Optimal defense strategies to improve the security and resilience of Smart Grids

Ziad Ismail

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2016-ENST-0026

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité « Informatique et Réseaux »

présentée et soutenue publiquement par

Ziad Ismail

le 29 avril 2016

Stratégies de défense optimales pour améliorer la sécurité et la résilience des Smart Grids

Directeur de thèse : **Jean Leneutre**

Jury

M. Tansu Alpcan, Associate Professor, University of Melbourne	Rapporteurs
Mme Isabelle Chrisment, Professeur, Télécom Nancy	
M. Mohamed Kaaniche, Directeur de recherches, LAAS-CNRS	
M. Hervé Debar, Professeur, Télécom SudParis	Examinateurs
M. Fabio Martinelli, Senior researcher, IIT-CNR	
M. Gérard Memmi, Professeur, Télécom ParisTech	
M. Arnaud Ulian, Chef du Département MIRE, EDF R&D	
M. Lin Chen, Maître de conférences, Université Paris-Sud 11	Invité
M. Jean Leneutre, Maître de conférences, Télécom ParisTech	Directeur de thèse
Mme Alia Fourati, Ingénieur-Chercheur, EDF R&D	Encadrant industriel

TELECOM ParisTech
école de l'Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

T H È S E

Abstract

The smart grid is a modernized grid envisioned to provide new services both to the utility company and the customers. It will therefore increasingly rely on information and communication technologies, which will have the potential to increase its attack surface. The evolution of the threat landscape has made the security risk management in the smart grid a challenging task. Protecting critical assets, prioritizing the deployment of defense resources, and maintaining an all-time security awareness became the new security paradigm.

In the first part of this thesis, we employ game theoretical techniques to optimize the deployment of defense resources in the smart grid, focusing in particular on the impact of attacks on equipment. By analyzing the interactions between the attacker and the defender, we find the optimal choice of security modes to enable on each equipment in the Advanced Metering Infrastructure (AMI) to protect the confidentiality of customers' data. In the smart grid, the interdependency between the communication and the electric infrastructures also renders the management of the overall security risk a challenging task. Using non-cooperative game theory, we address this issue by presenting an analytical model for identifying and hardening the most critical communication equipment used in the power system. We validate our model via a case study based on the polish electric power transmission system.

The smart grid relies on industrial control systems to deliver electricity efficiently, reliably, and securely. In order to improve the security of these systems, the defense strategy needs to be both proactive by anticipating potential targets of adversaries, and reactive by adjusting the type and strength of the response to the threat level. In the second part of this thesis, we address this challenge by presenting a solution that computes the optimal security policy that guarantees that the defender's objectives are satisfied, based on information in an attack graph representing the evolution of the attacker's state in the system. The solution, based on Constrained Markov Decision Processes (CMDPs), can be used as a decision-making support system to assist the defender in responding to intrusions efficiently, or to prioritize the deployment of security countermeasures in the system before any attack attempt takes place. In addition, the solution can be combined with information in the attack graph to compare the relative security of two architectures or security configurations. We validate our approach on an AMI case study.

Acknowledgements

The work in this thesis could not have been accomplished without the generous help of a number of people. I would like to thank Dr. Jean Leneutre, my academic advisor, for his invaluable support and guidance throughout my PhD study. I feel extremely lucky for having Jean as my advisor, who allowed me to follow my own research interests without lot of restrictions. Jean's encouragement, patience, and support were always dedicated to his students, which provided an ideal research environment.

I would like to thank EDF for financing this thesis. During my PhD, I had the opportunity to work with a number of people in the I2D group in the SINETICS department, who provided valuable insights that guided my research work. In particular, I would like to thank Dr. David Bateman, my supervisor at EDF for the first half of this thesis, and Dr. Alia Fourati, who believed in the value of this research from the beginning, for their advice and their support during my PhD.

I would like to thank sincerely my committee members: Dr. Tansu Alpcan, Dr. Lin Chen, Prof. Isabelle Chrisment, Prof. Hervé Debar, Dr. Mohamed Kaaniche, Dr. Fabio Martinelli, Prof. Gérard Memmi, and M. Arnaud Ulian for agreeing to serve in my committee.

During my PhD, I had the opportunity to collaborate with a number of researchers. I would like to thank in particular Dr. Lin Chen for his support and technical advice during my PhD, and for our discussions that had an important impact in shaping this thesis. I greatly enjoyed my collaboration with Christophe Kiennert, Fabian Suchanek and Danai Symeonidou. I would also like to thank Peter Jensen, Frédéric Colin, Frédéric Guyomard, and John McDonald at EDF for our discussions that guided the research work carried out in this thesis.

At Télécom ParisTech, I had the opportunity to spend my time with the former and current members of the DbWeb research group of the department of Computer Science and Networks. I would like to thank Talel Abdessalem, Marie Al-Ghossein, Antoine Amarilli, Oana Bălălău, Albert Bifet, Maximilien Danisch, Jean-Louis Dessalles, Luis Galárraga, Jean-Benoît Griesner, Miyoung Han, Roxana Horincar, Quentin Lobbé, Mikaël Monet, Sébastien Montenez, Jacob Montiel Lopez, Thomas Rebele, Pierre Senellart, Mauro Sozio, and Katerina Tzompanaki for their support and friendship. I would also like to thank Philippe Godlewski, Elie Najm, and Thomas Robert of the department of Computer Science and Networks for their support and guidance. I thank my past officemates Lamine Ba and Modou Gueye for sharing many wonderful moments with me along the way. They became my closest friends and a source of unconditional support and encouragement.

Finally, I thank my parents, Ali and Chadia, for their unfailing support, love, and encouragement, and for instilling in me the value of education and encouraging me to always do my best.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the beginning of the 1990s, information systems that control power grid operations have been the subject of a profound transformation. The new communication technologies that accompanied the era of personal computing enabled the emergence of a new type of remotely accessible and controllable systems. The breakthroughs in the fields of communications, such as wireless technologies and fiber optics, and electronics, such as chips with increased speeds and efficiency, have led to the emergence of new services. In the last two decades, the new communication and information technologies, which have the capacity to increase the efficiency and reliability of industrial processes, have been increasingly embraced by power grid operators. The smart grid is a modernized grid that enables bidirectional flows of energy, and uses two-way communication and control capabilities that will lead to an array of new functionalities and applications [Nat10]. It is envisioned to increasingly rely on information technology to deliver electricity efficiently, reliably, and securely. The communication infrastructure that enables such services is very important, as it allows control and electrical engineers to monitor the state of the grid in real-time, in such a way that system failures are isolated as soon as they are identified. In addition, monitoring the customers' power consumption enables operators to adapt the generation to the load and to use energy resources more efficiently.

The modernization and increased digitization has shifted the structure of the power grid from a set of interconnected to a complex set of interdependent systems. The increased dependence of the smart grid on ICT (Information and Communications Technology) will potentially expose it to additional threats. Given the type and the number of interconnected equipment, in addition to the rapid evolution in the type and sophistication of threat agents, a strategic security risk assessment and an efficient distribution of defense resources are needed to protect the smart grid.

1.1 Context and Motivation

In March 2007, the US Department of Homeland Security conducted an experiment called "Aurora" at the Department of Energy's Idaho National Laboratory. In this experiment, researchers launched a cyber attack that caused a generator to self-destruct [Mes07]. This event has marked one of the first attempts to show the feasibility to inflict physical damage from a cyberattack. In addition, the revelations concerning the Stuxnet worm [FMC11] targeting control systems of Iran's nuclear program revived the interest and the sense of urgency in the security community to the importance of securing industrial control systems. This worm is believed to be the first cyber weapon aimed at sabotaging industrial control processes. The number of recent security breaches showed that their design failed to consider the risks and the impact of deliberate attacks [Hop12, Reu12, Gor09]. The notion that control networks of industrial systems are *air gapped* (isolated from unsecured networks such as the internet) does not hold anymore [Byr13] and the increased number of cyber attacks targeting these systems is a clear indication [ICS14].

The nature and type of threats targeting critical infrastructures, and the power grid in particular, has significantly evolved in the past two decades. The potential impact of cyberattacks and the number of cyber incidents targeting the energy sector [ICS15] has alarmed governments and stakeholders to the potential threats facing the smart grid. The use of off-the-shelf operating systems and standardized communication protocols have been a key contributing factor in increasing the reach and the scope of cyberattacks. Even though attackers' techniques became more sophisticated, defenders still relied on classic security approaches to protect their systems. The lack of agility in those approaches and the evolution in the threat landscape have made the security risk management a challenging task. The evolving security environment and ecosystem have made prioritizing defense resources, protecting critical assets, and maintaining a security awareness at all-time the new security paradigm.

The need to introduce new communication media to cope with the challenge of collecting and analyzing the massive volumes of data in the smart grid has exposed it to additional threats. The nature of the threats ranges from attacks aiming at disrupting the power grid operations, to compromising customer information, and manipulating the electricity market. To improve the security of critical infrastructures, standard bodies, industry groups, and governments have recently started to propose a set of recommendations [Cen, Nat14c, Dep09b], which include best-known practices tailored for the smart grid [Eur12a, Dep, Nat14b]. Beyond the application of this set of recommendations, the success of the smart grid depends on the dependability and the secure functioning of each component of the system. The security policy must move beyond static risk assessment frameworks to preemptive defense actions through a strategic analysis of attackers' methods and objectives. Therefore, the security of the smart grid needs to be examined through the lenses of both the attacker and the defender. In addition to the need to optimize the available defense resources, it becomes quickly clear that a new approach for securing the smart grid is needed.

As in any complex system such as the smart grid, different security objectives exist for each part of the infrastructure. The difficulty resides in understanding the relationships and interdependencies between the different components and their impact on the security policy. In the constrained environment of industrial control systems, it is important to ensure that the impact of a cyberattack can be absorbed without disrupting critical industrial processes. The impact of the attack should also remain within the utility's risk tolerance thresholds.

In the smart grid, the increased dependence on ICT will potentially expose it to additional threats. The complex interaction between the ICT and the power system makes it difficult to assess the impact of malicious attacks on the reliability, availability, and safety of the power grid. Different risk analysis methods exist to assess the reliability of the power grid [Wen05] and the security of the ICT infrastructure [ANS10]. However, most of these methods treat each infrastructure independently. The security of the power and ICT systems needs to be evaluated jointly to determine the risk of an unintended failure or accident or deliberate attack on each component of these systems. Through the study of the cascading impact of an attack, it becomes possible to identify the most critical parts of each system that cause the highest impact on the power grid.

Securing the smart grid aims also at protecting customer's data. In the smart grid, the electric utility is expected to collect a large amount of information about customers' power consumptions. Analyzing this data can expose habits and potentially be used to predict consumers' behaviors. Therefore, the widespread deployment of smart meters, which are electronic devices installed at the consumers' premises and are part of the Advanced Metering Infrastructure (AMI), raises privacy concerns. In France, 35 million smart meters are expected to be installed by 2021 [Ele]. Given the large number of meters, choosing the optimal security mode to enable on each equipment in the AMI becomes challenging.

Smart meters, in addition to the different sensors that will be installed in the smart grid, are expected to generate a large volume of data. Therefore, new computational capacities are needed to analyze the collected information. To cope with this challenge, utilities may be tempted to outsource the storage and the analysis of the data to the Cloud. However, from a security and business perspectives, each type of data has a different criticality. Therefore, the cloud provider may be forced to provide a different set of guarantees for each type of data. While encryption may prevent the disclosure of the protected information, from a risk management perspective, guarantees must be provided on the availability of the data. In particular, backup copies must be provided if an attacker managed to delete the original copies. The number of backups will eventually depend on the criticality of the data. Therefore, it is important to audit the cloud provider to ensure that the backup process is being respected.

Finally, to secure the smart grid, it is important to assess the motives, means, and methods of the adversaries. The optimal distribution of defense resources will eventually depend on the profile of the attacker. It is also of critical importance to identify the set of actions the adversary can undertake to compromise critical equipment. Identifying the set of

attack paths provides the defender with a holistic overview of the security interdependencies between the different equipment that were leveraged by the adversary. With this additional knowledge, and taking into account the set of constraints in an industrial environment, it becomes possible to generate an optimal security policy to protect the control system of the smart grid.

1.2 A Quick Overview of the State of the Art

In the power grid, a Supervisory Control and Data Acquisition (SCADA) system allows the operator to control physical equipment and collect data from field sensors. The need for interconnected equipment in the smart grid to monitor and control the system remotely requires the development of both secure and reliable communication protocols. However, one of the main challenges when designing security protocols or architectures for the smart grid is taking into account system evolution. For reasons of interoperability and backward compatibility, modifying the set of communication protocols is challenging.

The communication protocols for this type of systems such as DNP3 (Distributed Networking Protocol v3.0) [DNP], and standards such as IEC 61850 [IECa], among others, were not designed to ensure a secure data exchange between equipment. In this type of environment, communication protocols were initially designed to be reliable. However, the use of these protocols and the fact that equipment have limited computational resources made them vulnerable to cyberattacks [JNY11]. Even though secure versions for DNP3 [MPPW06, Gil08] and new standards have been proposed to handle the security of standards such as IEC 61850 [IECb], the potential threat of cyberattacks has not been thwarted.

The complexity of securing a system such as the smart grid is increased by the long life cycle of industrial control systems. In such environment, a complete upgrade of an existing vulnerable infrastructure is economically challenging. Therefore, in order to protect the system, recent efforts have focused on attack detection. An Intrusion Detection System (IDS) deployed to detect attacks in a critical industrial control system should be able to analyze the signatures of protocols data values, detect the communication patterns between equipment, and evaluate traffic inconsistencies [VM08]. Traffic patterns in industrial control networks are different from traditional IT networks [BSP12]. Therefore, special IDSs for these systems were developed [Tof, Mat].

To protect the smart grid, different approaches were proposed to detect attacks in the SCADA system [PSS+10, ZWS+11, CDF+07]. Some approaches do not require training data [YJ13], while others improve detection techniques for real-time attack detection in large critical infrastructures [BNT07]. The deterministic nature of the traffic in an industrial control system has led some researchers to model the expected behavior of the communications between equipment. Such attempts relied on comparing the observed communications

pattern against a known legitimate pattern [YUH06, GW13, RGS14]. Whitelisting approaches, aimed at classifying legitimate connections in a SCADA environment, have been proposed [BSP13]. Other works focused on combining different technologies to improve attack detection [YML+13]. A new method to detect zero-day attacks, based on detecting power consumption changes, was developed to protect industrial control systems [PFP].

While improving the attack detection capabilities in the smart grid is welcomed, it is important to assess the impact of traditional security solutions on the safety and dependability of equipment in an industrial control system [SW08b]. In addition, the smart grid is envisioned to provide new services, further relying on the communication infrastructure. An attack on a communication equipment used to control an industrial process can have severe impact on critical infrastructures [Pou03]. Reciprocally, an electrical node responsible of providing power to a set of communication equipment is important to the communication infrastructure: if the power source of these equipment is compromised, the communication nodes will not be able to achieve their objectives. The complex interaction between the ICT and the power system makes it difficult to assess the impact of malicious attacks on the reliability and availability of the power grid.

In addition to controlling the power grid, the control system in the smart grid is also responsible of collecting data from a set of field sensors. This data is not only used to know the current state of the grid, but also to predict power consumption, which helps the electric utility to balance power generation with demands. One of the features of the smart grid is an electricity market enabling a dynamic and real-time energy pricing [Nat10]. The electricity prices depend, among other factors, on the total energy consumption in the grid. Therefore, the users are motivated to reduce power usage when energy prices increase in peak consumption hours. The data about power consumptions is retrieved from smart meters installed at customers' premises. The electricity market is dependent on the availability and integrity of the data exchanged between the utility company and the customer where attacks can impact energy prices [LH11, XMS10]. In addition, attacks targeting information about users' energy needs could have undesirable impact on the grid [LYY+12]. False data injection attacks on the power system state estimation are also a serious threat in the smart grid [LNR11, HLCH11]. This type of attacks is generally carried out with incomplete information [RMR12] and does not always require a prior knowledge of the network topology [ENZH11].

The current trends in the state of the art to secure the smart grid mainly focus on attack detection. In the smart grid, the security solution must take into account the different constraints that exist in this type of critical infrastructures. In particular, the long life cycles, the constrained defense budget, and the criticality of the services provided by the different equipment will entail unavoidable choices for the deployment of security measures over the network. Therefore, the security strategy must aim at optimizing the utility of the available defense resources. For example, one of the most significant challenges to secure industrial control systems is managing the process of patching vulnerabilities. To tackle this problem, the defender must first assess the potential risk that an unpatched vulnerability poses to the

system. Recent advances in risk assessment methods are addressing two main limitations of the previous generations. First, a significant effort is made to propose quantitative security metrics to evaluate the security state of the system. Therefore, comparing the efficiency of different security configurations or defense strategies becomes easier. Second, the security risk on the system is evaluated taking into account the different methods an attacker can use to compromise critical equipment or services. In particular, the sequence of actions executed by the attacker will eventually impact the success likelihood of the attack.

1.3 Towards a Quantitative Risk Assessment Method

The nature of the threats targeting the smart grid varies from physical to cyber attacks. In this thesis, we will focus on developing methods and tools to secure the system against cyber attacks while taking into account the different constraints that exist in the smart grid. In general, to protect a system, we proceed in two steps. In the first step, we evaluate the risk of cyber attacks. During this phase, we try to identify the assets that are of interest to the defender and assess the probability of them being subject to attacks and the impact of these attacks. In the second step, we try to manage the risk by proposing adequate security measures to thwart cyber attacks. The efficiency of the security solution will depend on the methodology and the tools that are used in each of these steps. Therefore, it is essential to examine the advantages and limitations of each risk analysis method before choosing the most appropriate method in the context of the smart grid.

