsilon}(x_i) = \begin{cases} \frac{n}{\bar{n}^*}, & x_i \in \bar{X}^*, \\ 0, & x_i \notin \bar{X}^*, \end{cases} \qquad \bar{\sigma}_2^\epsilon(u_i) = \begin{cases} \frac{1}{\bar{m}^*}, & u_i \in \bar{U}^*, \\ 0, & u_i \notin \bar{U}^*. \end{cases} \tag{7.10}$$

Put differently, P1 focuses on improving the indices of the actuators $U_{\min}$, while P2 focuses on targeting the actuators $\bar{U}^* \subseteq U_{\min}$. These strategies can be seen as a modification of the strategies σ_1^ϵ and σ_2^ϵ, where the players focus on the subset of the actuators with the smallest security indices.

Let us define $\bar{\varphi}_{\min} = \min_{u_i \in \mathcal{U} \setminus U_{\min}} \varphi_i$, and assume that

$$\bar{\varphi}_{\min} > \varphi_{\min} + (1 - \varphi_{\min})\frac{n}{\bar{n}^*}. \tag{7.11}$$

Observe that the condition (7.11) holds when P1 is low in resources. Indeed, if $n/\bar{n}^* \approx 0$, then (7.11) reduces to $\bar{\varphi}_{\min} > \varphi_{\min}$. The following proposition investigates the strategies $\bar{\sigma}_1^\epsilon$ and $\bar{\sigma}_2^\epsilon$ assuming that the condition (7.11) holds.

Proposition 7.2. *Assume that the condition* (7.11) *is satisfied. The following statements then hold:*

(i) There exists a strategy profile $(\bar{\sigma}_1^\epsilon, \bar{\sigma}_2^\epsilon)$ that satisfies (7.10)*.*

(ii) Any profile $(\bar{\sigma}_1^\epsilon, \bar{\sigma}_2^\epsilon)$ *that satisfies* (7.10) *is an* $\bar{\epsilon}$*-NE of the game, where*

$$\bar{\epsilon} \leq (1-\varphi_{\min})\frac{\bar{n}^* - n}{\bar{n}^*} - (1-\varphi_{\min})\max\left\{0, \frac{\bar{m}^* - n}{\bar{m}^*}\right\}. \tag{7.12}$$

(iii) $F(\bar{\sigma}_1^\epsilon, \sigma_2) \geq \bar{F}_{LB}(n)$ *holds for any* $\sigma_2 \in \Delta_2$*, where*

$$\bar{F}_{LB}(n) = 1 - (1-\varphi_{\min})\frac{\bar{n}^* - n}{\bar{n}^*}. \tag{7.13}$$

(iv) $\bar{F}_{LB}(n) \geq F_{LB}(n, X^*)$ *for any minimal set cover* X^* *of actuators* $\mathcal{U}$*.*

Proof. We refer the reader to Appendix 7.E. ■

Proposition 7.2 shows that the worst-case guarantees on the payoff of the monitoring strategy $\bar{\sigma}_1^\epsilon$ cannot be lower than those for σ_1^ϵ once (7.11) holds (Statement (iv)). If in addition $\bar{n}^* = \bar{m}^*$, then $\bar{\sigma}_1^\epsilon$ is a NE monitoring strategy (Statement (ii)).

7.4.3 Approach 3: Column Generation Procedure

CGP can be used to solve linear programs with a large number of decision variables and a relatively small number of constraints [210]. Since Problem 7.1 satisfies these properties, we can use CGP to tackle it. The first step is to solve a master problem of Problem 7.1, which can be formulated as

$$\begin{array}{ll} \underset{\tilde{z}\geq 0, \tilde{\sigma}_1 \geq 0}{\text{maximize}} & \tilde{z} \\ \text{subject to} & \sum_{X\in\tilde{\mathcal{A}}_1} F_X \tilde{\sigma}_1(X) \geq \tilde{z} 1_{n_u}, \ \sum_{X\in\tilde{\mathcal{A}}_1} \tilde{\sigma}_1(X) = 1, \end{array} \tag{7.14}$$

where $F_X \in \mathbb{R}^{n_u}$ is the column vector defined by

$$F_{Xi} = \begin{cases} \varphi_i, & \text{if } u_i \notin U_X, \\ 1, & \text{if } u_i \in U_X. \end{cases} \tag{7.15}$$

In words, the master problem (7.14) is obtained from Problem 7.1 by considering only a subset of pure strategies $\tilde{\mathcal{A}}_1$ instead of the whole set $\mathcal{A}_1$. To improve the strategy σ_1^ϵ, one should initialize $\tilde{\mathcal{A}}_1$ with pure strategies that are played with nonzero probability under σ_1^ϵ. However, we stress that one can also initialize $\tilde{\mathcal{A}}_1$ arbitrarily, and try to compute a NE monitoring strategy directly.

Let $(\tilde{z}^*, \tilde{\sigma}_1^*)$ be a solution of (7.14). The next step is to check if the optimal value $\tilde{z}^*$ of (7.14) can be further improved. This can be done by solving the subproblem

$$\text{maximize}_{X\in\mathcal{A}_1} \ (\rho^*)^T F_X - \pi^*, \tag{7.16}$$

where $\rho^* \in \mathbb{R}^{n_u}$ is an optimal dual solution of (7.14) that corresponds to the inequality constraints, and $\pi^* \in \mathbb{R}$ is an optimal dual solution of (7.14) that corresponds to the equality constraint. We stress that the elements of $\tilde{\mathcal{A}}_1$ cannot be a solution of the subproblem [210]. Hence, CGP is guaranteed to converge.

Let the optimal value of (7.16) be $\tilde{c}$. If $\tilde{c} > 0$, then $\tilde{z}^*$ can be improved. A solution of (7.16) is then added to $\tilde{\mathcal{A}}_1$, and the procedure is repeated with the new set $\tilde{\mathcal{A}}_1$. If $\tilde{c} \leq 0$, then $\tilde{z}^*$ is the optimal value and $\tilde{\sigma}_1^*$ is a solution of Problem 7.1. The procedure then terminates.

The crucial point of CGP is to solve the subproblem (7.16) in a scalable manner, which is not always possible. However, we show that a solution of the subproblem in our case can be computed by solving the following integer linear program:

$$\begin{aligned} \underset{p,q}{\text{maximize}} \quad & \sum_{i=1}^{n_u} \rho_i^* \big(\varphi_i(1-q_i) + q_i\big) - \pi^* \\ \text{subject to} \quad & \sum_{i=1}^{n_x} \mathbb{1}_{[u_j \in U_i]}\, p_i \geq q_j,\ \forall j \in \{1,\dots,n_u\}, \\ & \sum_{i=1}^{n_x} p_i \leq n,\ p \in \{0,1\}^{n_x},\ q \in \{0,1\}^{n_u}. \end{aligned} \tag{7.17}$$

An important observation is that the problem (7.17) has $n + n_u$ decision variables, and $n_u + 1$ constraints. This implies that the size of this problem does not grow combinatorially with n. In the next section, we show that this allows us to use CGP even when the system is of a large scale. Before we proceed, we explain in the next proposition how a solution and the optimal value of the problem (7.17) can be transformed to a solution and the optimal value of the problem (7.16).

Proposition 7.3. *If $\tilde{c}$ is the optimal value and $(\tilde{p}, \tilde{q})$ is a solution of* (7.17)*, then $\tilde{c}$ is the optimal value and $\tilde{X} = \{x_i \in \mathcal{X} \mid \tilde{p}_i = 1\}$ is a solution of* (7.16)*.*

Proof. We refer the reader to Appendix 7.F. ■

7.5 Illustrative examples

This section gives an example of the scaling function, shows that σ_1^ϵ can be constructed in a scalable manner in large-scale systems, investigates optimality of σ_1^ϵ, and tests CGP. The experiments are performed on Intel Core i7-8650U computer.

Table 7.1: The table contains five categories of the security level, and explains how to determine to which category an actuator belongs based on its security index.

Security level (qualitative scale)	Security level (quantitative scale)	Assigning security level to actuators
Very low	0.2	$\delta_{\text{ER}}(X, u_i) \leq 5$
Low	0.4	$5 < \delta_{\text{ER}}(X, u_i) \leq 15$
Moderate	0.6	$15 < \delta_{\text{ER}}(X, u_i) \leq 20$
High	0.8	$20 < \delta_{\text{ER}}(X, u_i) < +\infty$
Very high	1.0	$\delta_{\text{ER}}(X, u_i) = +\infty$

7.5.1 Model: Power grid

We consider the IEEE 2383 bus power grid (3037 states and 327 generators). We model the system using linearized swing equations, where the generators are represented by two states (rotor angle and frequency), and load buses with one state (voltage angle) [204]. We assume that all the states are measurable, and that the attacker can conduct an attack using some of the loads [206]. We randomly select 30% of the loads to be attackable.