1.3.1 A Brief History

To enhance our understanding and improve the chance of choosing the right method, let us examine the evolution of the history of the development of risk analysis methods[1], which are classified into four generations. One of the most relevant work conducted on this subject was carried out by Baskerville [Bas93]. He classified the evolution of risk analysis methods into three generations. Each new generation focuses on addressing the limitations of the previous one. In the first generation, we find checklist-based methods. These methods assume that a complete list of security countermeasures exists. For example, an exhaustive list of every conceivable control that can be applied to protect the system exists. The security analysis and the design tasks are entirely rooted in physical system elements. In this generation, it is assumed that systems can be adequately protected by any generic set of security controls. The objective is to select security solutions to reduce the probability of a threat occurrence and the cost of an attack if it eventually occurs. LRAM [Gua87] and SPAN [ZHM90] are examples of risk analysis methods that belong to this generation.

[1] In this section, risk analysis methods refer to security risk assessment methods, which can sometimes also include a risk treatment phase.

The second generation of risk analysis methods attempted to address the limitations of the first generation by partitioning the system into a set of subsystems. In this case, a solution can be found for each subsystem where a particular requirement needs to be satisfied. The methods belonging to this generation make a number of assumptions. First, it is assumed that security system elements are complex and interconnected and therefore, they must be partitioned to be analyzed efficiently. Second, it is assumed that security controls are specific to each subsystem. Finally, modifying the system impacts its security and therefore, maintenance reviews must be conducted to ensure that security objectives are always satisfied. However, there is a major limitation of these methods. By analyzing each subsystem individually, we lack a global vision of the system. This prevents us from assessing the impact of the application of a security control in one subsystem on the others. In this category, we find the CRAMM [Far91] and the RISKPAC [Gar89] risk analysis methods.

To regain an overview of the impact of security solutions on the entire system, third generation methods addressed this issue by abstracting the problem and solution space. In this case, one of the main challenges resides in selecting the essential set of attributes that should be abstracted in the model. In general, during abstraction stages, the system model is represented independently from its concrete implementation. Therefore, only the correct level of abstraction will offer the necessary flexibility to assess the impact of the solution on the global system. In addition, as a design issue, the utility of the cost of security solutions outweighs the benefits of the solutions in importance, which is also inversely proportional to the abstraction level. SSADM-CRAMM [SSA91] is an example of a risk analysis method that belongs to the third generation. Methods based on Markov chains and Stochastic Petri Nets can also be classified in this generation as well.

The fourth generation of risk analysis methods was proposed by E. Bursztein [Bur08]. In this generation, the objective of risk analysis methods can be regarded as security properties that need to be verified (e.g. ensuring that an attacker cannot compromise an equipment that results in the violation of a security property). The verification of the properties and the construction of the model need to be done automatically while taking into account the complexity of the model to scale to large systems. In this category, the methods are referred to as frameworks as they are generally provided with software tools to construct and analyze the model automatically. In the remaining of this section, we will take a closer look at the most relevant works carried out in this generation of risk analysis methods.

In the fourth generation, the earliest work was carried by Baldwin [Bal94]. He developed SU-KUANG, which is a rule-based security checker system. The tool aims at finding security holes in the configuration of a machine automatically. Using a backward goal-based search algorithm, Zerkle and Levitt [ZL96] extended SU-KUANG system to the network environment. In 1994, Dacier and Deswarte [DD94] proposed the notion of the privilege graph. In this type of graphs, the nodes represent privileges and the edges represent vulnerabilities. They later converted the privilege graph into a Stochastic Petri Net (SPN) [DDK96]. The SPN is analyzed to compute the Mean Effort To security Failure (METF), which is a probabilistic metric based on assigning likelihoods to reach security failed states. The

implementation of this model was carried out by Ortalo et al. [ODK99]. In 1998, Phillips and Swiler [PS98] proposed a new model to generate attack graphs. The model requires a database of common attacks as input in addition to the attacker profile and information about the network configuration. A tool that implemented their model was later developed [SPEC01].

In 1999, Bruce Schneier popularized the notion of an attack tree [Sch99], which was later used to assess vulnerabilities in SCADA systems [BFM04]. Even though the model proposed by Schneier has not been formally presented and focused on attacks against a physical safe, it set the motion to a number of later works on attack graphs. One of the basic components of modern attack graph models is the *requires/provides* model. Developed by Templeton and Levitt [TL00], it represents and reduces the complexity of modeling chains of network attacks. The *requires* part of the model lists the necessary preconditions to execute an attack. The *provides* part lists the set of postconditions or the effects that result after a successful execution of the attack. Other works tackled this problem in specific scenarios. For example, Dawkins et al. [DCH02] provide a language to model exploits focusing in particular on attack trees, while Cuppens and Miege [CM02] propose an approach to model attacks in the specific framework of intrusion detection using their modeling language LAMBDA [CO00] for this task.

With respect to attack graphs, Ramakrishnan and Sekar [RS98] use model checking to search for unknown vulnerabilities on a single host. On the other hand, given a set of known exploits, Ritchey and Ammann [RA00] use model checking to analyze the security of a network and generate a single attack scenario if a security property is violated. Jha et al. [JSW02] and Sheyner et al. [SHJ+02] extend this work to generate all possible attack scenarios when generating attack graphs of networks. The limitation on the size of the network that can be analyzed using model checking techniques pushed researchers to search for other methods to generate attack graphs that can scale to large networks. In this line of research, Jajodia et al. [JNO03, JN10] compute attack paths based on a directed graph of the dependencies (via preconditions and postconditions) among exploits. In order to scale to large systems, Ingols et al. extend their previous work on predictive graphs [LIS+06] and propose the multiple-prerequisite graphs [ILP06].

The fourth generation of risk analysis methods offers significant improvements with respect to the previous generations. However, in the last decade, most of the work in this area focused on scaling to large systems. The increased complexity of information systems nowadays renders qualitative-based risk assessment methods impractical in general. In the next section, we will argue the need for quantitative-based methods for security risk analysis for critical infrastructures such as the smart grid.

1.3.2 Quantitative Metrics for Security Risk Assessment

In general, one of the main challenges of security risk assessment is to find a method or a set of metrics to quantify the security of a system. Security metrics are measurements used to assess the security posture of a system and were traditionally used in Information Technology (IT) systems. However, these same metrics cannot be directly applied to assess the security state of critical infrastructures [BSP12]. These systems have different objectives than traditional IT networks and are subject to strict functional constraints. In this type of systems, security metrics can be categorized into organizational, operational, and technical metrics [MBH07]. In general, security metrics can be used to quantify the security state of a system or to compare the security of different system configurations. For example, the notion of *attack surface*, first proposed by Manadhata and Wing [MW04], is used to compare the relative security of two versions of a system. The system is more vulnerable to attacks when its attack surface is more exposed. This metric could be used as a starting point to compare the security state of two configurations of a critical infrastructure.

One of the benefits of using security metrics is to identify vulnerable components. Wei and Ji [WJ10] propose metrics to estimate the resilience of control systems. Such metrics include the time needed to identify an incident, and the time during which the system can withstand an incident without performance degradation. McQueen et al. [MBFB06] propose a method to calculate a quantitative risk reduction estimate of security enhancements applied to a specific SCADA system. The risk is reduced when the value of the metric *time-to-compromise* a device decreases. The metric is a function of known vulnerabilities and attacker skill level and requires the knowledge of the security state of each device in the system.

Other techniques leverage attack graphs to assess the security state of the system. Based on the notion of privilege graphs developed at LAAS (Laboratoire d'analyse et d'architecture des systèmes), M. Dacier [Dac94] and R. Ortalo [Ort98] propose security metrics based on the time and effort needed to carry out an attack. G. Vache [Vac09] focuses on the set of vulnerabilities that can be exploited in the system and proposes a set of security metrics, which are probabilities that are a function of time, that include, among others, the probability of being compromised and the probability of being in a safe state at a certain time. Mehta et al. [MBZ+06] propose a metric to rank states in an attack graph based on Google's PageRank Algorithm. Along this direction, Sawilla and Ou propose the AssetRank algorithm to compute the importance of vertices in an attack graph based on their dependency relations [SO07]. In order to assess and compare the security of different network configurations, Wang et al. [WSJ07a] propose an attack resistance metric. In addition to this metric, Wang et al. [WSJ07b] later included the potential damage caused by attacks and the cost of reconfiguring a network in an integrated framework for measuring network security.

In large and complex systems such as the smart grid, conducting a qualitative security assessment of the system is a difficult task. Quantitative security metrics can offer an important insight on the security level of a system. However, despite all previous attempts, many challenges for adopting this type of approach remain. For example, a standard methodology to identify the metrics to use to evaluate the security of a particular system does not exist. In addition, it is always difficult to combine multiple metrics, each used to evaluate a certain security aspect of the system, in a global metric that reflects the vulnerability of the system. While the urgent need to quantitative security metrics is undeniable, more work needs to be done to identify and define the most appropriate metrics to use in critical and complex systems such as the smart grid.

Optimal defense resources needed to protect and defend the system can be viewed as a security metric that quantifies the level of vulnerability of the system. However, in this case, an optimal strategy of the defender should take into account the attacker's actions. Therefore, this type of interdependencies needs to be analyzed. In addition, the motives and the incentives to attack are a decisive factor affecting the behavior of attackers. In the next section, we present our approach to study the interdependent relationship between the defender and the attacker strategies and discuss the advantages and limitations of our method.

1.4 A Game Theoretical Approach

In addition to the classic security solutions such as Intrusion Detection Systems (IDSs), security researchers are applying novel approaches to study and analyze smart grid security problems. In fact, relying only on classic security approaches may not in general achieve an optimal distribution of the available defense resources. This is particularly challenging in a complex and large-scale system such as the smart grid. An efficient and intelligent approach to deploy defense measures in a system must aim at taking into account the potential actions of the attacker. Recently, a growing interest in using game theory to analyze cyber security related problems has emerged. Game theory is a mathematical tool that allows the analysis of complex interactions between different players with the same or conflicting interests. It has been used to study and analyze network security problems [MZA+10, AB10, SKTO13], and more specifically intrusion detection [CL09]. In the smart grid, it has been used to study, among others, distributed control and management of micro-grids [WK09, MK11, SHP11, SHPB12], energy consumption scheduling for demand-side management [MRWJ+10, VVR+10, NAC14, SV14, NAC15], and automatic response to attacks [ZKSY09].

1.4.1 Game Theory in Security

In recent years, game theory became an important tool to analyze the impact of the interactions between attackers and defenders on the security state of information systems. In this

environment, each player, namely the attacker and the defender, have conflicting interests. The outcome of each action of one player depends on the action of the other player. Capturing the interdependencies between players' actions becomes of great importance when protecting strategic assets.

The framework of non-cooperative games, in which players have conflicting interests and do not collaborate with each other, is generally used when analyzing the interactions between an attacker and a defender. In some cases, the interactions between multiple attackers and defenders are considered [CL09, AAG09]. In the security domain, the gain of one player can equal the loss of the other player [AB06, ZHZB10]. This type of games is referred to as zero-sum games. On the other hand, nonzero-sum games refer to scenarios where the sum of the attacker and the defender utilities is different than zero [CCZ08, ZFBB09].

Game theory has been used to optimize interdependent security investments [LFB15]. The model of an interdependent security game, proposed by Kunreuther and Heal [KH03], was the first attempt to study this problem. In this setting, the authors studied whether a player, in this case firms, have adequate incentives to invest in the protection against a risk whose magnitude depends on the actions of other players. This problem was later studied by Miura-Ko et al. [MKYBM08, MKYMB08]. Their work was based on using linear influence networks to analyze security decision-making in interdependent organizations.

In general, in an information system, the success likelihood of an attack on the network depends on the security investments on a certain set of equipment. In this case, the impact of individual security investment decisions on each equipment needs to be studied [GCC08, GJ09]. An effective solution to protect the system is optimizing the distribution of intrusion detection resources in the network. In this research area, a set of related work applied game-theoretical techniques for IDS optimization. For example, Nguyen et al. [NAB09] formulate the problem as a stochastic security game where node security assets and vulnerabilities are correlated. In [OMA+08], Otrok et al. present a game model for maximizing the detection probability of an attack split over multiple packets. The model is analyzed and practical guidelines are provided for IDS optimal sampling strategy. Other related work addressed intrusion detection problems in specific networks. Such attempts in Mobile Ad-hoc Networks (MANET) are presented in [PP11].

In information security, incomplete information about players' incentives impact the efficiency of security investment decisions [GJC10]. In addition, for each player, the limited information about the actions of the other players will directly affect his choice and eventually his payoff. Under these conditions, Alpcan and Başar [AB06] presented a zero-sum stochastic game for intrusion detection and explored various learning schemes players can use to optimize their strategies. In order to improve attack detection accuracy, collaboration techniques between IDSs have also been proposed [ZKL05]. However, in the absence of incentives, collaboration systems might suffer from the free-rider problem. Free-riding occurs when an element of the system takes advantage of the information shared by the others without itself contributing. Zhu et al. [ZFBB09] studied this problem and proposed

an incentive model based on trust management, using game theory, to remedy against the lack of collaboration.

In another set of related work, game-theoretical approaches were proposed to address the threat of Advanced Persistent Threats (APTs). In this context, van Dijk et al. [DJOR12] proposed the FlipIt game in which an attacker and a defender compete in taking control over a shared resource. Multiple extensions to this game, such as increasing the set of the defender's actions [PC12] and taking into account multiple resources [LHFB13], were proposed. Game theoretical techniques were also used to analyze the security of outsourced data in the cloud. In general, when outsourcing computations to untrusted contractors, cryptographically verifiable computations [GGP10, SVP+12] were used to prove the correctness of outsourced computational tasks. However, these solutions remain complex. Therefore, a game-theoretical model, through incentives and fines, was proposed to mitigate the malicious behavior of untrusted contractors [BCE+08, NK12a, KPC14].

1.4.2 Advantages and Limitations

In the last two decades, game theory, which is sometimes referred to as the science of conflict, has proved to be an effective tool to analyze the interactions between attackers and defenders. The concept of the Nash equilibrium (NE) in game theory allows the definition of the optimal strategies of players in which none of them has an incentive to unilaterally deviate from. From a security point of view, it means that it is possible to characterize the optimal strategy of the defender that takes into account the attacker's actions. Traditional security approaches such as Decision Theory fails to capture this feature. It assumes that defenders view attackers' actions as exogenous [CRY08]. In fact, it assumes that the attacker's strategy is given as an input to the model. The defender then uses this information to optimize his resources. Therefore, the defender's actions have no impact on the attacker. However, in general, attackers choose their targets based on the deployed defense mechanisms [CN06] and the expected rewards from successful attacks [SS03]. Game theory provides a framework to analyze the dynamic interactions between attackers and defenders and study their interdependent strategies. By operating at the NE, the defender is confident that the attacker cannot increase the impact of his attacks by changing his strategy unilaterally.

Different types of games exist. The decision of which type of games to choose depends on the nature and the characteristics of the interactions between the attacker and the defender. There are two types of games that are usually used and are of interest in the security domain. The first type is referred to as the *one-shot* game [OR94]. In this case, players choose their strategies simultaneously. However, the concept of simultaneity in game theory is different than its classic interpretation. From a game theoretic point of view, the actions of both players need not to be synchronous (at the same time) as in the classic definition. In a *one-shot* game, simultaneity refers to a player's lack of any observation of the other players' strategies before choosing his own strategy. The second type of games is referred to as a

Stackelberg game [OR94]. In most cases, the attacker chooses his attack strategy based on the deployed security measures in the system. In a Stackelberg game, a leader chooses his strategy first. Then, the follower, informed by the leader's choice, chooses his strategy. In general, the defender is the leader and the follower is the attacker. In this case, the defender tries to anticipate the attacker's response and chooses a strategy that minimizes the potential impact of attacks on the system.

In addition to the choice of the type of the game, it is important to closely examine the hypotheses we make about the profile of the attacker. Game theory is an abstraction that can be applied to analyze real-world scenarios only to the extent that its requirements are met. This is especially important when deploying security measures in a critical infrastructure such as the smart grid. In general, in game theory, players are assumed to be rational decision makers. However, the rationality assumption is usually the most controversial. This assumption has its root in the theory of rational choice. The objective of this theory is to explain human preference and choice by assuring that people are rational choosers. In this case, two main assumptions are made [Sch00]. First, that people have complete information about the costs and benefits associated with each of their actions. Second, that people compare their actions on a single scale of preference, or value, or utility. From a set of possible actions, people choose the action that maximizes their preferences, or values, or utilities. Assessment of utilities, which are subjective by nature, is a prerequisite for the application of game theory.

In real-world scenarios, some studies have found that these assumptions do not always hold. Sometimes, people violate the principles of rational choice [KT79, TK81, KT84]. In fact, people tend to minimize costs or weigh the benefits versus the costs when choosing their actions. In addition, any information about where people's preferences come from and the fact that rational choosers should always be able to express preferences is generally absent. To overcome these limitations in the security domain, it is important to assess the security of the system against a certain profile of the attacker. In the smart grid, we are interested in the worst-case scenario where we are trying to protect the system against intelligent attackers with strategic objectives. In general, we assume that attackers choose the actions that maximize the impact of their attacks on the system and the rewards of successful attacks. The behavior of one player depends also on the behavior of the other players. Therefore, each player must take into account the impact of the other players' actions on his preferences when comparing the set of available actions to choose from. For example, it may seem sometimes that an attacker chooses an action that does not reflect his preferences in the present. However, his decision can be deliberately chosen so as to maximize his chances of achieving his objective in the future. A compromise in the short term may prove to be a better strategy over the long haul.

Finally, the same knowledge about some of the features of the game is not always accessible to each player. This information asymmetry arises in some security situations. For example, the defender may be uncertain about the type of attackers targeting the system. In game theory, Bayesian games can be used to address this issue by offering the necessary

framework to derive optimal defense strategies under uncertainties about players' types. In addition, the number of defenders protecting a system and their incomplete knowledge about some of the system characteristics can impact the efficiency of their protections [JGCC10]. This uncertainty can be leveraged by the attacker and renders the detection of his actions more challenging. Game theory provides the necessary tools to cope with these challenges and analyze situations where partial knowledge is only accessible to some of the players.

1.5 Contributions and Outline of the Dissertation

In this thesis, we address the challenge of managing the security risk in the smart grid by defining optimal security policies and proposing an approach to optimize the distribution of defense resources on critical vulnerable equipment. In order to assess the risk of an attack on the system, the defender must go through two distinct, though complementary, evaluation stages. First, the defender focuses on the nature of the threat. In this case, he must identify the different ways that enable the attacker to compromise a target equipment. The resulting assessment, often called an attack graph, will be one of the main components for defining an optimal security policy. Second, the defender shifts his attention on the evaluation of the impact of attacks on target equipment and services in the system. By assessing the probability and the impact of attacks on the targeted assets, the defender is therefore better equipped with the knowledge needed to harden security on vulnerable and critical assets.

This thesis is divided into two parts, where each part focuses on harnessing the results of one of the risk assessment evaluation stages in order to protect the system.

In the first part, we will be interested in optimizing the distribution of defense resources on vulnerable equipment. We will focus on the impact of attacks on a set of equipment in the system without getting into the details of how the attacker managed to compromise these equipment. Our approach, based on game theory, aims at optimizing the distribution of defense resources while taking into account the interactions between the attacker and the defender. The interdependent nature of the decision making process for the attacker and the defender yields an optimal strategy for the defender in which the strategy of the attacker has been taken into account. By operating at the Nash equilibrium, the defender is confident that the attacker cannot improve his payoff by deviating from his strategy unilaterally. In this part, we apply our game theoretic approach on two case studies in the smart grid: the optimal configuration of security modes on equipment in the Advanced Metering Infrastructure (AMI) to maximize the protection of customers' data, and the security risk management of interdependent communication and electrical infrastructures.

The smart grid relies on industrial control systems, which are one of its main components, to offer the envisioned services reliably, efficiently, and securely. In the second part of this thesis, we will tackle the problem of identifying the different attack paths in an industrial control system that enable the attacker to achieve his objectives. Based on information about the profile of the attacker and the configuration of an industrial control system, we

build an attack graph, which we will later rely on to define optimal security policies that satisfy the defender's security objectives. Therefore, using the exhaustive list of attack paths, it becomes possible to harden security on intermediate compromised equipment in the system before critical target equipment or services are compromised.

To summarize, our contributions lie in the modeling of the behavior of the attacker and the defender in order to optimize the distribution of defense resources on vulnerable equipment in the smart grid, and the definition of optimal security policies based on attack graphs tailored for industrial control systems. The main results of this thesis are presented in Chapters 2-5. Since we tackle security issues in different areas in the smart grid, a related work is provided in each chapter, which focuses on the particular context of that chapter. In this thesis, we assume that the reader is familiar with the basic notions and concepts in game theory. In the remaining of this section, we give a summary of our contributions in this thesis.

Optimal configuration of security modes on equipment in the smart grid AMI

In the smart grid, smart meters, which are intelligent electronic devices, will be installed at customers' premises. These devices will enable two-way communication capabilities between the customers and the utility company. For example, these devices will be responsible of sending the power consumption of customers to the utility regularly. In addition, if a customer enabled demand response, the utility, through the smart meter, can control home appliances to save energy and reduce the customer's power consumption. Smart meters are equipment in the Advanced Metering Infrastructure (AMI). Therefore, the confidentiality of the data transiting in the AMI, which can include private information about customers that can expose their habits and be used to predict their behavior, must be guaranteed. In Chapter 2, we study this problem and present a game theoretical model for optimizing the configuration of security modes on equipment in the AMI to protect the confidentiality of customer's data. In our case, we focus on the encryption rate of outbound data sent by a device to the power utility. We formulate the problem as a non-cooperative game and analyze the behavior of the attacker and the defender at the Nash equilibrium. The attacker targets equipment in the AMI in order to collect the maximum amount of data on consumers. On the other hand, the defender chooses the encryption rate of outbound data on each device in the AMI. Using our model, we derive the minimum defense resources required to thwart attacks and the optimal strategy of the defender. We provide a case study to illustrate how our game theoretical model can be applied to configure encryption rates in the AMI in realistic scenarios.

Security risk management of interdependent communication and electrical infrastructures

The smart grid will increasingly rely on the communication infrastructure to ensure a reliable and secure delivery of electricity. The use of off-the-shelf operating systems in the communication infrastructure has the potential to increase the attack surface of the power grid. Therefore, the interdependencies between the communication system and the power

grid need to be studied to assess the impact of attacks on one infrastructure on the other. In Chapter 3, we address the issue of the security risk management of interdependent communication and electrical infrastructures in the smart grid by proposing an analytical model for hardening security on critical communication equipment used to control the power grid. Using non-cooperative game theory, we study the impact of the interdependencies between the two infrastructures on the behavior of the attacker and the defender. The attacker tries to compromise communication equipment to cause the maximum impact on the power grid. On the other hand, the defender tries to protect the power system by hardening the security on communication equipment, while taking into account the existence of backup control equipment in the communication infrastructure. We formulate the resulting scenario as one-shot and Stackelberg games and derive the optimal strategy of the defender that minimizes the risk on the power system in each case. We propose a method to assess the values of the parameters of the analytical model used to evaluate the impact of equipment failures in the power system. We also validate our model via a case study based on the polish electric power transmission system.

An attack execution model for industrial control systems security assessment

Critical infrastructures, such as the smart grid, rely on Industrial Control Systems (ICSs) to control and manage industrial processes. The improved communication capabilities of these systems in the last decade have the potential to increase their attack surface. In order to secure an ICS, it is important to identify the different methods that can be used by an adversary to compromise critical system components. In Chapter 4, we present the Attack Execution Model (AEM) to address this challenge. The AEM is an attack graph composed of a set of attack paths representing the sequence of actions that an attacker, with a certain profile, can execute in the system. In fact, the profile of the attacker, including his skill set and his knowledge of the topology and configuration of the system, plays a pivotal role in the success likelihood of his attack objectives. The AEM is generated automatically by analyzing information about the system and the attacker profile. In the generation process of our attack graph, we take into account the interdependencies between the different system components. In particular, the impact of an action executed by the attacker depends on the impact of successful actions he had already executed in the system. In addition, the impact of the attack can go beyond targeted equipment to impact services in the industrial process. In generating the AEM, our objective is to assess the risk of cyber attacks on an ICS before the next maintenance period. Therefore, the system operator will be better positioned to weigh the risk of waiting for the next maintenance period to harden security on vulnerable equipment. In Chapter 4, we give a formal definition of the AEM. We also analyze the performance of our tool that implements the model and show that is well suited to assess the risk of attacks on a typical sized ICS.

Optimal security policies for industrial control systems

In general, given budget constraints, the management of defense resources in a system comes with a tradeoff. In addition, the allocation of defense resources depends on the success likelihood of attacks and their impact on critical equipment. In Chapter 5, we address the challenging task of defining optimal security policies tailored for industrial control systems. In this context, two questions arise: What is the best strategy of the defender knowing the capabilities of the attacker and the fact that he was able to compromise a set of equipment in the system? What is the best response of the defender knowing the next action that the attacker will attempt to execute in the system? Based on the information in the attack graph generated in Chapter 4 and the available defense countermeasures, we automatically construct two types of Constrained Markov Decision Processes (CMDPs) to answer each of these questions. In our approach, we compute the optimal security policy taking into account the different constraints defined by the defender (e.g. the existence of a maximum defense budget and maximum tolerated thresholds for the probabilities of compromising critical equipment and services after the deployment of security countermeasures). In Appendix C, we present a game theoretical model for evaluating the probability distribution over the initial state, which is a parameter in the CMDP, when we are facing a threat posed by a rational and strategic external attacker. The solution of the CMDP problem can have multiple usage scenarios. The optimal security policy can be used as a decision-making support system to assist the defender in responding to intrusions efficiently. The solution can also be used to prioritize the deployment of security countermeasures in the system before any attack attempt takes place. Finally, our approach, combined with information in the attack graph generated in Chapter 4, can be used to compare the relative security of two architectures or security configurations. We validate our approach on an AMI case study in Chapter 6.

We conclude the thesis in Chapter 7 by summarizing the overall results and providing directions for future research.

In Appendix B, we analyze the problem of verifying data availability in the case of data outsourced to a cloud provider. A Service Level Agreement (SLA) is usually signed between the cloud provider and the customer. For redundancy purposes, it is important to verify the cloud provider's compliance with data backup requirements in the SLA. This task can be performed by the customer or be delegated to an independent entity that we will refer to as the verifier. We model the interactions between the verifier and the cloud provider as a non-cooperative game. The cloud provider's objective is to increase the storage capacity on his servers by not respecting the data backup agreement with the customer without being detected. On the other hand, the verifier's objective is to check the existence of the required number of backup copies for each type of data. We analyze different verification strategies and discuss the implications of operating at the Nash equilibrium in each case. We validate our game theoretical model numerically on a case study and provide guidelines on how to evaluate the game parameters in real-world scenarios.

Some of the results presented in this thesis were published in the following journals and conferences. The work on the game theoretical analysis of data confidentiality attacks on smart grid AMI was published in *IEEE Journal on Selected Areas in Communication (JSAC)* [ILBC14]. Our work on the security risk management of interdependent communication and electrical infrastructures was presented in part in the *IEEE 16th International Symposium on High Assurance Systems Engineering (HASE)* [ILBC15]. The work on generating attack graphs for industrial control systems security assessment was presented in part in the *First Conference on Cybersecurity of Industrial Control Systems (CyberICS)* [ILF15]. Our work on the verification of data availability in the case of data outsourced to a cloud provider is to be published in *IEEE Transactions on Information Forensics and Security*. The complete list of publications is given in Appendix A.

Part I

Security Strategies Based on Game Theoretical Analysis

One of the most challenging problems when securing the smart grid is optimizing the distribution of defense resources. In addition to the defender's preferences and strategic priorities, the optimal defense strategy will also depend on the potential actions of the attacker.

In this part, we employ game theoretic techniques to analyze the interactions between an attacker and a defender and derive the optimal distribution of defense resources. We focus in particular on two problems in the smart grid. In Chapter 2, we tackle the issue of protecting the confidentiality of customers' data in the Advanced Metering Infrastructure (AMI). We consider the scenario in which an attacker is trying to target AMI equipment to collect the maximum amount of data about customers. On the other hand, with a constrained defense budget, the defender aims at configuring security modes on equipment in the AMI to maximize the protection of customers' data. By taking into account the budget constraint and the potential strategy of the attacker, we derive the optimal security mode to enable on each equipment in the AMI. In addition, we compute the optimal defense budget needed to thwart attack attempts against customers' data in the AMI.

In Chapter 3, we study the impact of the interdependencies between the communication and electric infrastructures on the security risk management in the smart grid. An attack on a communication equipment used to control the power system can have undesirable impact on the grid. In particular, the failure of a number of equipment in the power system could lead to a cascading impact that threatens the stability of the power grid. By modeling the strategic interactions between an attacker and defender using game theory, we derive the optimal distribution of defense resources on communication equipment in order to minimize the impact of attacks on the power grid.

Chapter 2

A Game Theoretical Analysis of Data Confidentiality Attacks on Smart Grid AMI

The widespread deployment of smart meters in the Advanced Metering Infrastructure (AMI) raises privacy concerns. An attacker can potentially use data collected from a set of compromised smart meters to expose habits and predict consumers' behaviors. In this chapter, we analyze the confidentiality of information in an AMI consisting of nodes with interdependent correlated security assets. On each node, the defender can choose one of several security modes available. We try to answer the following questions: What is the expected behavior of a rational attacker? What is the optimal strategy of the defender? Can we configure security modes on each node so as to discourage the attacker from launching any attacks?

In this chapter, we formulate the problem as a non-cooperative game and analyze the behavior of the attacker and the defender at the Nash equilibrium. The attacker chooses his targets in order to collect the maximum amount of data on consumers, and the defender chooses the encryption rate of outbound data on each device in the AMI. Using our model, we derive the minimum defense resources required and the optimal strategy of the defender. Finally, we show how our framework can be applied in a real-world scenario via a case study.

2.1 Introduction

According to the report of the U.S. Energy Information Administration on the international energy outlook, the world energy consumption will grow by 56 % between 2010 and 2040 [Ene13]. To meet the increasing demand for energy, electric utilities need to produce energy more efficiently, and consumers to manage and control their power consumptions.

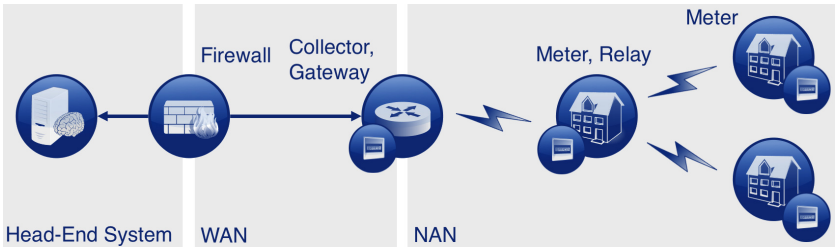


FIGURE 2.1: AMI hierarchy and network components

The Advanced Metering Infrastructure (AMI) is an integrated system of smart meters, communications networks, and data management systems that enables two-way communication between utilities and customers. In the AMI, smart meters are electronic devices installed at the consumers' premises. These devices send users' power consumptions to the utility. The power utility uses this data, among others, to predict power consumption curves for each area or neighborhood, to enable demand response, and to bill the user for the power consumed.

The AMI is mainly composed of three hierarchical areas (Figure 2.1). The Neighborhood Area Network (NAN) is a network of meters and collectors in the same geographical area [Eur12b]. Each collector in the NAN is responsible of collecting data from a set of smart meters. The WAN network includes gateways and routers that are responsible of connecting the utility head-end system to devices in the NANs. The utility head-end system analyzes the data collected from smart meters. In addition, it communicates with smart meters through collectors to request data or to send control commands.

The security of the smart grid, and in particular the AMI, is an active research topic. Due to the nature of industrial infrastructures, they were long been viewed as isolated, and therefore partially secured from external attacks. In fact, devices in this type of infrastructures used to communicate at the local level or through dedicated private connections. Most of these devices were not connected to the internet. However, the smart grid is envisioned to provide new services, further relying on the communication infrastructure. This increased number of connections with the telecommunication infrastructure, and in particular with the internet, has the potential to increase the attack surface of the smart grid.

In the context of smart metering, security objectives are different from other smart grid operations where priority is often given to guarantee data availability and integrity [Eur12c]. Data sent by smart meters is sensitive and need to be protected from attackers [Cle08]. Therefore, smart meters can be configured to operate in different security modes. Each mode protects a set of information sent to the utility. In this chapter, we refer to the security mode as the encryption rate of data sent by a device to the power utility.

The large number of devices deployed in the AMI renders the management of the overall security a challenging task. With a constrained defense budget, the defender often prioritizes the protection of assets in the system. In addition to protecting assets that are important to

the utility, the defender should protect targets that are identified as attractive to attackers. Therefore, in addition to the value of the assets, the security strategy of the defender should take into account the potential actions of attackers. In this chapter, we investigate this problem and propose a security game with two players, an attacker and a defender. The attacker's objective is to attack devices in the AMI in order to compromise data sent from these devices to the utility company. On the other hand, the defender has to choose which security mode to enable on each device in order to protect the maximum amount of data from the attacker. Our main contributions are as follows:

- We provide a game theoretical framework of data confidentiality in the AMI where nodes have different security assets.
- We derive the expected behavior of both the attacker and the defender, the optimal defense strategy that discourages the attacker from launching any attack and the minimum defense resources required to deploy that strategy.
- We provide a case study to demonstrate how our game theoretical framework can be implemented to optimize the defense resources in the AMI.

This chapter is organized as follows. We discuss related work in Section 2.2. We introduce our game model in Section 2.3. In Section 2.4, we analyze two types of interactions between the attacker and the defender and analyze the behavior of both players at the Nash equilibrium. In Section 2.5, we show via a case study, how our framework can be applied to configure security modes in the AMI. Finally, we conclude the chapter in Section 2.6.