7.5.2 Example 1: Scaling function

We now explain a possible way to form the scaling function φ. Firstly, we define five security levels according to Table 7.1, Column 1. The qualitative scale we adopt was used in [17] to characterize values of several security metrics. Secondly, since the scaling function takes values from the set of real numbers, we need to assign quantitative values to each of the levels. We adopt the values from Table 7.1, Column 2. Other ways to transform the qualitative scale to quantitative scale can also be used [17]. Thirdly, we use the actuator security index δ_{ER} to determine a security level of an actuator (Table 7.1, Column 3). In summary, the scaling function maps δ_{ER} to the security levels, and can be defined as follows:

$$\varphi(x) = 0.2 \cdot \mathbb{1}_{[x \leq 5]} + 0.4 \cdot \mathbb{1}_{[5 < x \leq 15]} + 0.6 \cdot \mathbb{1}_{[15 < x \leq 20]} + 0.8 \cdot \mathbb{1}_{[20 < x < +\infty]} + 1 \cdot \mathbb{1}_{[x = +\infty]}.$$

7.5.3 Example 2: Comparing monitoring strategies

We now construct and compare two monitoring strategies: (i) a monitoring strategy σ_1^ϵ; and (ii) a NE monitoring strategy σ_1^* using CGP. To construct σ_1^ϵ, we: (i) compute a minimum set cover using an integer linear program solver that is included in the Matlab package; (ii) compute the marginal probabilities $\rho_{\sigma_1^\epsilon}$ according to (7.4); and (iii) use the procedure from [144, Section EC.4.] to obtain σ_1^ϵ from $\rho_{\sigma_1^\epsilon}$. The strategy σ_1^* is constructed using CGP, as explained in Section 7.4.3.

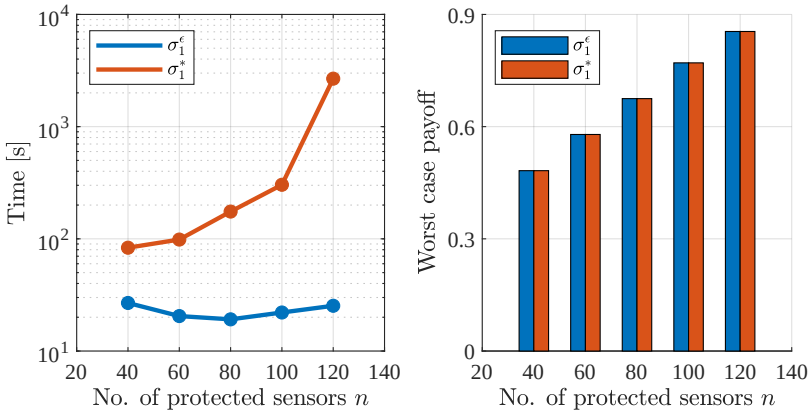

Figure 7.2: The times needed to construct the monitoring strategies σ_1^ϵ and σ_1^*, and the worst-case payoffs of these strategies.

The plots of the execution times and the worst-case payoffs with respect to the number of protected sensors are shown in Figure 7.2. Observe that the strategies performed equally well in the terms of the worst-case payoff. In words, the strategy σ_1^ϵ is a NE monitoring strategy in this case. Moreover, it can be seen that the time needed to construct σ_1^ϵ is not significantly affected by the number of deployed sensors. Particularly, it remains bellow 30 seconds in all the cases. Thus, σ_1^ϵ can also be efficiently constructed.

Although we are able to construct σ_1^* using CGP, the time required to construct this strategy is rapidly increasing with the number of deployed sensors. The maximum time is reached in the case of 120 deployed sensors, and it is equal to 44.61 minutes. This indicates that CGP may become restrictive to use if the network size and/or the number of protected sensors exceeds several thousand. In that case, we can first try to construct σ_1^ϵ, and then run a limited number of iterations of CGP to improve this strategy (if needed).

7.6 Summary

This chapter investigated an operator-attacker security game. The objective behind studying this game was to compute a mixed monitoring strategy for improving the actuator indices. Since the problem of computing a NE monitoring strategy was difficult to construct and solve for large-scale systems, we derived an ϵ-NE of the game. The monitoring strategy from this ϵ-NE can be constructed in a scalable manner and further improved by deploying additional sensors, focusing on the most vulnerable actuators, and using CGP. We also presented several situations in which

the ϵ-NE becomes exact, and outlined game-theoretic interpretations behind this equilibrium. Finally, we demonstrated in simulations that the approaches we proposed can be used to compute NE monitoring strategies in a large-scale system. We now move to the next chapter, in which we conclude the thesis.

Appendix to Chapter 7

7.A The security index $\delta_{\mathbf{ER}}$

Let us first define the protected sensor measurements by

$$y_p(k) = C_X x(k).$$

Here, $C_X \in \mathbb{R}^{n \times n_x}$ depends on the states $X \subseteq \mathcal{X}$ measured by the protected sensors. Particularly, if the states $X = \{x_{i_1}, \dots, x_{i_n}\}$ are measured, then $C_X = [e_{i_1} \dots e_{i_n}]^T$. The problem of computing $\delta_{\mathrm{ER}}(X, u_i)$ can then be formulated as follows:

$$\begin{aligned}
\underset{a_u, a_y}{\text{minimize}} \quad & \|a_u\|_0 + \|a_y\|_0 \\
\text{subject to} \quad & x(k+1) = Ax(k) + Ba_u(k), && \text{(C1)} \\
& y(k) = Cx(k) + a_y(k), && \text{(C2)} \\
& y_p(k) = C_X x(k), && \text{(C3)} \\
& y \equiv 0,\ y_p \equiv 0,\ x(0) = 0_{n_x}, && \text{(C4)} \\
& a_u, a_y \text{ form an extended replay attack against } u_i. && \text{(C5)}
\end{aligned}$$

The objective function reflects the attacker's desire to find the minimum number of components to conduct an attack. Constraints (C1)–(C3) ensure that the attack satisfies the system dynamics, (C4) imposes that the attack is perfectly undetectable, and (C5) ensures that the attack follows the extended replay strategy.

7.B Proof of Lemma 7.1

We first introduce an auxiliary lemma.

Lemma 7.5. *Let Assumptions 7.1–7.3 hold, U_a be attacked actuators, Y_a be attacked sensors, $u_i \in U_a$, and*

$$X_a = \{x_j \in \mathcal{X} : (u_k, x_j) \in \mathcal{E}_{ux}, u_k \in U_a \setminus u_i\}. \tag{7.18}$$

Additionally, assume that protected sensors are not present in the system. P2 can conduct a perfectly undetectable extended replay attack against u_i if and only if $X_a \cup Y_a$ is a vertex separator of u_i and t in $\mathcal{G}_t$.

Proof. ($\Rightarrow$) The proof is by contradiction. Assume that $X_a \cup Y_a$ is not a vertex separator of u_i and t in $\mathcal{G}_t$. Then there exists a directed path from u_i to at least one non-attacked sensor y_j (Path 1), which is not intersected with $X_a \cup Y_a$. Let x_p be the state that is directly measured by y_j. We show that any extended replay attack against u_i affects x_p, and therefore, is visible in y_j.

First, observe that under the extended replay attack, any state from X_a remains equal to zero for any $k \in \mathbb{Z}_{\geq 0}$. Hence, the states of the system (7.1) under the extended replay attack propagate according to

$$x(k+1) = A'x(k) + B(:,i)a_{ui}(k),$$

where $A'(q,r) = A(q,r)\mathbb{1}_{[x_q \notin X_a]}$ and $x(0) = 0_{n_x}$. Next, let $\mathcal{G}'_t$ be the extended graph defined based on A',B,C. Note that $\mathcal{G}'_t$ can be obtained from $\mathcal{G}_t$ by deleting the edges from $\mathcal{E}_{xx}$ that end in the states from X_a. Since Path 1 does not contain the nodes from X_a, then Path 1 exists in $\mathcal{G}'_t$ as well. It then follows from Assumption 7.3 that the transfer function from u_i to x_p is nonzero. Hence, extended replay attacks against u_i are not perfectly undetectable, since they are visible from the non-attacked sensor y_j that directly measures x_p.

($\Leftarrow$) In the proof of Theorem 6.2, we used the extended replay strategy to prove that the attacker can conduct a perfectly undetectable attack against u_i in any system realization when $X_a \cup Y_a$ is a vertex separator of u_i and t. Hence, we can follow the steps from the proof of Theorem 6.2 to complete this part of the proof. ■

From Lemma 7.5, it follows that prior to the placement of protected sensors, the minimum number of sensors and actuators needed to conduct a perfectly undetectable extended replay attack against u_i equals $m_i + 1$. Namely, m_i is the size of a minimum vertex separator of u_i and t consisting of: (i) a subset of unprotected sensors; and (ii) a subset of states which are directly controlled by the actuators $\mathcal{U} \setminus u_i$. Additionally, we add one to m_i to account for u_i. Hence, the scaled index of u_i prior to the placement of protected sensors equals to φ_i.