2.2 Related Work

In general, a large number of devices in the smart grid have constrained computational resources. Therefore, cryptographic mechanisms and security protocols need to be adapted to this constrained environment [WL13]. In addition, security resources should be intelligently allocated to best protect both the utility and the consumers' data. In particular, the security solution should protect consumers' data along the communication path to the utility company [MWB11]. A possible solution based on homomorphic encryption is proposed by Li et al. [LLL10]. The authors propose a distributed incremental smart meter data aggregation approach using homomorphic encryption. This type of encryption allows certain algebraic operations on the plaintext to be performed directly on the cyphertext without the need to decrypt the data. In this system, each node is responsible of aggregating its own data with the data collected from its children. Therefore, users' data is protected and intermediate results remain secure. However, to guarantee that consumers' data will not be manipulated, the authors' solution assumes that intermediate nodes are not compromised. False data injection attacks have also been studied in the context of smart metering. Most of the literature regarding false data injection attacks assumes that the attackers have knowledge

of the network topology. In this case, Kosut et al. [KJTT10] propose an algorithm to find the minimum number of compromised meters that is needed to carry out an unobservable attack. On the other hand, the authors propose an algorithm at the control center level to detect and localize these attacks.

Data stored on smart meters and sent to utility companies is generally sensitive and therefore needs to be protected. By compromising this data, attackers could leverage information that could be used to threaten a customer's physical security. In addition, monitoring the behavior of customers through insecure AMI communications could help attackers commit crimes or perform robberies. In fact, the behavior of consumers could be predicted using Nonintrusive Load Monitoring (NILM) technology [LLC+03]. NILM can determine the operating schedule of electrical loads from measurements stored in a centralized location.

To protect the privacy of consumers' energy consumption metering data, approaches based on aggregating power consumptions of multiple consumers [RVK13], using in-residence batteries to mask appliance features [MMA11], and smart metering data anonymization [EK10] were proposed. Another approach, the Load Signature Moderation (LSM) technique [KED+10], changes appliances load signatures which makes it harder to distinguish the timing and the nature of appliances being used. In addition, multiple protocols were proposed to protect the confidentiality of consumers' data. Rial and Danezis [RD11] propose a privacy-preserving protocol that allows consumers to perform calculations on meter readings without disclosing any power consumption data. Rottondi et al. [RVC13] propose an infrastructure and a communication protocol to protect consumers' smart meters data. Special nodes, referred to as Privacy Preserving Nodes (PPN), are responsible of collecting consumers' data. The authors assume the integrity of PPNs. However, these nodes can be attractive targets to attackers for their potential value and importance in the system. Therefore, assuming the integrity of PPNs cannot be totally guaranteed.

In our model, we rely on an intrusion detection system installed on each device in the AMI to detect attacks. Designing intrusion detection systems for the AMI is an active research domain. One of the promising IDS solutions is proposed by Berthier and Sanders [BS11]. The authors' solution is a specification-based intrusion detection system for AMIs. In a specification-based intrusion detection system, any sequence of operations executed outside the system's specifications is considered to be a security violation. Therefore, this type of IDSs is capable of detecting unknown attacks.

A security solution for the AMI should take into account potential threats from adversaries. Attackers can take advantage of vulnerable points in the system to disrupt the service or compromise system equipment. Defenders often deploy security solutions with a constrained defense budget. Our work contributes to the existing literature by providing a game theoretical model to protect the confidentiality of consumers' data in the AMI. In the rest of this chapter, we analyze this problem, derive the minimum encryption resources required to thwart attacks in the AMI, and illustrate how our model can be used to configure data encryption rates in real-world scenarios via a case study.

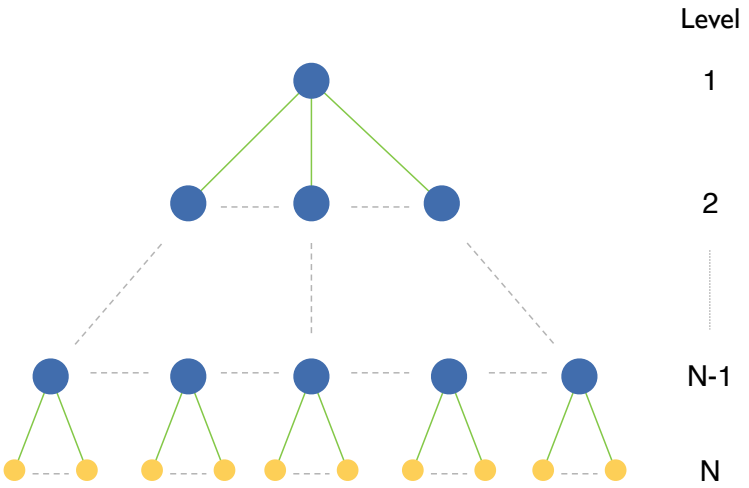


FIGURE 2.2: AMI communication architecture

2.3 System Model and Game Formulation

2.3.1 System Model

We consider a tree-like communication architecture $\mathcal{T}$ for the AMI with one root node as in Figure 2.2. In this architecture, nodes represent equipment in the AMI. Each node collects data from its children, aggregates it, and finally sends it to its parent node. We consider that there exists N aggregation levels. Let $\mathcal{V} = \{1, 2, ..., Y\}$ be the set of nodes in the tree $\mathcal{T}$, where Y is the total number of nodes. Let L_i be the set of nodes that belong to the i^{th} aggregation level. We consider that each node can only belong to one aggregation level. We refer by 1, the root node of $\mathcal{T}$. Smart meters are represented by nodes that belong to the N^{th} aggregation level.

Table 2.1 lists the main symbols used throughout this chapter.

We define the following functions:

$f : \mathcal{V}\backslash\{1\} \rightarrow \mathcal{V}$, function that returns for each node $i \in \mathcal{V}\backslash\{1\}$, its parent node.

$Ch : \mathcal{V} \times [\![1; N]\!] \rightarrow 2^{\mathcal{V}}$ (where $2^{\mathcal{V}}$ denotes the power set of $\mathcal{V}$), function such that for a particular node i and an aggregation level k, $Ch(i, k)$ refers to the set containing the children of $i \in L_m$ at level $k > m$. To simplify notations, we will refer by $Ch(i)$ the set containing the children of node i at level $m + 1$.

Data on each node i has a value or security asset W_i, which quantifies the loss in data confidentiality if the attack on node i is successful. We suppose that these values have been quantified as a result of the application of a security risk assessment method (e.g. [ANS10]). The parent node i collects data from all its children $Ch(i)$. A node could be responsible of processing and analyzing a set of the data collected from its children. The result of

TABLE 2.1: List of main symbols in Chapter 2

$\mathcal{T}$	a tree (a connected graph without cycles)
$\mathcal{L}(\mathcal{T})$	set of leaves of tree $\mathcal{T}$
$\mathcal{V}$	set of nodes in the tree $\mathcal{T}$
$\mathcal{V}_S$	set of sensible target nodes in $\mathcal{T}$
$\mathcal{T}_S$	a subtree of $\mathcal{T}$ consisting of nodes $i \in \mathcal{V}_S$
N	number of aggregation levels in $\mathcal{T}$
Y	total number of nodes in the tree $\mathcal{T}$
$N_S(i)$	maximum aggregation level of the leaves of the subtree $\mathcal{T}_i$ of $\mathcal{T}_S$ that has i as root node
L_i	set of nodes that belong to the i^{th} aggregation level
W_i	security asset of node i
W_i^k	security asset of the parent of node $i \in L_m$ at level $k < m$
∇_i^k	number of children of node $i \in L_m$ at level $k > m$
Δ_i^k	number of children of node $i \in L_m$ at level $k > m$ that belong to the sensible target set $\mathcal{V}_S$
$f(i)$	parent of node i
$Ch(i,k)$	set of the children of node $i \in L_m$ at level $k > m$
$Ch(i)$	set of the children of node $i \in L_m$ at level $m+1$
$Ch_S(i,k)$	set of the children of node $i \in L_m$ at level $k > m$ that belong to the sensible target set $\mathcal{V}_S$
$Ch_S(i)$	set of the children of node $i \in L_m$ at level $m+1$ that belong to the sensible target set $\mathcal{V}_S$
$\overline{Ch_S(i,k)}$	$Ch(i,k) \backslash Ch_S(i,k)$
$\mathbb{1}_{expr}$	equals 1 if $expr$ is true, 0 otherwise
p_i	probability of attacking node i
s_i	encryption rate of outbound data on node i
P	total attack resources
S	total encryption resources

this analysis is then sent with the aggregated data from children nodes to the parent node. Therefore, we consider that $W_i \geq \sum_{j \in Ch(i)} W_j$. The value of the data on node i is the sum of the value of the data generated by the node and the value of the data collected from its children.

For presentation reasons, we first consider that the tree $\mathcal{T}$ has N aggregation levels such that $\forall j \in L_{N-1}$, $Ch(j) \cap L_N \neq \varnothing$. However, we will show throughout this chapter that our framework can be applied to any types of trees.

Finally, let $\mathcal{L}(\mathcal{T})$ refer to the set of leaves of the tree $\mathcal{T}$. We refer by ∇_i^k, the number of children of node $i \in L_m$ at level $k > m$, and W_i^r the security asset of the parent of node $i \in L_m$ at level $r < m$. As notations, let $\nabla_i^k = 1$ and $W_i^k = W_i \ \forall i \in L_k$.

2.3.2 Game Formulation

We consider a game with two players, an attacker and a defender. On each node, the defender can choose one of a set of security modes available on that node. In our case, we consider that the defender chooses an encryption level of outbound data on each node. For example, if 100 packets are sent from the node, the defender chooses how many packets need to be encrypted. We consider that data on each communication link is encrypted with different encryption keys or using different encryption algorithms. At the root node, data is encrypted for storage after being analyzed.

The objective of the attacker is to intercept data by attacking the nodes without being detected. If the attacker wants to intercept data sent by node i, he can either attack node i or attack the parent node of i. We consider that encryption keys are stored in a cryptoprocessor that cannot be accessed by the attacker. The inbound data arrive at a device and is decrypted using the appropriate cryptographic key, processed and then encrypted using a different key. The attacker has no access or control on the decryption and encryption processes. We consider that on each node, an Intrusion Detection System (IDS) is installed with a detection rate of a. The IDS can be a combination of hardware and software detection capabilities.

Let p_i be the probability of attacking node i. The attacker's strategy is subject to a budget constraint $\sum_i p_i \leq P \leq 1$ $(0 \leq p_i \leq 1 \ \forall i)$. We consider that the attacker can attack only one particular device at any given time. Let s_i be the encryption rate of the packets at node i. In our model, the defender's strategy is subject to a budget constraint $\sum_i s_i \leq S \leq Y$ $(0 \leq s_i \leq 1 \ \forall i)$. In general, defense mechanisms deployed to protect a device depend on the value of the data generated, stored, or processed by that device. The efficiency, robustness and therefore the cost of the countermeasures deployed by administrators to protect devices are often proportional to the value of the assets on these devices. The attacker's effort to compromise data on a device increases with the efficiency of defense measures deployed to protect that device. Therefore, we consider that the cost of attacking and encrypting data on node i are proportional to the value of the data W_i and are given by $C_a W_i$ and $C_e W_i$ respectively, where $0 \leq C_a, C_e \leq 1$.

To intercept data sent by node i, the attacker can choose either to attack node i or its parent node $f(i)$. Therefore, the probability of compromising unencrypted data sent by i with an encryption level of s_i for W_i without being detected is given by $W_i(p_i + p_{f(i)})(1 - a)(1 - s_i)$. We assume that $1 - a > C_a$. Otherwise, the attacker has no incentive to attack since the cost to attack is greater than the payoff when the attack is successful and undetected.

The utility functions U_A and U_D of the attacker and the defender respectively are as follows:

$$
\begin{aligned}
U_A(p,s) &= \sum_{i\in\mathcal{V}} \big(W_i(p_i + p_{f(i)})(1-a)(1-s_i) - p_i C_a W_i\big) \\
&= \sum_{i\in\mathcal{V}} \big(W_i p_i (1-a)(1-s_i) - p_i C_a W_i\big) + \sum_{\substack{i\in\mathcal{V} \\ i\notin L_N}} \sum_{j\in Ch(i)} p_i W_j (1-a)(1-s_j) \\
U_D(p,s) &= -\sum_{i\in\mathcal{V}} \big(W_i p_i (1-a)(1-s_i) + s_i C_e W_i\big) - \sum_{\substack{i\in\mathcal{V} \\ i\notin L_N}} \sum_{j\in Ch(i)} p_i W_j (1-a)(1-s_j)
\end{aligned}
$$

2.4 Solving the Game

We model the interactions between the attacker and the defender as a non-cooperative game. We consider that the attacker and the defender have complete knowledge of the architecture of the system. In the context of non-cooperative games, we are interested in the concept of Nash equilibrium, in which none of the players has an incentive to deviate unilaterally [OR94]. The Nash equilibrium is considered as the most profitable strategy profile that gives each player the maximum utility given the actions of other players. Let $p = (p_1, ..., p_Y) \in \mathscr{P}$ and $s = (s_1, ..., s_Y) \in \mathscr{S}$ be the strategy profiles of the attacker and the defender respectively, where $\mathscr{P}$ and $\mathscr{S}$ refer to the strategy spaces of each player. We define the Nash equilibrium of our game as follows:

Definition 2.1 (Nash equilibrium)**.** *A* Nash equilibrium *is a strategy profile (p^*, s^*) in which each player cannot improve his utility by altering his decision unilaterally.*

More precisely, we have:

$$
\begin{aligned}
& U_A(p^*, s^*) \geq U_A(p, s^*) \ for\ all\ p \in \mathscr{P} \\
and\quad & U_D(p^*, s^*) \geq U_D(p^*, s) \ for\ all\ s \in \mathscr{S}
\end{aligned}
$$

2.4.1 Sensible Target Set

In Section 2.3, we considered that the attacker and the defender have limited attack and defense resources respectively. With a limited budget, it is rational to assume that both players will try to intelligently distribute their resources to maximize their utilities. Therefore, we can predict that the attacker will try to identify targets that yield the maximum profit, and then allocate resources to compromise data on these devices. On the other hand, the objective of the defender is to identify the targets that are most likely to be attacked, and protect the confidentiality of data by increasing data encryption rates on these devices.

Let $\mathcal{R}$ be a subset of the set of nodes $\mathcal{V}$. We refer by $\mathcal{M}(\mathcal{R})$, the set of nodes $i \in \mathcal{R}$ such that there are no node $j \in L_k \cap \mathcal{R}$ with $j \in Ch(i,k)$. For each node $i \in \mathcal{R}$, let $N_S(i)$ be the highest aggregation level of any node $j \in \mathcal{R}$ that is a child of i. Therefore, $N_S(i) = \max_k\{j \in L_k \cap \mathcal{M}(\mathcal{R}) \cap Ch(i,k)\}$. In the case where nodes in the set $\mathcal{R}$ form a tree $\mathcal{T}_R$, we have $\mathcal{M}(\mathcal{R}) = \mathcal{L}(\mathcal{T}_R)$.

We define the sensible target set $\mathcal{V}_S$ as a subset of $\mathcal{V}$ as follows:

Definition 2.2 (Sensible target set)**.** *The* sensible target set $\mathcal{V}_S$ *is a subset of* $\mathcal{V}$ *consisting of* $Y_A = |\mathcal{V}_S|$ *nodes and defined such that for every node* $i \in \mathcal{V}_S$*, we have:*

$$\begin{cases} W_i > \dfrac{1}{\alpha\big(1-\dfrac{C_a}{1-a}\big)}\Big(Y_A\big(1-\dfrac{C_a}{1-a}\big)+\beta-S\Big) & if\ N_S(i)=N \\ W_i > \dfrac{1}{\alpha\big(1-\dfrac{C_a}{1-a}\big)}\Big(Y_A\big(1-\dfrac{C_a}{1-a}\big)+\beta-S-\alpha \displaystyle\sum_{\substack{k\in Ch(j)\\ j\in Ch_S(i,N_S(i))}} W_k\Big) & if\ N_S(i)\neq N \end{cases}$$

$$\text{where } \alpha = \sum_{i\in\mathcal{V}_S}\frac{1}{W_i} + \sum_{r=1}^{N-1}\sum_{i\in L_r\cap(\mathcal{V}_S\setminus\mathcal{M}(\mathcal{V}_S))}\frac{\sum_{j=r+1}^{N_S(i)}(-1)^{j-r}\Delta_i^j}{W_i}$$

$$\text{and } \beta = -\sum_{r=1}^{N-1}\sum_{i\in L_r\cap(\mathcal{V}_S\setminus\mathcal{M}(\mathcal{V}_S))}\sum_{j=r+1}^{N_S(i)}\frac{(-1)^{j-r}}{W_i}\Big(\frac{C_a}{1-a}\sum_{m\in Ch_S(i,j)}W_m + \sum_{\substack{m\in\overline{Ch_S(i,j)}\\ f(m)\in\mathcal{V}_S\setminus\mathcal{M}(\mathcal{V}_S)}}W_m\Big) + \sum_{r=1}^{N-1}\sum_{i\in L_r\cap\mathcal{M}(\mathcal{V}_S)}\sum_{j\in Ch(i)}W_j\sum_{l=1}^{r}\frac{(-1)^{r-l}}{W_i^l}$$

From Definition 2.2, it follows that if a node $i \in \mathcal{V}_S$, $f(i) \in \mathcal{V}_S$ since $W_{f(i)} \geq \sum_{j\in Ch(f(i))} W_j \geq W_i$. For the rest of this chapter, we refer by $\mathcal{T}_S$, the tree with root node 1 formed by nodes in $\mathcal{V}_S$. Therefore, we have $\mathcal{M}(\mathcal{V}_S) = \mathcal{L}(\mathcal{T}_S)$. Let $Ch_S(i,j)$ refer to the set of the children of node i at level j that belong to $\mathcal{V}_S$. The intuition behind the sensible target set is to have a set of targets whose security assets' compromise yields the maximum payoff for the attacker. In our context, the security asset refers to the confidentiality of data processed by nodes. Analyzing certain types of information such as customers' data or power billing information can have severe impacts on both the customers and the utility company. The analysis can provide the attacker with the necessary information to predict a customer's behavior and habits, or even impact the utility's corporate image by exposing customers' credentials and power consumptions.

The sensible target set $\mathcal{V}_S$ is determined using Algorithm 1. We start by considering all the nodes in the set $\mathcal{V}$ and computing for each node i, a new value W_{t_i} that depends on the position of node i in the tree $\mathcal{T}$. Then, we sort these new values in descending order. In the new sorted set, we have $W'_1 \geq W'_2 \geq ... \geq W'_Y$. We start with the lowest value of W' and proceed by removing any node that does not belong to the sensible target set. We note that from Definition 2.2, the parent of any node that belongs to the sensible target set $\mathcal{V}_S$ is also a member of $\mathcal{V}_S$.

Lemma 2.1. *α is a positive real number.*

Algorithm 1

Input: Tree $\mathcal{T}$ and the set of nodes $\mathcal{V}$
Result: The sensible target set $\mathcal{V}_S$

function FindSensibleTargetSet($\mathcal{T}$, $\mathcal{V}$)
  for $x \in \mathcal{V}$ **do**
    if $x \in \mathcal{V} \backslash \mathcal{L}(\mathcal{T})$ **then**
      $W_{t_i} \leftarrow W_i + \frac{1}{(1-\frac{C_a}{1-a})} \sum_{j \in Ch(i)} W_j$
    else
      $W_{t_i} \leftarrow W_i$
    end if
  end for
  $W'_i \leftarrow$ SortInDescendingOrder($W_{t_{\sigma(i)}}$)
  Initialization: $Y_A = Y$, α, β
  while $Y_A \geq 1$ & $W'_{Y_A} \leq \frac{1}{\alpha(1-\frac{C_a}{1-a})}\big(Y_A(1-\frac{C_a}{1-a}) + \beta - S\big)$ **do**
    $Y_A \leftarrow Y_A - 1$
    update(α)
    update(β)
  end while
  $\mathcal{V}_S = \{\sigma(i) \in \mathcal{V},\ s.t.\ i \in [\![1; Y_A]\!]\}$
end function

Proof. For presentation reasons, we will prove that $\alpha > 0$ in the special case where $\mathcal{V}_S = \mathcal{V}$. The general case can be proved in a similar way. We prove the result by dividing α into disjoint sets and analyzing each set individually.

We assumed that $W_i \geq \sum_{j \in Ch(i)} W_j$. Therefore, $W_i \geq W_j\ \forall j \in Ch(i)$.

We start by dividing α into three disjoint parts.

$$\text{Let } \alpha = \sum_i \frac{1}{W_i} + \sum_{r=1}^{N-1} \sum_{i \in L_r} \frac{\sum_{j=r+1}^{N} (-1)^{j-r} \nabla_i^j}{W_i} = \text{I} + \text{II} + \text{III}$$

$$\text{I: } \sum_{i \in L_N} \frac{1}{W_i} + \sum_{j \in L_{N-1}} \frac{-\nabla_j^N}{W_j}$$

$$\begin{aligned} \text{We have, } \quad & Wi \geq W_j,\ \forall j \in Ch(i) \\ & \Rightarrow \frac{1}{W_i} \leq \frac{1}{W_j} \\ & \Rightarrow \frac{Ch(i)}{W_i} \leq \sum_{j \in Ch(i)} \frac{1}{W_j} \\ & \Rightarrow \text{I} \geq 0 \end{aligned}$$

$$
\begin{aligned}
\text{II:} \quad & \sum_{p=1}^{\lfloor \frac{N}{2}-1 \rfloor} \Bigg\{ \sum_{i \in L_{N-2p+1}} \frac{1}{W_i} + \sum_{i \in L_{N-2p}} \frac{1}{W_i} \\
& + \sum_{m \in L_{N-2p}} \frac{1}{W_m} \sum_{j=N-2p+1}^{N} (-1)^{j-N+2p} \nabla_m^j + \sum_{l \in L_{N-2p-1}} \frac{1}{W_l} \sum_{j=N-2p}^{N} (-1)^{j-N+2p+1} \nabla_l^j \Bigg\} \\
= & \sum_{p=1}^{\lfloor \frac{N}{2}-1 \rfloor} \Bigg\{ \sum_{i \in L_{N-2p+1}} \frac{1}{W_i} + \Bigg(\sum_{i \in L_{N-2p}} \frac{1}{W_i} - \sum_{l \in L_{N-2p-1}} \frac{\nabla_l^{N-2p}}{W_l} \Bigg) \\
& + \Bigg(\sum_{m \in L_{N-2p}} \frac{1}{W_m} \sum_{j=N-2p+1}^{N} (-1)^{j-N+2p} \nabla_m^j - \sum_{l \in L_{N-2p-1}} \frac{1}{W_l} \sum_{j=N-2p+1}^{N} (-1)^{j-N+2p} \nabla_l^j \Bigg) \Bigg\} \\
= & \sum_{p=1}^{\lfloor \frac{N}{2}-1 \rfloor} \Bigg\{ \sum_{i \in L_{N-2p+1}} \frac{1}{W_i} + \sum_{i \in L_{N-2p}} \underbrace{\left(\frac{1}{W_i} - \frac{1}{W_i^{N-2p-1}} \right)}_{\geq 0} \\
& + \sum_{m \in L_{N-2p}} \underbrace{\left(\sum_{j=N-2p+1}^{N} (-1)^{j-N+2p} \nabla_m^j \right)}_{>0} \underbrace{\left(\frac{1}{W_m} - \frac{1}{W_m^{N-2p-1}} \right)}_{\geq 0} \Bigg\} \\
\Rightarrow & \ \text{II} \geq 0
\end{aligned}
$$

$$
\begin{aligned}
\text{III:} \quad & \underbrace{\left(\frac{1-(-1)^N}{2} \right) \left(\sum_{i \in L_2} \frac{1}{W_i} + \sum_{m \in L_1} \frac{1}{W_m} \sum_{j=2}^{N} (-1)^{j-1} \nabla_m^j \right)}_{\geq 0} + \sum_{i \in L_1} \frac{1}{W_i} \\
\Rightarrow & \ \text{III} > 0
\end{aligned}
$$

Therefore, we conclude that $\alpha > 0$ for any tree $\mathcal{T}$ s.t. $\forall i \in \mathcal{T}, \exists k \in L_N \ s.t. \ k \in \nabla_i^N$.

We can verify that the lemma is valid for any types of trees with one root node. □

Lemma 2.2. *Data on all nodes will be encrypted with non-zero encryption rates if the defender has at least S_{min} encryption resources, where S_{min} is given by:*

$$S_{min} = Y\left(1 - \frac{C_a}{1-a}\right) + \beta$$

Proof. Follows directly from Definition 2.2. □

For the rest of this chapter, we consider that encryption resources are limited s.t. $S \leq S_{min}$.

Theorem 2.1. *A rational attacker attacks only nodes in the sensible target set $\mathcal{V}_S$.*

Proof. We consider the vector $s^0 = (s_1^0, ..., s_Y^0)$ where:

$$s_i^0 = \begin{cases} 1 - \frac{C_a}{1-a} - \frac{1}{\alpha W_i}(Y_A(1 - \frac{C_a}{1-a}) + \beta - S) + \frac{\mathbb{1}_{(N_S(i)\neq N)}}{W_i} \sum_{j \in Ch(i)} W_j & \forall i \in \mathcal{L}(\mathcal{V}_S) \\ 1 - \frac{C_a}{1-a} - \frac{A_i}{(1-a)W_i} & \forall i \in L_k \cap (\mathcal{V}_S \backslash \mathcal{L}(\mathcal{T}_S)) \\ 0 & i \in \mathcal{V} \backslash \mathcal{V}_S \end{cases}$$

where A_i is given in Appendix E.

First, we prove that the choice of s^0 is valid s.t. $s_i^0 \geq 0 \ \forall i$.

It is straightforward to show that $\forall i \in L_k \cap \mathcal{V}_S$ s.t $k \leq N-1$, we have:

$$\frac{C_a}{(1-a)W_i} \sum_{j=k+1}^{N_S(i)} (-1)^{j-k} \sum_{m \in Ch_S(i,j)} W_m \leq 0$$

This is done by grouping elements of the sum $\{k+1, k+2\}$, $\{k+3, k+4\}$, etc. and realizing that $-\sum_{m \in Ch_S(i,j)} W_m + \sum_{m \in Ch_S(i,j+1)} W_m \leq 0$, $\forall j \in [\![1; N-1]\!]$.

Similarly, we prove that $\frac{1}{W_i} \sum_{j=k+1}^{N_S(i)} (-1)^{j-k} \sum_{\substack{m \in \overline{Ch_S(i,j)} \\ f(m) \in \mathcal{V}_S \backslash \mathcal{L}(\mathcal{T}_S)}} W_m \leq 0$, $\forall i \in L_k \cap \mathcal{V}_S$ s.t. $k \leq N-1$.

Let $\phi = \frac{1}{\alpha}\Big(Y_A\big(1 - \frac{C_a}{1-a}\big) + \beta - S\Big)$.

ϕ is a positive real number since we have from Lemma 2.1 that $\alpha > 0$ and we supposed that $S \leq S_{min}$ where S_{min} is given in Lemma 2.2.

We know that $\forall i \in \mathcal{L}(\mathcal{T}_S)$, we have $W_i > \frac{1}{\alpha\big(1 - \frac{C_a}{1-a}\big)}\Big(Y_A\big(1 - \frac{C_a}{1-a}\big) + \beta - S - \alpha \mathbb{1}_{N_S(i)\neq N} \sum_{j \in Ch(i)} W_j\Big)$.

$\forall i \in L_{N_S(i)-1} \cap (\mathcal{V}_S \backslash \mathcal{L}(\mathcal{T}_S))$, we have:

$$\begin{aligned} W_i \geq \sum_{j \in Ch(i)} W_j \geq \sum_{j \in Ch_S(i)} W_j &> \frac{\Delta_i^{N_S(i)}\phi}{1 - \frac{C_a}{1-a}} - \frac{1}{1 - \frac{C_a}{1-a}} \sum_{j \in Ch_S(i)} \mathbb{1}_{N_S(j)\neq N} \sum_{k \in Ch(j)} W_k \\ &> \frac{(1 - \Delta_i^{N_S(i)})\phi}{1 - \frac{C_a}{1-a}} - \frac{1}{1 - \frac{C_a}{1-a}} \sum_{j \in Ch_S(i)} \mathbb{1}_{N_S(j)\neq N} \sum_{k \in Ch(j)} W_k \end{aligned}$$

Let us suppose that,

$$W_i > \frac{\phi}{1 - \frac{C_a}{1-a}}\Big(1 + \sum_{j=k+1}^{N_S(i)} (-1)^{j-k} \Delta_i^j\Big) - \frac{1}{1 - \frac{C_a}{1-a}} \sum_{j=k+1}^{N_S(i)} \sum_{m \in L_j \cap \mathcal{L}(\mathcal{T}_S) \cap Ch_S(i,j)} \mathbb{1}_{(j\neq N)} (-1)^{j-k} \sum_{t \in Ch(m)} W_t$$

is true $\forall i \in L_k \cap (\mathcal{V}_S \backslash \mathcal{L}(\mathcal{T}_S)), k \leq N-1$. Therefore, $\forall m \in L_{k'}$ s.t $k' = k-1$ we have:

$$
\begin{aligned}
W_m \;&\geq \sum_{j \in Ch(m)} W_j \geq \sum_{j \in Ch_S(m)} W_j \geq \sum_{l \in Ch_S(m,k+1)} W_l \\
&> \sum_{l \in Ch_S(m,k+1)} \frac{\phi}{1-\frac{C_a}{1-a}} \Big(1 + \sum_{r=k+2}^{N_S(l)} (-1)^{r-k-1} \Delta_j^r\Big) - \frac{1}{1-\frac{C_a}{1-a}} \sum_{j \in Ch_S(i)} \mathbb{1}_{N_S(j) \neq N} \sum_{k \in Ch(j)} W_k \\
&\quad - \sum_{j \in Ch_S(i)} \frac{1}{1-\frac{C_a}{1-a}} \sum_{r=k'+2}^{N_S(j)} \sum_{m \in L_r \cap \mathcal{L}(\mathcal{T}_S) \cap Ch_S(j,r)} \mathbb{1}_{(r \neq N)} (-1)^{r-k'-1} \sum_{t \in Ch(m)} W_t \\
&> \frac{\phi}{1-\frac{C_a}{1-a}} \Big(\Delta_m^{k'+2} + \sum_{r=k'+3}^{N_S(i)} (-1)^{r-k'} \Delta_m^r\Big) - \frac{1}{1-\frac{C_a}{1-a}} \sum_{j \in Ch_S(i)} \mathbb{1}_{N_S(j) \neq N} \sum_{k \in Ch(j)} W_k \\
&\quad + \frac{1}{1-\frac{C_a}{1-a}} \sum_{j \in Ch_S(i)} \sum_{r=k'+2}^{N_S(j)} \sum_{m \in L_r \cap \mathcal{L}(\mathcal{T}_S) \cap Ch_S(j,r)} \mathbb{1}_{(r \neq N)} (-1)^{r-k'} \sum_{t \in Ch(m)} W_t \\
&> \frac{\phi}{1-\frac{C_a}{1-a}} \Big(1 - \Delta_m^{k'+1} + \Delta_m^{k'+2} + \sum_{r=k'+3}^{N_S(i)} (-1)^{r-k'} \Delta_m^r\Big) \\
&\quad - \frac{1}{1-\frac{C_a}{1-a}} \sum_{r=k'+1}^{N_S(i)} \sum_{m \in L_r \cap \mathcal{L}(\mathcal{T}_S) \cap Ch_S(i,r)} \mathbb{1}_{(r \neq N)} (-1)^{r-k'} \sum_{t \in Ch(m)} W_t \\
&= \frac{\phi}{1-\frac{C_a}{1-a}} \Big(1 + \sum_{r=k'+1}^{N_S(i)} (-1)^{r-k'} \Delta_i^r\Big) \\
&\quad - \frac{1}{1-\frac{C_a}{1-a}} \sum_{r=k'+1}^{N_S(i)} \sum_{m \in L_r \cap \mathcal{L}(\mathcal{T}_S) \cap Ch_S(i,r)} \mathbb{1}_{(r \neq N)} (-1)^{r-k'} \sum_{t \in Ch(m)} W_t
\end{aligned}
$$

Therefore, $1 - \frac{C_a}{1-a} - \frac{A_i}{(1-a)W_i} > 0 \; \forall i \in L_k \cap (\mathcal{V}_S \backslash \mathcal{L}(\mathcal{T}_S))$. As a result, we proved that vector $s_i^0 > 0 \; \forall i$.

We have $\sum_{i \in \mathcal{V}_S} s_i^0 = S$. Let $s = (s_1, ..., s_Y)$ be the strategy of the defender. By the pigeonhole principle, $\sum_{i \in \mathcal{V}_S} s_i \leq S$, thus $\exists m \in \mathcal{V}_S$ s.t. $s_m \leq s_m^0$.