To complete the proof of Lemma 7.1, it suffices to show that: (i) the scaled index of u_i increases to one when we place a protected sensor to measure a state from X_i ($\delta_{\text{ER}}(X, u_i)$ becomes equal to $+\infty$); and (ii) the scaled index of u_i remains the same when we place protected sensors to measure states not contained in X_i. This is equivalent to showing that: (i) there does not exists a vertex separator of u_i and t when we place a protected sensor to measure a state from X_i; and (ii) the size of a minimum vertex separator of u_i and t does not change when we place protected sensors to measure states outside X_i. To show these claims, we can use the same procedure as in the proof of Theorem 6.3.

7.C Proof of Lemma 7.2

Proof of (i): Observe that $n_1 \in Z_x(n_1, X^*)$ and $n_2 \in Z(n_2, X^*)$, so both p_1 and p_2 exist. Next, we have

$$\alpha_{p_1} \overset{(*)}{\leq} 1 - \frac{p_1 - n_1}{\sum_{i=1}^{p_1}(1-\alpha_i)^{-1}} \overset{n_1 \leq n_2}{<} 1 - \frac{p_1 - n_2}{\sum_{i=1}^{p_1}(1-\alpha_i)^{-1}},$$

where (*) holds since $p_1 \in Z_x(n_1, X^*)$. Hence, it follows that $p_1 \in Z_x(n_2, X^*)$. Since p_2 is the largest element of $Z_x(n_2, X^*)$, we conclude that $p_2 \geq p_1$ holds.

Proof of (ii): The proof is by contradiction. Assume that $\alpha_p = \alpha_{p+1}$. We show that this implies that $p+1 \in Z_x(n, X^*)$. If $p+1 \in Z_x(n, X^*)$, then

$$\alpha_{p+1} \leq 1 - \frac{p+1-n}{\sum_{i=1}^{p}(1-\alpha_i)^{-1} + (1-\alpha_{p+1})^{-1}}$$

has to hold. By multiplying both sides by $\sum_{i=1}^{p}(1-\alpha_i)^{-1} + (1-\alpha_{p+1})^{-1}$ and rearranging the terms, we obtain

$$\alpha_{p+1} \leq 1 - \frac{p-n}{\sum_{i=1}^{p}(1-\alpha_i)^{-1}}.$$

Since $\alpha_{p+1} = \alpha_p$ and $p \in Z_x(n, X^*)$, the last inequality holds. Thus, $p+1 \in Z_x(n, X^*)$, which is inconsistent with the fact that p is the largest element of $Z_x(n, X^*)$. Thus, $\alpha_p < \alpha_{p+1}$ has to hold.

7.D Proof of Theorem 7.1

Proof of (i): To prove existence of σ_1^ϵ, we prove that $\rho_{\sigma_1^\epsilon}(x_i) \in [0,1]$ for any $x_i \in \mathcal{X}$ and $\sum_{x_i \in \mathcal{X}} \rho_{\sigma_1^\epsilon}(x_i) = n$. It then follows from Farkas' lemma that $\sigma_1^\epsilon \in \Delta_1$ (for instance, see [144, Lemma EC.6]).

Note that $n \in Z_x(n, X^*)$, so $p \geq n$. Hence, from (7.4), we have $\rho_{\sigma_1^\epsilon}(x_i) \leq 1$ for any $x_i \in \mathcal{X}$. From $\alpha_1 \leq \ldots \leq \alpha_p$, it follows that

$$\frac{p-n}{(1-\alpha_1)S_x} \leq \ldots \leq \frac{p-n}{(1-\alpha_p)S_x}. \tag{7.19}$$

From the fact that $p \in Z_x(n, X^*)$, we have

$$\alpha_p \leq 1 - \frac{p-n}{S_x} \Longrightarrow \frac{p-n}{S_x} \leq 1 - \alpha_p \Longrightarrow \frac{p-n}{(1-\alpha_p)S_x} \leq 1.$$

From the latter, (7.19), and (7.4), we conclude that $0 \leq \rho_{\sigma_1^\epsilon}(x_i)$ holds for any $x_i \in \mathcal{X}$. Thus, $\rho_{\sigma_1^\epsilon}(x_i) \in [0,1]$. Additionally, we have

$$\sum_{x_i \in \mathcal{X}} \rho_{\sigma_1^\epsilon}(x_i) \overset{(7.4)}{=} \sum_{x_i \in X_p^*} \left(1 - \frac{p-n}{S_x(1-\bar{\varphi}_i)}\right) = p - \sum_{i=1}^{p} \frac{p-n}{S_x(1-\alpha_i)} = p - \frac{p-n}{S_x} S_x = n.$$

Therefore, we conclude that $\sigma_1^\epsilon \in \Delta_1$.

We now prove that $\sigma_2^\epsilon \in \Delta_2$. From (7.5), $0 \leq \sigma_2^\epsilon(u_i)$ holds for any $u_i \in \mathcal{U}$. Additionally, by the definition of S_u , we have $S_u(1-\varphi_i) \geq 1$ for any $u_i \in \mathcal{U}_q^*$. It then follows that $\sigma_2^\epsilon(u_i) \leq 1$ holds for any $u_i \in \mathcal{U}$. Finally, we have

$$\sum_{u_i \in \mathcal{U}} \sigma_2^\epsilon(u_i) \overset{(7.5)}{=} \sum_{u_i \in U_q^*} \frac{1}{S_u(1-\varphi_i)} = \sum_{i=1}^{q} \frac{1}{S_u(1-\beta_i)} = \frac{1}{S_u} S_u = 1,$$

so we conclude that $\sigma_2^\epsilon \in \Delta_2$.

Proof of (ii): Let $(\sigma_1^\epsilon, \sigma_2^\epsilon)$ be a strategy profile that satisfies (7.4) and (7.5). We first establish a lower bound on P1's payoff when he/she plays σ_1^ϵ. Consider an actuator u_l from a set U_i, where $x_i \in X_p^*$. Let $\mathbb{P}_{\sigma_1}(\varphi_l \not\to 1)$ be the probability that the security index of u_l is not improved when P1 plays σ_1. We then have

$$\begin{aligned}
F(\sigma_1^\epsilon, u_l) &= \sum_{X \in \mathcal{A}_1} \sigma_1^\epsilon(X) \left(\mathbb{1}_{[X \cap X_l \neq \emptyset]} + \varphi_l \mathbb{1}_{[X \cap X_l = \emptyset]} \right) \\
&= 1 - \mathbb{P}_{\sigma_1^\epsilon}(\varphi_l \not\to 1) + \varphi_l \mathbb{P}_{\sigma_1^\epsilon}(\varphi_l \not\to 1) \\
&= 1 - (1-\varphi_l)\mathbb{P}_{\sigma_1^\epsilon}(\varphi_l \not\to 1) \\
&\overset{(*)}{\geq} 1 - (1-\varphi_l)(1-\rho_{\sigma_1^\epsilon}(x_i)) \\
&\overset{(7.4)}{=} 1 - (1-\varphi_l)\frac{p-n}{S_x(1-\bar{\varphi}_i)} \\
&\overset{(**)}{\geq} 1 - \frac{p-n}{S_x} = F_{\mathrm{LB}}(n, X^*).
\end{aligned} \tag{7.20}$$

Here, (*) follows from the fact that u_l belongs to U_i. Thus, the security index of u_l is improved every time we place a sensor at x_i, so $\mathbb{P}_{\sigma_1^\epsilon}(\varphi_l \not\to 1) \leq 1 - \rho_{\sigma_1^\epsilon}(x_i)$ holds. Additionally, (**) follows from the fact that $\bar{\varphi}_i$ is the smallest scaled security index within U_i. Since u_l was arbitrarily selected, we have that $F(\sigma_1^\epsilon, u_j) \geq F_{\mathrm{LB}}(n, X^*)$ for any actuator u_j whose index is improved by measuring states X_p^*. Hence, $F_{\mathrm{LB}}(n, X^*)$ lower bounds P1's payoff if $p = n^*$.

Next, we show that $F_{\mathrm{LB}}(n, X^*)$ lower bounds P1's payoff in the case $p < n^*$. Since $p+1 \notin Z_x(n, X^*)$, it follows from the definitions of $Z_x(n, X^*)$ and S_x that

$$\alpha_{p+1} > 1 - \frac{p+1-n}{S_x + (1-\alpha_{p+1})^{-1}}. \tag{7.21}$$

By multiplying both sides by $S_x + (1-\alpha_{p+1})^{-1}$ and rearranging the terms, we obtain that (7.21) is equivalent to

$$\alpha_{p+1} > 1 - \frac{p-n}{S_x} = F_{\mathrm{LB}}(n, X^*). \tag{7.22}$$

Next, observe that all the actuators with the scaled indices smaller than α_{p+1} belong to $U_{X_p^*}$. Thus, the minimum scaled security index among the actuators whose indices are not improved cannot be smaller than α_{p+1}. Hence, by targeting an actuator whose index is not improved, the lowest value of the payoff that P2 can achieve is α_{p+1}. This corresponds to the case when the scaled index α_{p+1} of an actuator cannot be improved by measuring the states X_p^*. Therefore, $F_{\mathrm{LB}}(n, X^*)$ lower bounds the operator's payoff in this case as well.

We now derive an upper bound on P2's payoff when he/she plays σ_2^ϵ. For any $X \in \mathcal{A}_1$, we have

$$\begin{aligned}
F(X, \sigma_2^\epsilon) &= \sum_{u_i \in U_q^*} \sigma_2^\epsilon(u_i) \left(\mathbb{1}_{[U_X \ni u_i]} + \varphi_i \mathbb{1}_{[U_X \not\ni u_i]} \right) \\
&= \sum_{u_i \in U_q^*} \sigma_2^\epsilon(u_i) \left(\mathbb{1}_{[U_X \ni u_i]} + \mathbb{1}_{[U_X \not\ni u_i]} - \mathbb{1}_{[U_X \not\ni u_i]} + \varphi_i \mathbb{1}_{[U_X \not\ni u_i]} \right) \\
&= 1 - \sum_{u_i \in U_q^*} \sigma_2^*(u_i)(1 - \varphi_i) \mathbb{1}_{[U_X \not\ni u_i]} \\
&\overset{(7.5)}{=} 1 - \sum_{u_i \in U_q^*} \frac{1}{S_u} \mathbb{1}_{[U_X \not\ni u_i]} \\
&\overset{(*)}{\leq} 1 - \max\left\{0, \frac{q-n}{S_u}\right\} = F_{\mathrm{UB}}(n, U^*),
\end{aligned} \tag{7.23}$$

where (*) holds since U_q^* is a set packing, so the indices of at most n actuators can be improved by the placement X.

Therefore, we conclude that $(\sigma_1^\epsilon, \sigma_2^\epsilon)$ is an ϵ-NE, where

$$\epsilon \leq F_{\mathrm{UB}}(n, U^*) - F_{\mathrm{LB}}(n, X^*) = \frac{p-n}{S_x} - \max\left\{0, \frac{q-n}{S_u}\right\}.$$

Proof of (iii): Follows from (7.20) and (7.22).

7.E Proof of Proposition 7.2

Proof of (i): We first show that if the condition (7.11) holds, then $n \leq \bar{n}^*$. Assume that $n > \bar{n}^*$. We then have

$$\bar{\varphi}_{\min} \overset{(7.11)}{>} \varphi_{\min} + (1 - \varphi_{\min}) \frac{n}{\bar{n}^*} \overset{n > \bar{n}^*, \varphi_{\min} < 1}{\geq} \varphi_{\min} + 1 - \varphi_{\min} = 1.$$

From the later, it follows that $\bar{\varphi}_{\min} > 1$, which cannot hold due to the properties of the scaling function φ.

Therefore, $\rho_{\bar{\sigma}_1^\epsilon}(x_i) \in [0,1]$ for any $x_i \in \mathcal{X}$ and $\sum_{x_i \in \mathcal{X}} \rho_{\bar{\sigma}_1^\epsilon}(x_i) = n$. From the latter, it follows that there exists $\bar{\sigma}_1^\epsilon$ that satisfies (7.10) (for instance, see [144, Lemma EC.6]). It is also easy to see that $\bar{\sigma}_2^\epsilon(u_i) \in [0,1]$ for any $u_i \in \mathcal{U}$ and $\sum_{u_i \in \mathcal{U}} \bar{\sigma}_2^\epsilon(u_i) = 1$. Hence, Statement (i) holds.

Proof of (ii): Let P1 plays $\bar{\sigma}_1^\epsilon$, $u_i \in U_{\min}$, and $\mathbb{P}_{\bar{\sigma}_1^\epsilon}(\varphi_i \not\to 1)$ be the probability that the security index of u_i is not improved when P1 plays $\bar{\sigma}_1^\epsilon$. We then have

$$\begin{aligned} F(\bar{\sigma}_1^\epsilon, u_i) &= \sum_{X \in \mathcal{A}_1} \bar{\sigma}_1^\epsilon(X) \left(\mathbb{1}_{[X \cap X_i \neq \emptyset]} + \varphi_{\min} \mathbb{1}_{[X \cap X_i = \emptyset]} \right) \\ &= 1 - \mathbb{P}_{\bar{\sigma}_1^\epsilon}(\varphi_i \not\to 1) + \varphi_{\min} \mathbb{P}_{\bar{\sigma}_1^\epsilon}(\varphi_i \not\to 1) \\ &= 1 - (1 - \varphi_{\min}) \mathbb{P}_{\bar{\sigma}_1^\epsilon}(\varphi_i \not\to 1) \\ &\overset{(*)}{\geq} 1 - (1 - \varphi_{\min}) \frac{\bar{n}^* - n}{\bar{n}^*} = \bar{F}_{\mathrm{LB}}(n). \end{aligned} \tag{7.24}$$

Here, (*) follows from the fact that u_i belongs to $U_{\min}$. Therefore, the security index of u_i is improved with probability that is greater than or equal to $n/\bar{n}^*$. Hence, $\mathbb{P}_{\bar{\sigma}_1^\epsilon}(\varphi_i \not\to 1) \leq 1 - n/n^*$ holds. If P2 attacks $u_i \notin U_{\min}$, then

$$F(\bar{\sigma}_1^\epsilon, u_i) \overset{(*)}{\geq} \bar{\varphi}_{\min} \overset{(7.11)}{>} \varphi_{\min} + (1 - \varphi_{\min}) \frac{n}{\bar{n}^*} = \bar{F}_{\mathrm{LB}}(n), \tag{7.25}$$

where (*) holds since the lowest payoff occurs if the scaled index of u_i is not improved and equals $\bar{\varphi}_{\min}$. Thus, $\bar{F}_{\mathrm{LB}}(n)$ lower bounds P1's payoff when he/she plays according to $\bar{\sigma}_1^\epsilon$.

Let us now assume that P2 plays $\bar{\sigma}_2^\epsilon$. For any $X \in \mathcal{A}_1$, we have

$$\begin{aligned} F(X, \bar{\sigma}_2^\epsilon) &= \sum_{u_i \in \bar{U}^*} \bar{\sigma}_2^\epsilon(u_i) \left(\mathbb{1}_{[U_X \ni u_i]} + \varphi_{\min} \mathbb{1}_{[U_X \not\ni u_i]} \right) \\ &= \sum_{u_i \in \bar{U}^*} \frac{1}{\bar{m}^*} \left(\mathbb{1}_{[U_X \ni u_i]} + \mathbb{1}_{[U_X \not\ni u_i]} - \mathbb{1}_{[U_X \not\ni u_i]} + \varphi_{\min} \mathbb{1}_{[U_X \not\ni u_i]} \right) \\ &= 1 - (1 - \varphi_{\min}) \sum_{u_i \in \bar{U}^*} \frac{1}{\bar{m}^*} \mathbb{1}_{[U_X \not\ni u_i]} \\ &\overset{(*)}{\leq} 1 - (1 - \varphi_{\min}) \max \left\{ 0, \frac{\bar{m}^* - n}{\bar{m}^*} \right\} = \bar{F}_{\mathrm{UB}}(n). \end{aligned} \tag{7.26}$$

Here, (*) holds since $\bar{U}^*$ is a maximum set packing and $|X| \leq n$. Hence, the indices of at most n actuators from $\bar{U}^*$ can be improved by positioning X.

From (7.24), (7.25), and (7.26), we have that $(\bar{\sigma}_1^\epsilon, \bar{\sigma}_2^\epsilon)$ is an $\bar{\epsilon}$-NE with

$$\bar{\epsilon} = \bar{F}_{\mathrm{UB}}(n) - \bar{F}_{\mathrm{LB}}(n) = (1 - \varphi_{\min}) \frac{\bar{n}^* - n}{\bar{n}^*} - (1 - \varphi_{\min}) \max \left\{ 0, \frac{\bar{m}^* - n}{\bar{m}^*} \right\}.$$

Proof of (iii): Follows from (7.24) and (7.25).

Proof of (iv): Let $x_{i_1}, \ldots, x_{i_{n^*}}$ be the elements of X^*, $\bar{\varphi}_{i_1}, \ldots, \bar{\varphi}_{i_{n^*}}$ be the scaled indices associated with these elements, and $\alpha_1 \leq \ldots \leq \alpha_{n^*}$ be a sorted sequence of these indices. Let r be the largest number for which $\alpha_r = \varphi_{\min}$. Note that $\bar{n}^* \leq r$ (otherwise, the size of a minimum set cover of $U_{\min}$ would be smaller than $\bar{n}^*$). We show that r is the largest element of $Z_x(n, X^*)$. Then $S_x = r(1-\varphi_{\min})^{-1}$, and

$$F_{\mathrm{LB}}(n, X^*) \overset{(7.7)}{=} \frac{n}{r}(1-\varphi_{\min}) + \varphi_{\min} \overset{\bar{n}^* \leq r}{\leq} \frac{n}{\bar{n}^*}(1-\varphi_{\min}) + \varphi_{\min} = \bar{F}_{\mathrm{LB}}(n).$$

To prove that r is the largest element of $Z_x(n, X^*)$, we show that $r \in Z_x(n, X^*)$ (Claim 1), and that the elements larger than r do not belong to $Z_x(n, X^*)$ (Claim 2).