We consider the attacker strategy satisfying $\sum_{i \in \mathcal{V} \backslash \mathcal{V}_S} p_i > 0$. We construct the attacker strategy profile $p^{'}$ s.t.:

$$
p_i^{'} = \begin{cases} p_i & i \in \mathcal{V}_S \text{ and } i \neq m \\ p_m + \sum_{j \in \mathcal{V} \backslash \mathcal{V}_S} p_j & i = m \\ 0 & i \in \mathcal{V} \backslash \mathcal{V}_S \end{cases}
$$

$$U_A(p^{'},s) = \sum_{i\in\mathcal{V}_S}\big(W_i p_i^{'}(1-a)(1-s_i) - p_i^{'}C_a W_i\big) + \sum_{\substack{i\in\mathcal{V}_S\\ i\notin\mathcal{L}(\mathcal{T})}}\sum_{j\in Ch(i)} p_i^{'}W_j(1-a)(1-s_j)$$

$$= \sum_{i\in\mathcal{V}_S, i\neq m}\big(W_i p_i(1-a)(1-s_i) - p_i C_a W_i\big) + \big(p_m + \sum_{j\in\mathcal{V}\setminus\mathcal{V}_S} p_j\big)W_m\big((1-a)(1-s_m) - C_a\big)$$
$$+ \sum_{\substack{i\in\mathcal{V}_S, i\neq m\\ i\notin\mathcal{L}(\mathcal{T})}}\sum_{j\in Ch(i)} p_i W_j(1-a)(1-s_j) + \mathbb{1}_{(N_S(m)\neq N)}\big(p_m + \sum_{j\in\mathcal{V}\setminus\mathcal{V}_S} p_j\big)\sum_{k\in Ch(m)}(1-a)W_k$$

Therefore,

$$U_A(p,s) - U_A(p^{'},s)$$
$$= \sum_{i\in\mathcal{V}\setminus\mathcal{V}_S}\big(W_i p_i(1-a)(1-s_i) - p_i C_a W_i\big) + \sum_{\substack{i\in\mathcal{V}\setminus\mathcal{V}_S\\ i\notin L_N}} p_i \sum_{j\in Ch(i)} W_j(1-a)(1-s_j)$$
$$- \sum_{i\in\mathcal{V}\setminus\mathcal{V}_S} p_i W_m\big((1-a)(1-s_m) - C_a\big) - \mathbb{1}_{N_S(m)\neq N}\sum_{i\in\mathcal{V}\setminus\mathcal{V}_S} p_i \sum_{k\in Ch(m)}(1-a)W_k$$
$$\leq \sum_{i\in\mathcal{V}\setminus\mathcal{V}_S}\big(W_i p_i(1-a)(1-s_i) - p_i C_a W_i\big) + \sum_{\substack{i\in\mathcal{V}\setminus\mathcal{V}_S\\ i\notin L_N}} p_i \sum_{j\in Ch(i)} W_j(1-a)(1-s_j)$$
$$- \sum_{i\in\mathcal{V}\setminus\mathcal{V}_S} p_i W_m\big((1-a)(1-s_m^0) - C_a\big) - \mathbb{1}_{N_S(m)\neq N}\sum_{i\in\mathcal{V}\setminus\mathcal{V}_S} p_i \sum_{k\in Ch(m)}(1-a)W_k$$
$$\leq \sum_{i\in\mathcal{V}\setminus\mathcal{V}_S}\big(W_i p_i(1-a) - p_i C_a W_i\big) + \sum_{\substack{i\in\mathcal{V}\setminus\mathcal{V}_S\\ i\notin L_N}} p_i \sum_{j\in Ch(i)} W_j(1-a)$$
$$- \sum_{i\in\mathcal{V}\setminus\mathcal{V}_S} p_i W_m\big((1-a)(1-s_m^0) - C_a\big) - \mathbb{1}_{N_S(m)\neq N}\sum_{i\in\mathcal{V}\setminus\mathcal{V}_S} p_i \sum_{k\in Ch(m)}(1-a)W_k$$

However, $- \sum_{\substack{i\in\mathcal{V}\setminus\mathcal{V}_S\\ i\in L_N}} p_i W_m\big((1-a)(1-s_m^0) - C_a\big) - \mathbb{1}_{N_S(m)\neq N}\sum_{\substack{i\in\mathcal{V}\setminus\mathcal{V}_S\\ i\in L_N}} p_i \sum_{k\in Ch(m)}(1-a)W_k$

$$+ \sum_{\substack{i\in\mathcal{V}\setminus\mathcal{V}_S\\ i\in L_N}}\big(W_i p_i(1-a) - p_i C_a W_i\big)$$
$$= -\sum_{\substack{i\in\mathcal{V}\setminus\mathcal{V}_S\\ i\in L_N}} p_i\frac{(1-a)}{\alpha}\Big(Y_A\big(1-\frac{C_a}{1-a}\big) + \beta - S\Big) + \sum_{\substack{i\in\mathcal{V}\setminus\mathcal{V}_S\\ i\in L_N}}\big(W_i p_i(1-a) - p_i C_a W_i\big)$$
$$= \sum_{\substack{i\in\mathcal{V}\setminus\mathcal{V}_S\\ i\in L_N}} p_i(1-a)\Big(W_i\big(1-\frac{C_a}{1-a}\big) - \frac{1}{\alpha}\big(Y_A(1-\frac{C_a}{1-a}) + \beta - S\big)\Big)$$
$$< 0$$

$$
\begin{aligned}
\text{and,}\quad & \sum_{\substack{i\in\mathcal{V}\setminus\mathcal{V}_S\\ i\notin L_N}} \big(W_i p_i(1-a) - p_i C_a W_i\big) + \sum_{\substack{i\in\mathcal{V}\setminus\mathcal{V}_S\\ i\notin L_N}} p_i \sum_{j\in Ch(i)} W_j(1-a) \\
& - \sum_{\substack{i\in\mathcal{V}\setminus\mathcal{V}_S\\ i\notin L_N}} p_i W_m\big((1-a)(1-s_m^0) - C_a\big) - \mathbb{1}_{N_S(m)\neq N} \sum_{\substack{i\in\mathcal{V}\setminus\mathcal{V}_S\\ i\notin L_N}} p_i \sum_{k\in Ch(m)} (1-a)W_k \\
& = \sum_{\substack{i\in\mathcal{V}\setminus\mathcal{V}_S\\ i\notin L_N}} p_i(1-a)\Big(W_i\big(1-\frac{C_a}{1-a}\big) + \sum_{j\in Ch(i)} W_j - \frac{1}{\alpha}\big(Y_A(1-\frac{C_a}{1-a}) + \beta - S\big)\Big) \\
& < 0
\end{aligned}
$$

Therefore, $U_A(p,s) - U_A(p',s) < 0$. The payoff of the attacker is greater when operating on $p^{'}$ instead of p. The attacker attacks only nodes in $\mathcal{V}_S$. □

While proving that the choice of s^0 is valid in Theorem 2.1, we proved that if data on a node j is encrypted with a certain rate, all data handled by each one of its parent nodes will be encrypted with non-zero encryption rates. As a result, we cannot expect all data to be sent in clear between nodes if one of the children of these nodes has encrypted a set of its data.

The sensible target set $\mathcal{V}_S$ is a set of nodes whose security assets are the most attractive to the attacker. To maximize his payoff, the attacker only needs to compromise data processed by these nodes. Any node that does not belong to the sensible target set is not attractive enough for the attacker, and therefore will not be attacked. In this case, from the point of view of the defender, data processed by these nodes does not need to be encrypted. An important security implication of this result is that the defender only needs to secure data processed by nodes in the sensible target set $\mathcal{V}_S$.

2.4.2 One-shot Game

In this section, we investigate the case where both the attacker and the defender take their decisions simultaneously while taking into account each other's strategies. This type of interactions falls under the one-shot game category [OR94]. Let p^* and s^* denote the attacker and the defender strategies at the Nash equilibrium respectively. Therefore, we have:

$$
\begin{aligned}
U_A(p^*,s^*) > U_A(p,s^*)\ \forall p \in \mathscr{P}\ s.t.\ \sum_i p_i \leq P \\
U_D(p^*,s^*) > U_A(p^*,s)\ \forall s \in \mathscr{S}\ s.t.\ \sum_i s_i \leq S
\end{aligned}
$$

Theorem 2.2. *Under the assumption that* $\sum_i p_i = P$ *and* $\sum_i s_i = S$*, a Nash equilibrium exists and is given by:*

$$
\begin{cases}
s_i = 1 - \dfrac{C_a}{1-a} - \dfrac{1}{\alpha W_i}\Big(Y_A\big(1-\dfrac{C_a}{1-a}\big) + \beta - S\Big) + \dfrac{\mathbb{1}_{(N_S(i)\neq N)}}{W_i} \displaystyle\sum_{j\in Ch(i)} W_j & \forall i \in \mathcal{L}(\mathcal{T}_S) \\
s_i = 1 - \dfrac{C_a}{1-a} - \dfrac{A_i}{(1-a)W_i} & \forall i \in L_k \cap (\mathcal{V}_S \setminus \mathcal{L}(\mathcal{T}_S))
\end{cases}
$$

$$\begin{cases} p_1^* = \frac{1}{\gamma}\Big(P - \frac{C_e}{1-a}\sum_{r=2}^{N}\sum_{i\in L_r\cap\mathcal{V}_S}\frac{(1+(-1)^r)}{2} \\ \qquad\qquad -\frac{W_1C_e}{1-a}\sum_{r=2}^{N}\sum_{i\in L_r\cap\mathcal{V}_S}\big(-\frac{1}{W_i}+\mathbb{1}_{(r>2)}\sum_{j=2}^{r-1}\frac{(-1)^{r-j+1}}{W_i^j}\big)\Big) \\ p_i^* = \frac{C_e}{1-a}\Big(\frac{1+(-1)^k}{2}\Big) + p_1^*W_1\Big(\frac{1}{W_i}+\sum_{j=1}^{k-1}\frac{(-1)^{k-j}}{W_i^j}\Big) \\ \qquad\qquad +\frac{W_1C_e}{1-a}\Big(-\frac{1}{W_i}+\mathbb{1}_{(k>2)}\sum_{j=2}^{k-1}\frac{(-1)^{k-j+1}}{W_i^j}\Big) \quad \forall i\in L_k\cap\mathcal{V}_S, k\geq 2 \end{cases}$$

where A_i and γ are given in Appendix E.

and $\mathbb{1}_{expr} = \begin{cases} 1 & if\ expr\ is\ true \\ 0 & otherwise \end{cases}$

Proof. The attacker needs to solve the following optimization problem:

$$\max_p U_A(p,s)\ s.t. \sum_i p_i = P$$

The Lagrangian of this optimization problem is given by:

$$\mathcal{L}_1(p,s,\lambda) = U_A(p,s) + \lambda(P - \sum_i p_i)\ \text{s.t}\ \lambda > 0$$

$\forall i \in \mathcal{L}(\mathcal{T}_S)$,
$$W_i(1-a)(1-s_i) - C_aW_i + \mathbb{1}_{(N_S(i)\neq N)}\sum_{j\in Ch(i)}(1-a)W_j = \lambda$$

$\forall i \in L_k\cap(\mathcal{V}_S\backslash\mathcal{L}(\mathcal{T}_S))$,
$$W_i(1-a)(1-s_i) + \sum_{j\in Ch_S(i)}W_j(1-a)(1-s_j) + \sum_{j\in\overline{Ch_S(i)}}W_j(1-a) - C_aW_i = \lambda$$

Let us assume that $\forall i \in L_k\cap(\mathcal{V}_S\backslash\mathcal{L}(\mathcal{T}_S))$, we have the general formula:

$$\begin{aligned} & W_i(1-a)(1-s_i) - C_aW_i + \sum_{j=k+1}^{N_S(i)}\sum_{m\in L_j\cap\mathcal{L}(\mathcal{T}_S)\cap Ch_S(i,j)}\mathbb{1}_{(j\neq N)}(-1)^{j-k}\sum_{t\in Ch(m)}(1-a)W_t \\ & = \lambda(1+\sum_{j=k+1}^{N_S(i)}(-1)^{j-k}\Delta_i^j) + C_a\sum_{j=k+1}^{N_S(i)}(-1)^{j-k}\sum_{m\in Ch_S(i,j)}W_m \\ & \quad + (1-a)\sum_{j=k+1}^{N_S(i)}(-1)^{j-k}\sum_{\substack{m\in\overline{Ch_S(i,j)} \\ f(m)\in\mathcal{V}_S\backslash\mathcal{L}(\mathcal{T}_S)}}W_m \end{aligned} \tag{2.1}$$

Equation 2.1 is true $\forall i \in L_{N-1} \cap \mathcal{V}_S$. We suppose it is true $\forall i \in L_k \cap (\mathcal{V}_S \backslash \mathcal{L}(\mathcal{T}_S))$. We want to prove that Equation 2.1 is valid $\forall i \in L_{k'} \cap (\mathcal{V}_S \backslash \mathcal{L}(\mathcal{T}_S)), k' = k-1$.

We have $\forall i \in L_{k'} \cap \mathcal{V}_S, k' = k-1$,

$$W_i(1-a)(1-s_i) - C_a W_i + \sum_{j \in Ch_S(i)} W_j(1-a)(1-s_j) + (1-a) \sum_{j \in \overline{Ch_S(i)}} W_j = \lambda$$

$$\Rightarrow W_i(1-a)(1-s_i) - C_a W_i + \sum_{j \in Ch_S(i)} C_a W_j + \sum_{j \in Ch_S(i)} (1-a) \sum_{l=k+1}^{N_S(j)} (-1)^{l-k} \sum_{\substack{m \in \overline{Ch_S(j,l)} \\ f(m) \in \mathcal{V}_S \backslash \mathcal{L}(\mathcal{T}_S)}} W_m$$

$$+ \sum_{j \in Ch_S(i)} \lambda\big(1 + \sum_{l=k+1}^{N_S(j)} (-1)^{l-k} \Delta_j^l\big) + C_a \sum_{j \in Ch_S(i)} \sum_{l=k+1}^{N_S(j)} (-1)^{l-k} \sum_{m \in Ch_S(j,l)} W_m$$

$$+ (1-a) \sum_{j \in \overline{Ch_S(i)}} W_j - \sum_{j \in Ch_S(i)} \sum_{l=k+1}^{N_S(j)} \sum_{m \in L_l \cap \mathcal{L}(\mathcal{T}_S) \cap Ch_S(j,l)} \mathbb{1}_{(l \neq N)} (-1)^{l-k} \sum_{t \in Ch(m)} (1-a) W_t$$

$$= \lambda$$

However

$$\sum_{j \in Ch_S(i)} C_a W_j + C_a \sum_{j \in Ch_S(i)} \sum_{l=k'+2}^{N_S(j)} (-1)^{l-(k'+1)} \sum_{m \in Ch_S(j,l)} W_m$$

$$= C_a \sum_{j \in Ch_S(i)} W_j - C_a \sum_{l=k'+2}^{N_S(i)} (-1)^{l-k'} \sum_{j \in Ch_S(i)} \sum_{m \in Ch_S(j,l)} W_m$$

$$= -C_a \Big(- \sum_{j \in Ch_S(i)} W_j + \sum_{l=k'+2}^{N_S(i)} (-1)^{l-k'} \sum_{m \in Ch_S(i,l)} W_m \Big)$$

$$= -C_a \sum_{l=k'+1}^{N_S(i)} (-1)^{l-k'} \sum_{m \in Ch_S(i,l)} W_m$$

and

$$\lambda - \sum_{j \in Ch_S(i)} \lambda\big(1 + \sum_{l=k+1}^{N_S(j)} (-1)^{l-k} \Delta_j^l\big)$$

$$= \lambda \Big(1 - \sum_{j \in Ch_S(i)} 1 - \sum_{j \in Ch_S(i)} \sum_{l=k'+2}^{N_S(j)} (-1)^{l-k'-1} \Delta_j^l \Big)$$

$$= \lambda \Big(1 - \Delta_i^{k'+1} + \sum_{l=k'+2}^{N_S(i)} (-1)^{l-k'} \Delta_i^l \Big)$$

$$= \lambda \Big(1 + \sum_{l=k'+1}^{N_S(i)} (-1)^{l-k'} \Delta_i^l \Big)$$

$$\text{and} \quad (1-a)\sum_{j\in\overline{Ch_S(i)}} W_j + \sum_{j\in Ch_S(i)}(1-a)\sum_{l=k+1}^{N_S(j)}(-1)^{l-k}\sum_{\substack{m\in\overline{Ch_S(j,l)}\\ f(m)\in\mathcal{V}_S\setminus\mathcal{L}(\mathcal{T}_S)}} W_m$$

$$= -(1-a)\Big(-\sum_{j\in\overline{Ch_S(i)}} W_j - \sum_{l=k'+2}^{N_S(i)}(-1)^{l-k'-1}\sum_{j\in Ch_S(i)}\sum_{\substack{m\in\overline{Ch_S(j,l)}\\ f(m)\in\mathcal{V}_S\setminus\mathcal{L}(\mathcal{T}_S)}} W_m\Big)$$

$$= -(1-a)\Big(-\sum_{j\in\overline{Ch_S(i)}} W_j + \sum_{l=k'+2}^{N_S(i)}(-1)^{l-k'}\sum_{\substack{m\in\overline{Ch_S(i,l)}\\ f(m)\in\mathcal{V}_S\setminus\mathcal{L}(\mathcal{T}_S)}} W_m\Big)$$

$$= -(1-a)\sum_{l=k'+1}^{N_S(i)}(-1)^{l-k'}\sum_{\substack{m\in\overline{Ch_S(i,l)}\\ f(m)\in\mathcal{V}_S\setminus\mathcal{L}(\mathcal{T}_S)}} W_m$$

$$\text{and finally} \quad -\sum_{j\in Ch_S(i)}\sum_{l=k+1}^{N_S(j)}\sum_{m\in L_l\cap\mathcal{L}(\mathcal{T}_S)\cap Ch_S(j,l)} \mathbb{1}_{(l\neq N)}(-1)^{l-k}\sum_{t\in Ch(m)} W_t$$

$$= -\sum_{j\in Ch_S(i)}\sum_{l=k'+2}^{N_S(j)}\sum_{m\in L_l\cap\mathcal{L}(\mathcal{T}_S)\cap Ch_S(j,l)} \mathbb{1}_{(l\neq N)}(-1)^{l-k'-1}\sum_{t\in Ch(m)} W_t$$

$$= \sum_{l=k'+2}^{N_S(i)}\sum_{j\in Ch_S(i)}\sum_{m\in L_l\cap\mathcal{L}(\mathcal{T}_S)\cap Ch_S(j,l)} \mathbb{1}_{(l\neq N)}(-1)^{l-k'}\sum_{t\in Ch(m)} W_t$$

$$= \sum_{l=k'+1}^{N_S(i)}\sum_{m\in L_l\cap\mathcal{L}(\mathcal{T}_S)\cap Ch_S(i,l)} \mathbb{1}_{(l\neq N)}(-1)^{l-k'}\sum_{t\in Ch(m)} W_t$$

Therefore, Equation 2.1 is true $\forall i \in L_k \cap (\mathcal{V}_S\setminus\mathcal{L}(\mathcal{T}_S))$.

$$\text{Let } A_i = \lambda\Big(1+\sum_{j=k+1}^{N_S(i)}(-1)^{j-k}\Delta_i^j\Big) + C_a\sum_{j=k+1}^{N_S(i)}(-1)^{j-k}\sum_{m\in Ch_S(i,j)} W_m$$
$$+(1-a)\sum_{j=k+1}^{N_S(i)}(-1)^{j-k}\sum_{\substack{m\in\overline{Ch_S(i,j)}\\ f(m)\in\mathcal{V}_S\setminus\mathcal{L}(\mathcal{T}_S)}} W_m$$
$$-\sum_{j=k+1}^{N_S(i)}\sum_{m\in L_j\cap\mathcal{L}(\mathcal{T}_S)\cap Ch_S(i,j)} \mathbb{1}_{(j\neq N)}(-1)^{j-k}\sum_{t\in Ch(m)}(1-a)W_t$$

$$\Rightarrow \begin{cases} s_i = 1 - \dfrac{C_a}{1-a} - \dfrac{\lambda}{(1-a)W_i} + \dfrac{\mathbb{1}_{(N_S(i)\neq N)}}{W_i}\displaystyle\sum_{j\in Ch(i)} W_j & \forall i \in \mathcal{L}(\mathcal{V}_S) \quad (a) \\ s_i = 1 - \dfrac{C_a}{1-a} - \dfrac{A_i}{(1-a)W_i} & \forall i \in L_k\cap(\mathcal{V}_S\setminus\mathcal{L}(\mathcal{T}_S)) \quad (b) \end{cases}$$

We have $\sum_i s_i = S$, where $0 \le s_i \le 1\ \forall i$.