Claim 1. Since $\alpha_1 = \ldots = \alpha_r = \varphi_{\min}$, we have

$$1 - \frac{r-n}{\sum_{i=1}^{r}(1-\alpha_i)^{-1}} = 1 - \frac{r-n}{r(1-\varphi_{\min})^{-1}} = \frac{n}{r}(1-\varphi_{\min}) + \varphi_{\min} \overset{\varphi_{\min} \leq 1}{\geq} \varphi_{\min} = \alpha_r.$$

From the latter and the definition of $Z_x(n, X^*)$, we have $r \in Z_x(n, X^*)$.

Claim 2. The proof is by induction. If $r+1 \notin Z_x(n, X^*)$, then

$$\alpha_{r+1} > 1 - \frac{r+1-n}{r(1-\varphi_{\min})^{-1} + (1-\alpha_{r+1})^{-1}}$$

has to hold. Observe that

$$\begin{aligned} &\alpha_{r+1} > 1 - \frac{r+1-n}{r(1-\varphi_{\min})^{-1} + (1-\alpha_{r+1})^{-1}} \\ &\iff \alpha_{r+1} r(1-\varphi_{\min})^{-1} > r(1-\varphi_{\min})^{-1} - r + n \\ &\iff \alpha_{r+1} > \varphi_{\min} + \frac{n}{r}(1-\varphi_{\min}). \end{aligned}$$

The last inequality is satisfied, because

$$\alpha_{r+1} \geq \bar{\varphi}_{\min} \overset{(7.11)}{>} \varphi_{\min} + \frac{n}{\bar{n}^*}(1-\varphi_{\min}) \overset{\bar{n}^* \leq r}{\geq} \varphi_{\min} + \frac{n}{r}(1-\varphi_{\min})$$

holds. Hence, $r+1 \notin Z_x(n, X^*)$.

We now show that if $s \notin Z_x(n, X^*)$, then $s+1 \notin Z_x(n, X^*)$. Let us define $S' = \sum_{i=1}^{s}(1-\alpha_i)^{-1}$. If $s+1 \notin Z_x(n, X^*)$, then

$$\alpha_{s+1} > 1 - \frac{s+1-n}{S' + (1-\alpha_{s+1})^{-1}} \tag{7.27}$$

has to hold. By multiplying both sides by $S' + (1-\alpha_{s+1})^{-1}$ and rearranging the terms, we obtain that (7.27) is equivalent to $\alpha_{s+1} > 1 - (s-n)/S'$. Since $\alpha_{s+1} \geq \alpha_s$ and $\alpha_s > 1 - (s-n)/S'$ (by induction), the inequality (7.27) holds. Therefore, Claim 2 holds, so r is the largest element of $Z_x(n, X^*)$.

7.F Proof of Proposition 7.3

Note that $|\tilde{X}| \leq n$, since $\tilde{p}$ has to satisfy the second constraint of (7.17). Thus, $\tilde{X}$ is a feasible point of (7.16). We now show that $\tilde{X}$ is a solution of (7.16), and that the optimal values of (7.16) and (7.17) coincide.

Note that $\varphi_i \in (0, 1]$ by the definition of the function φ, and $\rho^* \geq 0$ as a dual solution of (7.14). Thus, the best way to maximize the objective function of (7.17) is to set as many elements of q to one. Yet, the first constraint in (7.17) imposes that an element q_j can be set to one only if u_j belongs to the monitoring set U_i for which $\tilde{p}_i = 1$. Equivalently, q_j can be set to one only if $u_j \in U_{\tilde{X}}$. Hence, for a fixed $\tilde{p}$, the largest objective value over all feasible $\tilde{q}$ is

$$\tilde{c} = \sum_{i=1}^{n_u} \rho_i^* \varphi_i \mathbb{1}_{[u_i \notin U_{\tilde{X}}]} + \sum_{i=1}^{n_u} \rho_i^* \mathbb{1}_{[u_i \in U_{\tilde{X}}]} - \pi^*. \tag{7.28}$$

Next, observe that the value of the objective function from (7.16) for $\tilde{X}$ is given by

$$(\rho^*)^T F_{\tilde{X}} - \pi^* \overset{(7.15)}{=} \sum_{i=1}^{n_u} \rho_i^* \varphi_i \mathbb{1}_{[u_i \notin U_{\tilde{X}}]} + \sum_{i=1}^{n_u} \rho_i^* \mathbb{1}_{[u_i \in U_{\tilde{X}}]} - \pi^* \overset{(7.28)}{=} \tilde{c}.$$

Thus, the optimal value of (7.16) is at least $\tilde{c}$.

We now prove by contradiction that the optimal value of (7.16) cannot be larger than $\tilde{c}$. Let X' be a solution and c' be the optimal value of (7.16). Suppose that $c' > \tilde{c}$ and let p' be given by $p'_i = \mathbb{1}_{[x_i \in X']}$. Since $|X'| \leq n$, p' satisfies the constraints of (7.17). For this p', we define q' by $q'_i = \mathbb{1}_{[u_i \in U_{X'}]}$. By the construction, q' satisfies the constraints of (7.17). Furthermore, we have

$$\sum_{i=1}^{n_u} \rho_i^* \big(\varphi_i(1 - q'_i) + q'_i\big) - \pi^* = \sum_{i=1}^{n_u} \rho_i^* \varphi_i \mathbb{1}_{[u_i \notin U_{X'}]} + \sum_{i=1}^{n_u} \rho_i^* \mathbb{1}_{[u_i \in U_{X'}]} - \pi^* \dots$$
$$\overset{(7.15)}{=} (\rho^*)^T F_{X'} - \pi^* = c'.$$

This contradicts the assumption that the optimal value of the problem (7.17) equals $\tilde{c}$, since $c' > \tilde{c}$. Thus, $\tilde{c}$ is the optimal value and $\tilde{X}$ is a solution of (7.17).

Chapter 8

Concluding remarks

Control system security is an important topic to address. Namely, control systems operate critical physical processes such as power production, transportation, and water distribution, so attacks against them may have dire consequences. For instance, the attacks that occurred in Maroochy shire led to an environmental hazard [4], the Stuxnet attack sabotaged the Iranian nuclear program [6], and the attack against the Ukranian power grid operators left thousands of households without electricity [9]. Moreover, control systems are both challenging and expensive to secure. Some reasons for this include their long life span, real time availability requirements, and potentially large size. Therefore, it is essential to develop cost-effective defense strategies for these systems.

Motivated by control system security, we studied two security-related applications: (i) classifying and preventing security vulnerabilities; and (ii) characterizing and improving the security level of actuators in large-scale control systems. For both applications, we developed security metrics that can help control system operators to determine where to focus security resources (risk assessment). Additionally, we provided tools that allocate security resources in a cost-effective manner based on these metrics (risk response). In the following, we summarize the results presented in the thesis in more detail, and outline possible directions for future work.

8.1 Summary

Application 1: Classifying and preventing security vulnerabilities

Chapter 4 addressed the problem **P1** presented in the introduction. Particularly, we introduced and studied a novel type of impact estimation problem. Two impact metrics that can be used in stochastic linear systems were proposed: The probability that some of the critical states leave a safety region and the expected value of

the infinity norm of the critical states. For the first metric, we proved that the optimal value of the problem can be computed efficiently by solving a set of convex problems. For the second metric, we derived lower and upper bounds that are efficient to compute. We then showed that our framework can be used to estimate the impact of a range of attack strategies proposed throughout the literature, and explained how to use properties of these strategies to estimate the impact more efficiently. Finally, we demonstrated on a control system of a chemical process how our framework can be used for classifying security vulnerabilities.

Chapter 5 considered **P2**. We proposed an algorithm that utilizes several systematic search tools to constructs the security measure allocation problem. We then showed that the problem is NP-hard, and introduced two suboptimal approaches to tackle the problem. The first approach is to first simplify the problem, and then use integer linear program solvers to compute a solution. The second approach exploits submodular structure of the problem, and uses a polynomial-time algorithm to compute a suboptimal solution with performance guarantees. The applicability of our security measure allocation framework was demonstrated on a control system for regulating temperatures. We also explained how the impact estimation framework from Chapter 4 can be combined with the security measure allocation framework.

Application 2: Characterizing and improving the security level of actuators in large-scale control systems

Chapter 6 tackled **P3**. We introduced the actuator security indices δ and δ_r that can be used for localizing vulnerable actuators. A method for computing δ was derived, and it was shown that δ may increase (resp. decrease) by placing additional sensors (resp. actuators). We then explained that δ is NP-hard to compute, sensitive to system variations, and based on the assumption that the attacker knows the entire system model. In contrast, the robust security index δ_r can be computed efficiently by solving the minimum s-t cut problem, can characterize actuators vulnerable in any system realization, and can be related to both the full and limited model knowledge attackers. We argued that these properties make δ_r more suitable for large-scale systems than δ. Finally, we illustrated how the indices we proposed can be used to characterize vulnerable generators in power grids.

Chapter 7 studied **P4**. We modeled this problem as a game, where the operator allocates a limited number of protected sensors to improve actuator security indices, while the attacker selects an actuator with a small value of the security index to attack. We focused on the case where the attacker uses the extended replay strategy. Our goal was to compute a NE monitoring strategy, or a good approximation of this strategy. Since the problem of computing a NE monitoring strategy was difficult to construct and solve for large-scale systems, we derived an ϵ-NE of the game. We then presented cases when this ϵ-NE becomes exact, explained how actuator security indices impact decision making of the players, and discussed how the monitoring

strategy from this ϵ-NE can be further improved. Finally, we demonstrated on a benchmark of a large-scale power grid that the tools we proposed allow us to compute NE monitoring strategies in a scalable manner.

8.2 Future work

Extending the impact estimation framework: The impact estimation framework from Chapter 4 was developed for stochastic linear systems. Thus, a possible extension is to consider more general system models such as nonlinear or hybrid systems. One can also make the network model more realistic by incorporating network imperfections such as packet drops and delays. Additionally, the attack strategies considered in Chapter 4 are all feedforward. That is, these strategies do not use on-line information about measurements and control actions to update an attack sequence over time. Hence, incorporating feedback attack strategies in the framework is another relevant extension.

Studying monotonicity of the attack impact: As shown in Chapter 4, the impact of the optimal FDI attack strategy is largest at the end of the estimation horizon. This useful property helped us to estimate the impact for this strategy more efficiently. We also showed through examples that other attack strategies can sometimes posses this property. Deriving conditions under which this happens can be an interesting problem to explore.

Reformulating the security measure allocation problem: Another way to formulate the security measure allocation problem is to set an upper bound on the security budget, and then maximize the number of prevented critical vulnerability combinations under this constraint. In this case, it is insufficient to find the sufficient representation of minimum cardinality to construct the problem. The reason is that we do not know which of the critical vulnerability combinations would be prevented. Hence, novel tools for constructing and solving this type of security measure allocation problem are required.

Expanding systematic search tools: In Chapter 5, we introduced Algorithm 5.1 for constructing the security measure allocation problem. Although Algorithm 5.1 managed to construct the problem in the experiments we conducted, we saw that the running time of this algorithm grows rapidly with the number of vulnerabilities present in the system. Thus, one future research direction is to further increase the efficiency of Algorithm 5.1 by deriving additional systematic search tools. Particularly, the goal is to investigate how to exploit prior information about the structure and symmetry of the control system for this purpose. Another idea is to focus on specific instances of the impact and likelihood functions, and then use properties of these instances to derive additional tools.

Generalizing sensor allocation strategies: The sensor allocation strategy from Chapter 6 was focused on increasing the robust security index δ_r, and did not take

the index δ into consideration. The future work may investigate if it is possible to increase the indices δ and δ_r simultaneously. A starting point can be to investigate if there exist system states X_i that satisfy the following property: If we place sensors to measure a subset of states from X_i, then both $\delta(u_i)$ and $\delta_r(u_i)$ increase.

Probabilistic robust security index: The modeling framework from Chapter 6 can be extended by taking the probability that a realization of the system will occur into account. The attacker may then want to gather resources such as to conduct an attack with sufficiently high probability of success. Hence, it is relevant to develop new security indices to capture this scenario.

Relaxing assumptions in the sensor allocation game: Chapter 7 considered the case where the attacker uses the extended replay strategy. We plan to generalize our analysis by allowing the attacker to use an arbitrary attack strategy, and try to derive a scalable way to construct a NE monitoring strategy for this case. Another possible extension is to relax the assumptions that we made on the matrices B and C.

Bibliography

[1] A. Kott, C. Aguayo Gonzalez, and E. Colbert, *Introduction and preview.* In: E. Colbert, A. Kott (eds) Cyber-security of SCADA and other industrial control systems. Springer International Publishing, 2016.

[2] A. A. Cárdenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems," *in Proceedings of the 3rd Conference on Hot Topics in Security*, 2008.

[3] D. Sullivan, E. Luiijf, and E. Colbert, *Components of industrial control systems.* In: E. Colbert, A. Kott (eds) Cyber-security of SCADA and other industrial control systems. Springer International Publishing, 2016.

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

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

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

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

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

[9] *Analysis of the cyber attack on the Ukrainian power grid.* Electricity Information Sharing and Analysis Center, 2016.

[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] *Guide to increased security in industrial information and control systems.* Swedish Civil Contingencies Agency, 2014.

[13] E. Knapp and J. Langill, *Industrial network security: Securing critical infrastructure networks for smart grid, SCADA, and other industrial control systems.* Syngress, 2014.

[14] *Recommended practice: Improving industrial control systems cybersecurity with defense-in-depth strategies.* Department of Homeland Security, 2016.

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

[16] 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.

[17] *Guide for conducting risk assessment.* National Institute of Standards and Technology, 2012.

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

[19] 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.

[20] J. Milošević, A. Teixeira, T. Tanaka, K. H. Johansson, and H. Sandberg, "Security measure allocation for industrial control systems: Exploiting systematic search techniques and submodularity," *International Journal of Robust and Nonlinear Control*, 2018.

[21] D. Urbina, J. Giraldo, A. Cárdenas, J. Valente, M. Faisal, N. O. Tippenhauer, J. Ruths, R. Candell, and H. Sandberg, *Survey and new directions for physics-based attack detection in control systems.* National Institute of Standards and Technology, 2016.

[22] S. Singh and S. Silakari, "A survey of cyber attack detection systems," *International Journal of Computer Science and Network Security*, vol. 9, no. 5, pp. 1–10, 2009.

[23] 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. 5, no. 4, pp. 1517–1528, 2018.

[24] 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, 2018.

[25] 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.

[26] D. Umsonst, H. Sandberg, and A. Cárdenas, "Security analysis of control system anomaly detectors," *in Proceedings of the American Control Conference*, 2017.

[27] 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.

[28] 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.

[29] 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.

[30] R. M. Ferrari and A. M. Teixeira, "Detection and isolation of routing attacks through sensor watermarking," *in Proceedings of the American Control Conference*, 2017.

[31] 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*, 2014.

[32] 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.

[33] 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.

[34] T. Sommestad, M. Ekstedt, and H. Holm, "The cyber security modeling language: A tool for assessing the vulnerability of enterprise system architectures," *IEEE Systems Journal*, vol. 7, no. 3, pp. 363–373, 2013.

[35] M. Bozorgi, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond heuristics: Learning to classify vulnerabilities and predict exploits," *in Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010.

[36] M. Zeller, "Myth or reality – Does the Aurora vulnerability pose a risk to my generator?" *in Proceedings of the 64th IEEE Annual Conference for Protective Relay Engineers*, 2011.

[37] S. Weerakkody, X. Liu, S. H. Son, and B. Sinopoli, "A graph-theoretic characterization of perfect attackability for secure design of distributed control systems," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 60–70, 2017.

[38] J.-M. Dion, C. Commault, and J. Van Der Woude, "Generic properties and control of linear structured systems: A survey," *Automatica*, vol. 39, no. 7, pp. 1125–1144, 2003.

[39] I. Hwang, S. Kim, Y. Kim, and C. E. Seah, "A survey of fault detection, isolation, and reconfiguration methods," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 636–653, 2009.

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

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

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

[43] P. E. Caines, *Linear stochastic systems*. SIAM, 2018.

[44] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Corporation, 2012.

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

[46] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: Analysis and design*. Wiley New York, 2007.

[47] J. Ackermann, *Robust control: Systems with uncertain physical parameters*. Springer Science & Business Media, 2012.

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

[49] A. Seuret, L. Hetel, J. Daafouz, and K. H. Johansson, *Delays and networked control systems*. Springer, 2016.

[50] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[51] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[52] K. Gatsis, A. Ribeiro, and G. J. Pappas, "Optimal power management in wireless control systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1495–1510, 2014.

[53] 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.

[54] A. Teixeira, *Toward cyber-secure and resilient networked control systems.* KTH Royal Institute of Technology, 2014.

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

[56] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[57] D. Kuipers and M. Fabro, *Control systems cyber-security: Defense in depth strategies.* Idaho National Laboratory, 2006.

[58] 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.

[59] 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, 2013.

[60] C. Bai, V. Gupta, and F. Pasqualetti, "On kalman filtering with compromised sensors: Attack stealthiness and performance bounds," *IEEE Transactions on Automatic Control*, vol. 62, no. 12, pp. 6641–6648, 2017.