Therefore, we find that $\lambda = \frac{(1-a)}{\alpha}\Big(Y_A\big(1 - \frac{C_a}{1-a}\big) + \beta - S\Big)$, where α and β are given in Appendix E.

By substituting λ in Equations (a) and (b), we find the result in Theorem 2.2.

Defender optimization problem:

The defender needs to solve the following optimization problem:

$$\max_s U_D(p,s) \ s.t. \sum_i s_i = S$$

The Lagrangian of this optimization problem is given by:

$$\mathcal{L}_2(p,s,\lambda) = U_D(p,s) + \mu\big(S - \sum_i s_i\big) \text{ with } \mu > 0$$

We consider that the sensible target set $\mathcal{V}_S$ is nonempty. Therefore, at least the root node of the tree belongs to $\mathcal{V}_S$. We refer by 1, the root node of $\mathcal{T}$.

From Definition 2.2, we know that $\forall i \in \mathcal{V}_S,\ f(i) \in \mathcal{V}_S$.

We have $W_1(1-a)p_1 - C_eW_1 = \mu$

$\forall i \in L_k \cap \mathcal{V}_S, k \geq 2,\ W_i(1-a)(p_i + p_{f(i)}) - C_eW_i = \mu$

Let us assume that $\forall i \in L_k \cap \mathcal{V}_S, k \geq 2$, we have the general formula:

$$\begin{aligned} &W_i(1-a)p_i - C_eW_i\Big(\frac{1+(-1)^k}{2}\Big) \\ &= p_1W_1(1-a)\Big(1 + W_i\sum_{j=1}^{k-1}\frac{(-1)^{k-j}}{W_i^j}\Big) + C_eW_1\Big(-1 + \mathbb{1}_{(k>2)}W_i\sum_{j=2}^{k-1}\frac{(-1)^{k-j+1}}{W_i^j}\Big) \end{aligned} \tag{2.2}$$

We note that $W_i^k = W_i,\ \forall i \in L_k$. We want to prove that Equation 2.2 is true $\forall i \in L_{k'} \cap \mathcal{V}_S, k' = k+1$ and $f(i) \in \mathcal{V}_S$.

We have:

$$\begin{aligned} W_i(1-a)p_{f(i)} = \ & p_1W_1(1-a)\Big(\frac{W_i}{W_{f(i)}} + W_i\sum_{j=1}^{k-1}\frac{(-1)^{k-j}}{W_{f(i)}^j}\Big) \\ & + W_1C_e\Big(-\frac{W_i}{W_{f(i)}} + W_i\sum_{j=2}^{k-1}\frac{(-1)^{k-j+1}}{W_f(i)}\Big) + C_eW_i\Big(\frac{1+(-1)^k}{2}\Big) \end{aligned}$$

We know that $W_i^j = W_{f(i)}^j$. Therefore, $\forall i \in L_{k'}$:

$$p_i W_i(1-a) - C_e W_i = p_1 W_1(1-a) - C_e W_1 - p_{f(i)} W_i(1-a)$$

$$\Rightarrow p_i W_i(1-a) + C_e W_i\Big(-1 + \frac{1-(-1)^{k'}}{2}\Big)$$

$$= p_1 W_1(1-a)\Big(1 - \frac{W_i}{W_i^k} + W_i \sum_{j=1}^{k'-2} \frac{(-1)^{k'-j}}{W_i^j}\Big) + C_e W_1\Big(-1 + \frac{W_i}{W_i^k} + W_1 \sum_{j=2}^{k'-2} \frac{(-1)^{k'-j+1}}{W_i^j}\Big)$$

$$\Rightarrow W_i(1-a)p_i - C_e W_i\Big(\frac{1+(-1)^{k'}}{2}\Big)$$

$$= p_1 W_1(1-a)\Big(1 + W_i \sum_{j=1}^{k'-1} \frac{(-1)^{k'-j}}{W_i^j}\Big) + C_e W_1\Big(-1 + W_i \sum_{j=2}^{k'-1} \frac{(-1)^{k'-j+1}}{W_i^j}\Big)$$

We have $\sum_{i \in \mathcal{V}_S} p_i = P$, where $0 \leq p_i \leq 1 \ \forall i$. By substituting the values of p_i in this equation and solving it, we find the result in Theorem 2.2.

□

At the Nash equilibrium (NE), the attacker and the defender have no incentive to deviate from their strategies unilaterally. The NE consists of the optimal acceptable strategies for both players. In the worst case where the attacker has sufficient attack resources, the defender's NE strategy is his best response to the attacker's strategy. As we proved earlier, once the defender chooses to encrypt a set of data on a certain node, he needs to guarantee that the data transiting from this node to the root node is not sent in clear (without encryption). Therefore, the defender's strategy to encrypt data on node i does not only depend on the security asset W_i and the attacker's strategy, but also on the number and security assets of nodes along the path from node i to the root node.

2.4.3 Stackelberg Game

In general, the attacker chooses his strategy based on the deployed security measures in the system. In this section, we analyze the interactions between the attacker and the defender as a Stackelberg game [OR94]. In this type of games, a leader chooses his strategy first. Afterwards, the follower, notified by the leader's choice, chooses his strategy. The leader tries to anticipate the follower's response and chooses the strategy that yields the maximum payoff knowing what will be the reaction of the follower. In our case, the defender is the leader who tries to configure encryption rates on each device in order to protect the confidentiality of the maximum amount of data transiting in the AMI.

Stackelberg games are generally solved by backward induction. The solution is known as Stackelberg Equilibrium (SE). We start by computing the best response strategy of the follower as a function of the leader's strategy. Then, according to the follower's best response, we derive the optimal strategy of the leader.

The attacker solves the following optimization problem:

$$p(s) = \underset{p\in[0;1]^Y}{\operatorname{argmax}} U_A(p,s)$$

On the other hand, the defender solves the following optimization problem:

$$s(p) = \underset{s\in[0;1]^Y}{\operatorname{argmax}} U_D(p(s),s)$$

Theorem 2.3. *The game admits a Stackelberg equilibrium* (p^S, s^S) *given by:*

$$\begin{cases} p_i^S = 0 & \forall i \in \mathcal{V} \\ s_i^S = 1 - \dfrac{C_a}{1-a} & \forall i \in L_N \\ s_i^S = 1 - \dfrac{C_a}{1-a} - \dfrac{C_a \sum\limits_{j=k+1}^{N} (-1)^{j-k} \sum\limits_{m\in Ch(i,j)} W_m}{W_i(1-a)} & \forall i \in L_k, k \leq N-1 \end{cases}$$

Proof. Solving the system by backward induction, we get the best response of the follower given by:

$$p_i = \begin{cases} 1 & if\ (1-a)(1-s_i) - C_a > 0 \\ \in [0;1] & if\ (1-a)(1-s_i) - C_a = 0 \\ 0 & if\ (1-a)(1-s_i) - C_a < 0 \end{cases} \quad \forall i \in L_N$$

and $\forall i \in L_k, k \leq N-1$,

$$p_i = \begin{cases} 1 & if \sum\limits_{j\in Ch(i)} W_j(1-a)(1-s_j) + W_i(1-a)(1-s_i) - C_aW_i > 0 \\ \in [0;1] & if \sum\limits_{j\in Ch(i)} W_j(1-a)(1-s_j) + W_i(1-a)(1-s_i) - C_aW_i = 0 \\ 0 & if \sum\limits_{j\in Ch(i)} W_j(1-a)(1-s_j) + W_i(1-a)(1-s_i) - C_aW_i < 0 \end{cases}$$

The payoff of the defender is given by:

$$U_D(p,s) = -\sum_i \left(p_iW_i(1-a)(1-s_i) + s_iC_eW_i\right) - \sum_{i\notin L_N} \sum_{j\in Ch(i)} p_iW_j(1-a)(1-s_j)$$

We find the results in Theorem 2.3 by noticing that the defender's payoff is a decreasing function with respect to the attacker's strategy p. Therefore, the defender will choose his strategy in order to push the attacker to set his strategy p to 0. Therefore, $\forall i \in L_N$,

$$(1-a)(1-s_i) - C_a = 0 \Rightarrow s_i = 1 - \frac{C_a}{1-a}$$

and $\forall i \in L_k, k \leq N-1$, we prove that:

$$W_i(1-a)(1-s_i) = C_aW_i + C_a \sum_{j=k+1}^{N} (-1)^{j-k} \sum_{m\in Ch(i,j)} W_m$$

Therefore, we find the result in Theorem 2.3.

□

Operating exactly at s^S, the defender is not certain that the attacker will operate at $p^S = 0$. Therefore, in order to push the attacker to choose $p^S = 0$, the defender will operate at a strategy $s_i^{S'}$ slightly higher than s_i^S. In this case, when the defender operates at $s^{S'}$, the attacker will be better off not attacking at all. Otherwise, the attacker will get a negative payoff. The defender strategy $s^{S'}$ is given by:

$$\begin{cases} s_i^{S'} = 1 + \epsilon - \dfrac{C_a}{1-a} & \forall i \in L_N \\ s_i^{S'} = 1 + \epsilon - \dfrac{C_a}{1-a} - \dfrac{C_a \sum\limits_{j=k+1}^{N} (-1)^{j-k} \sum\limits_{m \in Ch(i,j)} W_m}{W_i(1-a)} & \forall i \in L_k, k \leq N-1 \end{cases}$$

where ϵ is a small positive number.

The defender needs additional encryption resources to maintain the Stackelberg equilibrium. However, the gain of adding the additional encryption resources on each node outweighs the potential cost of operating exactly at s^S. Otherwise, the attacker can significantly decrease the payoff of the defender by launching attacks. At the SE, the choice of the encryption rates discourages the attacker from launching any attacks against any node in the system.

Theorem 2.4. *The defender needs at least* $Y(1 - \frac{C_a}{1-a}) - \frac{C_a}{1-a} \sum\limits_{i \in L_k} W_i \sum\limits_{j=1}^{k-1} \frac{(-1)^{k-j}}{W_i^j}$ *encryption resources to discourage the attacker from launching any attacks.*

Proof. Follows directly from Theorem 2.3. □

Theorem 2.4 shows that with sufficient encryption resources, the defender is capable of preventing any attack attempts. In fact, from the point of view of the attacker, the cost of attacking in this case outweighs the potential payoff of attacks.

2.5 Case Study

In this section, we apply our game theoretical framework on an AMI topology as shown in Figure 2.3. In this case study, the number of aggregation levels as defined in our model is 4. Smart meters send consumers' data to the head-end system S_1 to be analyzed. Along the path, data from several smart meters are aggregated at two intermediate levels. On each communication link, different encryption keys or algorithms are used to encrypt outbound data. We consider that on each device in the AMI, we have an IDS with a detection rate a of 0.6. The cost weights C_a and C_e of attacking and encrypting data on a node i are set to 0.2 and 0.05 respectively. The budget P to attack the system is set to 1 while the total budget S of the defender to encrypt data is set to 8. In this section, we analyze the behavior of the attacker and the defender in the cases of the one-shot and the Stackelberg games. The results are depicted in Table 2.2.

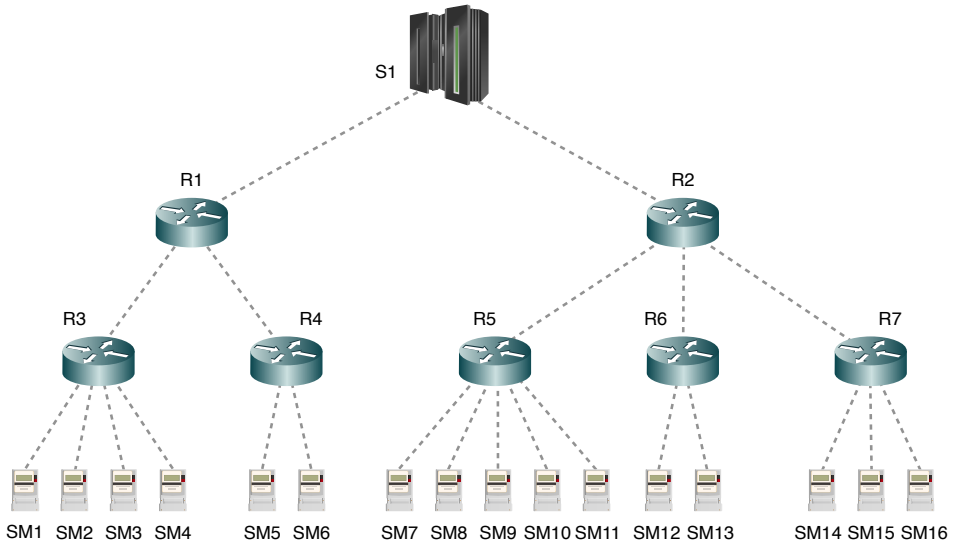


FIGURE 2.3: Example of an AMI architecture

2.5.1 One-Shot Game

In this type of games, both players choose their strategies at the same time. Nodes with security assets that are not attractive to the attacker are *self-protected*. The defender does not need to encrypt data on these nodes (SM1, SM2, etc.) since they will not be attacked.

We notice that most of the time, the defender's deployed defense resources on a node is an increasing function with respect to the security asset of that node. In addition to the value of the security asset, the topology of the network affects the strategy of the defender. For example, the security asset of router R2 is double than R1. However, the defender does not allocate twice as much resources to encrypt data on R2. At the smart meters level, we notice that the defender chooses the same data encryption rates on nodes with the same security assets values. However, in addition to security assets values, the attacker takes into account interconnections between devices and whether smart meters share the same parent node at level $N-1$ (3^{rd} level). For example, SM11 and SM14 have the same security asset value of 3 but do not share the same parent node at the 3^{rd} level. We notice that the attacker does not deploy the same amount of attack resources on SM11 and SM14 even though data encryption rates chosen by the defender at the NE are the same. Finally, both players do not evaluate nodes that belong to different aggregation levels but with the same security assets values in the same way. For example, R4 and SM9 have the same security asset value of 4, however both players treat each node differently.

2.5.2 Stackelberg Game

In the Stackelberg game, we have a leader and a follower. First, the leader chooses

TABLE 2.2: Nash and Stackelberg equilibriums of AMI data confidentiality games

	$\mathbf{W_i}$	**One-Shot game**		**Stackelberg game**	
		p^*	s^*	p^S	s^S
S1	65	0.1267	0.9316	0	0.7731
R1	20	0.0039	0.4618	0	0.6625
R2	40	0.0011	0.5361	0	0.725
R3	14	0.1291	0.9643	0	0.8214
R4	6	0.1398	1	0	0.875
R5	29	0.1278	0.9583	0	0.8103
R6	4	0.1519	0.809	0	0.8125
R7	15	0.1314	0.9509	0	0.8167
SM1	1	0	0	0	0.5
SM2	2	0	0	0	0.5
SM3	1	0	0	0	0.5
SM4	5	0.0183	0.2472	0	0.5
SM5	3	0.0225	0.0787	0	0.5
SM6	1.5	0	0	0	0.5
SM7	1	0	0	0	0.5
SM8	4	0.0251	0.184	0	0.5
SM9	6	0.0159	0.2894	0	0.5
SM10	4	0.0251	0.184	0	0.5
SM11	3	0.0345	0.0787	0	0.5
SM12	1	0	0	0	0.5
SM13	1.5	0	0	0	0.5
SM14	3	0.0309	0.0787	0	0.5
SM15	5	0.016	0.2472	0	0.5
SM16	1.5	0	0	0	0.5

his strategy. Then, the follower, informed by the leader's choice, chooses his strategy accordingly. In our case, the defender is the leader who tries to anticipate attacker's actions and configure encryption rates on each device to reduce the amount of data that can be accessed by the attacker. In addition to the security asset, the cost of attacking and the network topology play an important role in the choice of encryption rates in this case. For example, interestingly, encryption rates for data sent from nodes in the 3^{rd} aggregation level to R1 and R2 are higher than encryption rates for data sent from R1 and R2 to S1. Moreover, encryption rates are not proportional with respect to the security assets of the nodes ($s_{R2} \neq 2 \times s_{R1}$). Smart meters are treated in the same way regardless of their security assets. However, this choice of encryption rates is sufficient to discourage the attacker from launching any attacks. Finally, we note that in order to maintain this Stackelberg equilibrium, the defender needs at least a budget of 14.297.