[61] 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.

[62] S. Weerakkody, B. Sinopoli, S. Kar, and A. Datta, "Information flow for security in control systems," *in Proceedings of the 55th IEEE Conference on Decision and Control*, 2016.

[63] 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.

[64] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1495–1508, 2010.

[65] S. Sundaram, S. Revzen, and G. Pappas, "A control-theoretic approach to disseminating values and overcoming malicious links in wireless networks," *Automatica*, vol. 48, no. 11, pp. 2894 – 2901, 2012.

[66] 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.

[67] 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.

[68] E. E. Miciolino, R. Setola, G. Bernieri, S. Panzieri, F. Pascucci, and M. M. Polycarpou, "Fault diagnosis and network anomaly detection in water infrastructures," *IEEE Design Test*, vol. 34, no. 4, pp. 44–51, 2017.

[69] C. M. Ahmed, M. Ochoa, J. Zhou, A. P. Mathur, R. Qadeer, C. Murguia, and J. Ruths, "Noiseprint: Attack detection using sensor and process noise fingerprint in cyber physical systems," *in Proceedings of the Asia Conference on Computer and Communications Security*, 2018.

[70] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson, "Attack models and scenarios for networked control systems," *in Proceedings of the 1st International Conference on High Confidence Networked Systems*, 2012.

[71] Y. Z. Lun, A. D'Innocenzo, F. Smarra, I. Malavolta, and M. D. D. Benedetto, "State of the art of cyber-physical systems security: An automatic control perspective," *Journal of Systems and Software*, vol. 149, pp. 174 – 216, 2019.

[72] 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, 2017.

[73] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security – A survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1802–1831, 2017.

[74] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674 – 1683, 2018.

[75] S. M. Dibaji, M. Pirani, D. B. Flamholz, A. M. Annaswamy, K. H. Johansson, and A. Chakrabortty, "A systems and control perspective of CPS security," *Annual Reviews in Control*, 2019.

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

[77] 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*, 2013.

[78] 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.

[79] S. Feng and P. Tesi, "Resilient control under denial-of-service: Robust design," *Automatica*, vol. 79, pp. 42 – 51, 2017.

[80] 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*, 2010.

[81] 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.

[82] G. Franze, F. Tedesco, and W. Lucia, "Resilient control for cyber-physical systems subject to replay attacks," *IEEE Control Systems Letters*, vol. 3, no. 4, pp. 984–989, 2019.

[83] M. I. Müller, J. Milošević, H. Sandberg, and C. R. Rojas, "A risk-theoretical approach to $\mathcal{H}_2$-optimal control under covert attacks," *in Proceedings of the 57th IEEE Conference on Decision and Control*, 2018.

[84] 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.

[85] 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*, 2012.

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

[87] 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*, 2014.

[88] F. Miao, Q. Zhu, M. Pajić, 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.

[89] D. Shi, Z. Guo, K. H. Johansson, and L. Shi, "Causality countermeasures for anomaly detection in cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 2, pp. 386–401, 2018.

[90] S. Weerakkody and B. Sinopoli, "A moving target approach for identifying malicious sensors in control systems," *in Proceedings of the 54th Annual Allerton Conference on Communication, Control, and Computing*, 2016.

[91] M. Pajić, 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.

[92] 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.

[93] 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*, 2015.

[94] Y. H. Chang, Q. Hu, and C. J. Tomlin, "Secure estimation based Kalman filter for cyber-physical systems against sensor attacks," *Automatica*, vol. 95, pp. 399 – 412, 2018.

[95] 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.

[96] S. Sundaram and B. Gharesifard, "Distributed optimization under adversarial nodes," *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1063–1076, 2019.

[97] 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.

[98] Y. Mo and B. Sinopoli, "On the performance degradation of cyber-physical systems under stealthy integrity attacks," *IEEE Transactions on Automatic Control*, vol. 61, no. 9, pp. 2618–2624, 2016.

[99] G. Park, C. Lee, H. Shim, Y. Eun, and K. H. Johansson, "Stealthy adversaries against uncertain cyber-physical systems: Threat of robust zero-dynamics attack," *IEEE Transactions on Automatic Control*, vol. 64, no. 12, pp. 4907–4919, 2019.

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

[101] 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.

[102] B. Gerard, S. Bezzaoucha Rebai, H. Voos, and M. Darouach, "Cyber security and vulnerability analysis of networked control system subject to false-data injection," *in Proceedings of the American Control Conference*, 2018.

[103] 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.

[104] 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*, 2015.

[105] 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*, 2010.

[106] 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.

[107] I. Jovanov and M. Pajić, "Relaxing integrity requirements for attack-resilient cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 12, pp. 4843–4858, 2019.

[108] 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. 5, no. 3, pp. 1157–1168, 2018.

[109] 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.

[110] T. R, C. Murguia, and J. Ruths, "Tuning windowed chi-squared detectors for sensor attacks," *in Proceedings of the American Control Conference*, 2018.

[111] 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 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

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

[113] C. Murguia, I. Shames, J. Ruths, and D. Nešić, "Security metrics and synthesis of secure control systems," *Automatica*, vol. 115, p. 108757, 2020.

[114] N. H. Hirzallah and P. G. Voulgaris, "On the computation of worst attacks: A LP framework," in *Proceedings of the American Control Conference*, 2018.

[115] R. Anguluri, V. Gupta, and F. Pasqualetti, "Periodic coordinated attacks against cyber-physical systems: Detectability and performance bounds," *in Proceedings of the 55th IEEE Conference on Decision and Control*, 2016.

[116] K. Pan, A. M. H. Teixeira, M. Cvetković, and P. Palensky, "Combined data integrity and availability attacks on state estimation in cyber-physical power grids," *in Proceedings of the IEEE International Conference on Smart Grid Communications*, 2016.

[117] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, "Worst-case stealthy innovation-based linear attack on remote state estimation," *Automatica*, vol. 89, pp. 117 – 124, 2018.

[118] E. Kung, S. Dey, and L. Shi, "The performance and limitations of ϵ-stealthy attacks on higher order systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 941–947, 2017.

[119] 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*, 2010.

[120] 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.

[121] 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*, 2010.

[122] D. Deka, R. Baldick, and S. Vishwanath, "Data attack on strategic buses in the power grid: Design and protection," *in Proceedings of the IEEE PES General Meeting and Conference Exposition*, 2014.

[123] 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.

[124] 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.

[125] 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.

[126] 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.

[127] 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*, 2010.

[128] 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.

[129] K. C. Sou, H. Sandberg, and K. H. Johansson, "Electric power network security analysis via minimum cut relaxation," *in Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, 2011.

[130] ——, "Computing critical k-tuples in power networks," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1511–1520, 2012.

[131] 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*, 2014.

[132] Y. Yamaguchi, A. Ogawa, A. Takeda, and S. Iwata, "Cyber security analysis of power networks by hypergraph cut algorithms," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2189–2199, 2015.

[133] 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*, 2016.

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

[135] S. Zhao and F. Pasqualetti, "Networks with diagonal controllability Gramian: Analysis, graphical conditions, and design algorithms," *Automatica*, vol. 102, pp. 10 – 18, 2019.

[136] H. Cam, P. Mouallem, Y. Mo, B. Sinopoli, and B. Nkrumah, "Modeling impact of attacks, recovery, and attackability conditions for situational awareness," in *Proceedings of the IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, 2014.

[137] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Sensor placement for optimal Kalman filtering: Fundamental limits, submodularity, and algorithms," *in Proceedings of the American Control Conference*, 2016.

[138] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[139] 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.

[140] J. Milošević, A. Teixeira, H. Sandberg, and K. H. Johansson, "Actuator security indices based on perfect undetectability: Computation, robustness, and sensor placement," *IEEE Transactions on Automatic Control (Under review) [Online]. Available: https://arxiv.org/pdf/1807.04069.pdf*, 2019.

[141] A. Rahmattalabi, P. Vayanos, and M. Tambe, "A robust optimization approach to designing near-optimal strategies for constant-sum monitoring games," *in Proceedings of the Decision and Game Theory for Security*, 2018.

[142] M. Pirani, E. Nekouei, H. Sandberg, and K. H. Johansson, "A game-theoretic framework for security-aware sensor placement problem in networked control systems," *in Proceedings of the American Control Conference*, 2019.

[143] X. Ren and Y. Mo, "Secure detection: Performance metric and sensor deployment strategy," *IEEE Transactions on Signal Processing*, vol. 66, no. 17, pp. 4450–4460, 2018.

[144] M. Dahan, L. Sela, and S. Amin, "Network Inspection for Detecting Strategic Attacks," *Operations Research (Under review) [Online]. Available: https://arxiv.org/abs/1705.00349*, 2018.

[145] A. Krause, A. Roper, and D. Golovin, "Randomized sensing in adversarial environments," *in Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 2011.

[146] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Deployed armor protection: The application of a game theoretic model for security at the Los Angeles International Airport," *in Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2008.

[147] A. Washburn and K. Wood, "Two-person zero-sum games for network interdiction," *Operations Research*, vol. 43, no. 2, pp. 243–251, 1995.

[148] D. Bertsimas, E. Nasrabadi, and J. B. Orlin, "On the power of randomization in network interdiction," *Operations Research Letters*, vol. 44, no. 1, pp. 114–120, 2016.

[149] Ching-Tai Lin, "Structural controllability," *IEEE Transactions on Automatic Control*, vol. 19, no. 3, pp. 201–208, 1974.

[150] J. Willems, "Structural controllability and observability," *Systems & Control Letters*, vol. 8, no. 1, pp. 5–12, 1986.

[151] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[152] F. Bach *et al.*, "Learning with submodular functions: A convex optimization perspective," *Foundations and Trends in Machine Learning*, vol. 6, no. 2-3, pp. 145–373, 2013.

[153] M. Grant, S. Boyd, and Y. Ye, *CVX: Matlab software for disciplined convex programming*, 2008.

[154] L. Lovász, "Submodular functions and convexity," in *Mathematical Programming The State of the Art*. Springer, 1983.

[155] A. Krause and D. Golovin, "Submodular function maximization." *Technical Report*, 2014.

[156] L. Wolsey, "An analysis of the greedy algorithm for the submodular set covering problem," *Combinatorica*, vol. 2, no. 4, pp. 385–393, 1982.

[157] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*. Springer, 1972, pp. 85–103.

[158] V. V. Vazirani, *Approximation algorithms*. Springer Science & Business Media, 2013.

[159] M. Stoer and F. Wagner, "A simple min-cut algorithm," *Journal of the ACM*, vol. 44, no. 4, pp. 585–591, 1997.

[160] R. M. Gray, *Entropy and information theory.* Springer Science & Business Media, 2011.

[161] J. Duchi, "Derivations for linear algebra and optimization," *Technical Report*, 2007.

[162] T. M. Cover and J. A. Thomas, *Elements of information theory.* John Wiley & Sons, 2012.

[163] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.

[164] S. P. Strong, R. Koberle, R. R. d. R. van Steveninck, and W. Bialek, "Entropy and information in neural spike trains," *Physical Review Letters*, vol. 80, no. 1, p. 197, 1998.

[165] B. M. Roger, "Game theory: Analysis of conflict," *The President and Fellows of Harvard College, USA*, 1991.

[166] J. R. Marden and J. S. Shamma, "Game theory and distributed control," in *Handbook of game theory with economic applications.* Elsevier, 2015, vol. 4, pp. 861–899.

[167] ——, "Game theory and control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 105–134, 2018.

[168] T. Başar and P. Bernhard, *H-infinity optimal control and related minimax design problems: A dynamic game approach.* Springer Science & Business Media, 2008.

[169] R. N. Banavar and J. L. Speyer, "A linear-quadratic game approach to estimation and smoothing," *in Proceedings of the American Control Conference*, 1991.

[170] L. Deori, K. Margellos, and M. Prandini, "On the connection between Nash equilibria and social optima in electric vehicle charging control games," *IFAC Proceedings Volumes*, vol. 50, no. 1, pp. 14 320 – 14 325, 2017.

[171] E. Semsar-Kazerooni and K. Khorasani, "Multi-agent team cooperation: A game theory approach," *Automatica*, vol. 45, no. 10, pp. 2205 – 2213, 2009.

[172] R. Zhang and P. Venkitasubramaniam, "False data injection and detection in LQG systems: A game theoretic approach," *IEEE Transactions on Control of Network Systems*, 2019.

[173] Y. Li, D. Shi, and T. Chen, "False data injection attacks on networked control systems: A Stackelberg game analysis," *IEEE Transactions on Automatic Control*, vol. 63, no. 10, pp. 3503–3509, 2018.

[174] 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.

[175] K. Ding, S. Dey, D. E. Quevedo, and L. Shi, "Stochastic game in remote estimation under DoS attacks," *IEEE Control Systems Letters*, vol. 1, no. 1, pp. 146–151, 2017.

[176] P. N. Brown, H. P. Borowski, and J. R. Marden, "Security against impersonation attacks in distributed systems," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 1, pp. 440–450, 2019.

[177] A. Gupta, C. Langbort, and T. Başar, "Dynamic games with asymmetric information and resource constrained players with applications to security of cyberphysical systems," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 71–81, 2017.

[178] Q. Zhu and T. Başar, "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.

[179] 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, 2017.

[180] 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.

[181] A. R. Hota, A. A. Clements, S. Sundaram, and S. Bagchi, "Optimal and game-theoretic deployment of security investments in interdependent assets," *in Proceedings of the Decision and Game Theory for Security*, 2016.

[182] S. Amin, G. A. Schwartz, and S. S. Sastry, "Security of interdependent and identical networked control systems," *Automatica*, vol. 49, no. 1, pp. 186–192, 2013.

[183] K. G. Vamvoudakis, J. P. Hespanha, B. Sinopoli, and Y. Mo, "Detection in adversarial environments," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3209–3223, 2014.

[184] F. Miao, Q. Zhu, M. Pajić, and G. J. Pappas, "A hybrid stochastic game for secure control of cyber-physical systems," *Automatica*, vol. 93, pp. 55 – 63, 2018.

[185] D. Umsonst and H. Sandberg, "A game-theoretic approach for choosing a detector tuning under stealthy sensor data attacks," *in Proceedings of the IEEE Conference on Decision and Control*, 2018.

[186] A. Ghafouri, W. Abbas, A. Laszka, Y. Vorobeychik, and X. Koutsoukos, "Optimal thresholds for anomaly-based intrusion detection in dynamical environments," *in Proceedings of the International Conference on Decision and Game Theory for Security*, 2016.

[187] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1999.

[188] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe, "Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness," *Journal of Artificial Intelligence Research*, vol. 41, pp. 297–327, 2011.

[189] T. Lipp and S. Boyd, "Antagonistic control," *Systems & Control Letters*, vol. 98, pp. 44 – 48, 2016.

[190] 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*, 2018.

[191] M. Tsagris, C. Beneki, and H. Hassani, "On the folded normal distribution," *Mathematics*, vol. 2, no. 1, pp. 12–28, 2014.

[192] M. D. Perlman, "Jensen's inequality for a convex vector-valued function on an infinite-dimensional space," *Journal of Multivariate Analysis*, vol. 4, no. 1, pp. 52 – 65, 1974.

[193] S. Kaplan and B. Garrick, "On the quantitative definition of risk," *Risk analysis*, vol. 1, no. 1, pp. 11–27, 1981.

[194] R. Rymon, "Search through systematic set enumeration," *in Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, 1992.

[195] 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.

[196] J. W. Simpson-Porco, F. Dörfler, and F. Bullo, "Synchronization and power sharing for droop-controlled inverters in islanded microgrids," *Automatica*, vol. 49, no. 9, pp. 2603–2611, 2013.

[197] M. Amin and P. F. Schewe, "Preventing blackouts," *Scientific American*, vol. 296, no. 5, pp. 60–67, 2007.

[198] O. C. Imer, S. Yuksel, and T. Başar, "Optimal control of LTI systems over unreliable communication links," *Automatica*, vol. 42, no. 9, pp. 1429 – 1439, 2006.

[199] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 40–52, 2014.

[200] V. Tzoumas, M. A. Rahimian, G. J. Pappas, and A. Jadbabaie, "Minimal actuator placement with bounds on control effort," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 67–78, 2016.

[201] E. Tegling and H. Sandberg, "On the coherence of large-scale networks with distributed PI and PD control," *IEEE Control Systems Letters*, vol. 1, no. 1, pp. 170–175, 2017.

[202] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Sensor placement for optimal Kalman filtering: Fundamental limits, submodularity, and algorithms," *in Proceedings of the American Control Conference*, 2016.

[203] T. Cormen, *Introduction to algorithms*. MIT press, 2009.

[204] A. R. Bergen and D. J. Hill, "A structure preserving model for power system stability analysis," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-100, no. 1, pp. 25–35, 1981.

[205] S. K. M. Kodsi and C. A. Canizares, "Modeling and simulation of IEEE 14-bus system with facts controllers," *Technical Report*, 2003.

[206] A. Mohsenian-Rad and A. Leon-Garcia, "Distributed internet-based load altering attacks against smart power grids," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 667–674, 2011.

[207] H. Trentelman, A. Stoorvogel, and M. Hautus, *Control theory for linear systems*. Springer, 2012.

[208] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

[209] J. van der Woude, "The generic number of invariant zeros of a structured linear system," *SIAM Journal on Control and Optimization*, vol. 38, no. 1, pp. 1–21, 1999.

[210] J. Desrosiers and M. E. Lübbecke, *A primer in column generation.* In: G. Desaulniers, J. Desrosiers, M.M. Solomon (eds) Column generation. Springer, Boston, MA, 2005.