2.6 Conclusion

In this chapter, we analyzed data confidentiality attacks on smart grid Advanced Metering Infrastructure (AMI) network components. We modeled the interaction between an attacker and a defender as a non-cooperative game. The objective of the attacker is to collect the maximum amount of data about consumers by attacking devices in the AMI, whereas the defender tries to protect the data by encrypting it using different encryption keys or encryption algorithms for each network link. In this chapter, we derived the expected behavior of the attacker and the defender for two types of interactions between the two players. Based on our analysis, we identified the most profitable set of devices to compromise. In a leader and a follower game, where the defender anticipates attacker's actions, we derived the minimum defense budget and the optimal encryption rates on each device in the AMI that are required to thwart attacks. Finally, we showed via a case study how to apply our game framework to configure encryption rates on network devices in the AMI.

To provide the smart metering, among other services, the smart grid relies on a communication infrastructure. In general, this infrastructure is also used to control the different equipment in the power grid. However, the interdependency between the communication infrastructure and the power grid renders the security risk management in the smart grid a challenging task. In the next chapter, we present a game theoretical model for identifying and hardening the most critical communication equipment used in the power grid. By analyzing the interactions between the attacker and the defender, we derive the optimal distribution of defense resources on communication equipment that minimizes the risk of attacks on the power system.

Chapter 3

A Game Theoretical Model for Security Risk Management of Interdependent ICT and Electrical Infrastructures

In the last decade, the power grid has increasingly relied on the communication infrastructure for management and control of grid operations. The communication infrastructure, which can include equipment using off-the-shelf vulnerable operating systems, has the potential to increase the attack surface of the power system. The interdependency between the communication and the power system renders the management of the overall security risk a challenging task. In this chapter, we address this issue by presenting an analytical model for identifying and hardening the most critical communication equipment used in the power system. Using non-cooperative game theory, we model the interactions between an attacker and a defender. We derive the minimum defense resources required and the optimal strategy of the defender that minimizes the risk on the power system. We propose a method to assess the values of the parameters of the analytical model used to evaluate the impact of equipment failures in the power system. Finally, we validate our model via a case study based on the polish electric power transmission system.

3.1 Introduction

In the last few decades, critical infrastructures, such as transportation systems and power grids, have been essential to the development of nations' economies. The increased interdependency between these systems could have unintended cascading effects as a result of a failure or an attack on one of these infrastructures [RB06]. The power grid stands as

one of the most important critical infrastructures on which depends an array of services. Therefore, its resilience against failures and cyber attacks needs to be improved.

The increased dependence of the smart grid on ICT (Information and Communications Technology) will potentially expose it to additional threats. An attack on a communication equipment used to control an industrial process can have severe impact on critical infrastructures [Pou03]. Reciprocally, an electrical node responsible of providing power to a set of communication equipment is important to the communication infrastructure: if the power source of these equipment is compromised, the communication nodes will not be able to achieve their objectives. For instance, the electric blackout that affected Italy in September 2003 showed a bi-directional functional dependency between electrical and communication systems [RIT+08]. Throughout this chapter, the communication system refers to the telecommunication infrastructure responsible of controlling and monitoring the electrical system.

There exist 4 types of dependencies between the nodes in electric and communication systems [SCH13]:

- Type 1: From an electrical node to another electrical node. This type of relation models the flow of electricity between the nodes in the electrical system.
- Type 2: From a communication node to another communication node. This type of relation models the flow of information between the nodes in the communication infrastructure.
- Type 3: From an electrical node to a communication node. This relation models the electrical node that ensures the power supply to the communication node.
- Type 4: From a communication node to an electrical node. This relation models commands sent by the communication node to control the electrical node.

Traditionally, the reliability of the power grid and the security of the ICT infrastructure were assessed independently using different methodologies, for instance [Wen05] and [ANS10] respectively for electric and ICT infrastructures. More recently, a growing body of research has been dedicated to the modeling of interdependencies in critical infrastructures, focusing in particular on communication and electric systems. The objective is to assess the impact of cyber attacks occurring on communication equipment on the power grid. Behavioral or simulation-based models are one of the main categories of models proposed to analyze cascading failures and study blackout dynamics. In this category, different techniques were used including agent-based models [CGT07], petri nets [CSAB11], and co-simulation [LVS+12]. In some of these approaches, attacks targeting communication equipment are explicitly modeled to assess their impact on the power grid. However, the choice of the level of abstraction used to represent system components affects the nature and the type of interdependencies that will be investigated and their potential impact on the behavior of the system.

Along this line of research, we propose an analytical model based on game theory for quantifying the impact of interdependencies between electric and communication infrastructures. The model aims at identifying the most critical communication equipment used in the power system that must be hardened, and generating in this case the optimal distribution of defense resources on these equipment. Due to the abstract nature of such analytical models, assessing their relevance in real-world scenarios is a challenging task. In particular, applying the model on a case study assumes a correct evaluation of the values of its parameters. In this chapter, we propose a methodology for assessing the values of the parameters of the model related to the power grid infrastructure.

The analytical model is validated on a case study based on the polish electric transmission system. We use the publicly available dataset of the polish power grid at a peak load in the summer of 2004 provided in the MATPOWER computational package [ZMST11]. The dataset consists of 420 generators and 3504 transmission lines. We follow a similar approach to [PTC11] to compute cascading failures in the power grid by solving power flow equations using the DC power flow approximation [Zhu09]. To achieve this objective, we simulate individual failures and assess their impact on the power grid such as identifying generators with insufficient capacity to meet the demand and overloaded lines. Using this approach, we quantify the electric system parameters used in our analytical model.

This chapter is organized as follows. We start by discussing related work in Section 3.2. In Section 3.3, we present different factors that are used to assess the initial risk on each equipment in the communication and power infrastructures. We present our interdependency model in Section 3.4. In Section 3.5, we model the risk diffusion process of attacks between the nodes of the communication and the electric infrastructures. We define in Section 3.6 a security game between an attacker who tries to compromise communication equipment to cause the maximum impact on the power grid, and a defender whose objective is to protect the power system by hardening the security on communication equipment, while taking into account the existence of backup control equipment in the communication infrastructure. We prove the existence of a solution and solve the game analytically. In Section 3.7, we propose a method to evaluate the values of parameters used in the analytical model to assess the impact of failures in the power grid. We validate our model via a case study based on the polish power grid depicting interdependencies between the electric transmission system and its control network, and show how our framework can be applied to find optimal defense strategies that reduce the impact of attacks on the power system. Finally, we conclude the chapter in Section 3.8.

3.2 Related Work

For the past decade, the impact of failures and attacks in a single infrastructure has been extensively studied. For example, different risk assessment methods were used to analyze

the impact of failures and attacks on the electric and communication systems (see for example [Wen05] and [ANS10]). However, the complex interactions between interdependent infrastructures, such as the case of the electric and communication systems, render this type of analysis incomplete. In many scenarios, it could fail to assess the impact of attacks originating from one infrastructure to affect equipment and services in other infrastructures. In particular, the increased interdependencies between the power grid and its control system made the assessment of the impact of cyber attacks more challenging.

In an attempt to study the impact of these interdependencies, a number of methods and models were proposed. For example, Laprie et al. [LKK07] propose a qualitative model to address cascading, escalating, and common cause failures due to interdependencies between the electric and information infrastructures. In the case of quantitative models, we can distinguish two main categories used to analyze interdependent systems: analytical-based and simulation-based models. In the first category of models, we find the work of Buldyrev et al. [BPP$^+$10]. The authors develop a theoretical framework to study the process of cascading failures in interdependent networks caused by random initial failures of nodes. In [HGB$^+$11], a mathematical framework was developed to assess the robustness of interdependent networks under targeted attacks which depends on nodes' degrees. In simulation-based models, the main techniques that are used include agent-based [CGT07], petri nets [CSAB11] and co-simulation [LVS$^+$12]. In [TLM08] for example, Ten et al. use probabilistic methods based on Petri-nets to identify weaknesses in the control infrastructure in the power grid. The impact of an attack is evaluated by the potential loss of load in the power system.

Another set of related work regarding the analysis of the impact of electric and communication systems interdependencies use Stochastic Activity Network (SAN) formalism, which is a generalization of Stochastic Petri Nets. Using this formalism, Chiaradonna et al. [CDN11] model the electric grid and its control network, each organized as a set of interconnected regions. Beccuti et al. [BCG$^+$12] use a SWN (Stochastic Well-formed Nets) and a SAN to model the communication and electric systems respectively. However, their study is limited to the effect of a DoS attack on the communication system to affect the power system. Bloomfield et al. [BCP$^+$10] proposed a method and developed a tool to analyze the interdependencies that exist between critical infrastructures. Based on a preliminary description of services and their interdependencies (deterministic or stochastic), an execution engine based on the tool Möbius [Mob] simulates the model. The authors use a Monte Carlo simulation to quantify the impact of interdependencies on the behavior of the system.

In addition to modeling interdependencies between the electric and communication infrastructures, it is important to identify the critical equipment in the communication system of the power grid [SWB04]. In particular, identifying critical components in the control system whose compromise could lead to total blackouts must be performed [PM13]. Hardening these equipment is an essential step to protect the power grid. A number of solutions were proposed to improve the resiliency of the power system. Qi et al. [QWT$^+$11] achieve this objective by adding intelligent equipment to create a reconfigurable grid. Along this research direction, M. Amin [Ami01] proposes a layered architecture based on intelligent agents that

adapt to events and surroundings. These intelligent devices ensure that the system is dependable, robust, and can self-heal. Anwar et al. [AMGC09] addressed the issue of choosing an optimal combination of security hardening schemes to secure control networks for critical infrastructures under a certain defense cost budget. However, it is also important to evaluate the effect of coordinated attacks [SGL12] and assess the impact of the security solutions on the reliability of the cyber-physical system [MC13].

In complex interdependent systems, interactions between the attacker and the defender play an important role in defining the optimal defense strategy. In this context, game theory offers a mathematical framework to study interactions between different players with the same or conflicting interests. This theory has already been applied to assess the security of interdependent systems. For example, Law et al. [LAP12] investigate false data injection attacks on the power grid and formulate the problem as a stochastic security game between an attacker, trying to choose the intensity of false data injection, and a defender trying to determine the detection threshold level. Amin et al. [ASH13] present a framework to assess risks to cyber-physical systems when interdependencies between information and physical systems may result in correlated failures. They formulate the problem of security choices of the individual players as a non-cooperative game. After choosing his or her security strategy, each player chooses a control input sequence to maintain optimal closed-loop performance.

In our work, we make a number of assumptions on the capability of the attacker. For example, we suppose that the attacker knows the topology of the power grid. Even though this assumption does not always hold and depends on the profile of the attacker, Li et al. [LPS13] showed that an attacker, with access to limited data, can learn the topology of the power system. In this chapter, we present a quantitative model to assess the impact of cyber attacks on the power grid. Our analytical approach allows us to study the efficiency of hardening the security on a set of communication equipment in reducing the impact of attacks on the power grid. By using game theory to analyze the behavior of the attacker and the defender, and formally proving the existence and uniqueness of a NE, the defender takes into account attacker's actions and is confident of optimizing the distribution of his defense resources on critical communication equipment that most impact the power system and are likely to be the targets of attacks. The structure of player's utility functions, taking into account the existence of backups in the communication system, allows us to characterize analytically players' strategies at NE. Therefore, we are able to evaluate potential changes in the behavior of players to estimation errors on the values of a set of model parameters.

3.3 Initial Risk

There are multiple risk analysis methods designed for information systems risk assessment. These methods classify threats and define security objectives that are generally to ensure the integrity, confidentiality, and availability of data or communications. However, in general, such methods are not designed to assess risks on communication equipment in the

electrical system due to the interdependency that exists between the two infrastructures. The electrical system main objective is to ensure that electricity is delivered without service disruptions. The integrity of data, used to estimate the state of the power system, needs to be guaranteed. The combination of the availability and the integrity of data are essential to ensure the dependability and availability of the power grid. The electrical system uses a Supervisory Control and Data Acquisition (SCADA) system to monitor and control electrical equipment in the power system. SCADA uses several telecommunication infrastructures such as telephone lines, cellular networks, etc. to send data to a control center to be analyzed. This renders the power system dependent on the reliability and security of the telecommunication system.

In the electric grid, the impact of attacks on an electrical node depends, among other factors, on the nature of the node (e.g. generator, transformer, load). We refer by initial risk, the risk on a node before an attack or an accident occurs and its impact propagates between system nodes. Several methods exist to assess the risk of faults in the power system. For example, PROMAPS [PRO] calculates the probability and the financial consequences of fault conditions in the power system. However, deliberate attacks on control equipment can have severe impact on the power grid. Therefore, this type of events needs to be taken into account when assessing the risk on the power system. Different factors affect the initial risk $r_i^e(0)$ on an electric equipment i such as the power P generated/consumed by the node, the cost of recovery in the event of a failure, the number of affected customers if the node fails, etc.

The communication infrastructure is critical in today's power systems. On the other hand, communication equipment need electric power to function. Therefore, the risk on communication equipment should take into account the impact of compromised equipment in the power system. Similarly to electric nodes, we consider an initial risk $r_j^c(0)$ on the communication equipment j. As for $r_i^e(0)$, we do not provide a definition for computing $r_j^c(0)$. However, factors that may affect its value include the criticality/importance of electrical nodes' data processed by j, the number of electric equipment it controls, etc.

In this chapter, we assume that initial risk on a system node is a nonnegative real number and has been evaluated using risk assessment methods. We are interested in the risk diffusion process between nodes in the same infrastructure as well as between nodes of different infrastructures.

3.4 Interdependency Model

We use the framework in [AB09] as a basis to represent the risk dependencies using a graph-theoretic approach. We model the interdependency between the electrical and the communication infrastructures as a weighted directed interdependency graph $\mathcal{D}$. The graph $\mathcal{D}$ is defined as the triplet (V, E, f). $V = \{v_1, v_2, ..., v_N\}$ is a finite set of vertices representing the set of electrical and communication nodes. E is a particular subset of V^2 and referred

to as the edges of $\mathcal{D}$. Finally, $f : E \to \mathbb{R}^+$ is a function where $f(e_{ij})$ refers to the weight associated with the edge e_{ij}.

Let $V = \{\mathcal{T}^e, \mathcal{T}^c\}$ where $\mathcal{T}^e = \{v_1, v_2, ..., v_{N_e}\}$ represents the set of electrical nodes in the grid and $\mathcal{T}^c = \{v_{N_e+1}, v_{N_e+2}, ..., v_{N_e+N_c}\}$ represents the set of communication nodes.

Let $\mathcal{D}$ be represented by the weighted adjacency matrix $M = [m_{ij}]_{N\times N}$ defined as follows:

$$M = \begin{pmatrix} B & D \\ F & S \end{pmatrix}$$

$$\text{where} \begin{cases} B = [b_{ij}]_{N_e\times N_e} & s.t \ \sum\limits_i b_{ij} = 1 \ \forall j \\ D = [d_{ij}]_{N_e\times N_c} & s.t \ \sum\limits_i d_{ij} = 1 \ \forall j \\ F = [f_{ij}]_{N_c\times N_e} & s.t \ \sum\limits_i f_{ij} = 1 \ \forall j \\ S = [s_{ij}]_{N_c\times N_c} & s.t \ \sum\limits_i s_{ij} = 1 \ \forall j \end{cases}$$

Matrix M represents the effects of nodes on each other and is a block matrix composed of matrices B, D, F and S. Elements of these matrices are nonnegative real numbers. Without loss of generality, we assume that these matrices are left stochastic matrices. Therefore, for each node k, we evaluate the weight of other nodes to impact node k. Matrix B represents the dependency between electrical nodes. Each element b_{ij} of B represents the impact of the failure of electrical node i on electrical node j. Dependencies between communication nodes are represented in matrix S. Control engineers use the communication infrastructure to observe the state of the power system. An incident or attack on a set of communication nodes could have severe impact on power system control data routing and analysis. In addition, a failure of electric equipment can deprive communication equipment from their main power supply. We introduce matrices D and F to represent the dependency relation on communication nodes by electric nodes and vice versa respectively. M represents the effect of an accident or an attack on a node to impact nodes of both communication and electric infrastructures.

3.5 Risk Diffusion and Equilibrium

In this section, we are interested in computing the risk on communication equipment after an attacker compromises a set of nodes in the communication system. We consider that the first cascading effects of an attack on communication equipment take place in the communication infrastructure itself. Afterwards, the impact of the attack propagates to the electric system. Finally, the failures in the power grid will affect the power supply of communication nodes.

In the communication system, we consider that a set of Intrusion Detection Systems (IDSs) exists. We assume that devices that assure a security function such as IDSs, have security mechanisms protecting the availability of their function. The attacker tries to

compromise a set of communication nodes in order to control or disrupt the power system. The probability of being detected increases each time the attacker attempts to compromise a new equipment. Therefore, we consider that the payoff of future attacks decreases at each attack step. Let γ_c be a nonnegative real number that represents the weight of the impact payoff of future attacks s.t $\gamma_c \in [0, 1]$. γ_c is a function of the probability of detection of the IDS and attacker's profile. For example, an insider attacker could possess credentials that enable him to legitimately access control equipment without drawing suspicions.

We introduce a metric t_c in the communication system that defines the scale of attacks' impacts propagation between the system's nodes. The average propagation time t_c in the communication system is the average time for the impact of an attack on communication equipment to propagate in the communication infrastructure.

Let $R^e(t) = [r_i^e(t)]_{N_e \times 1}$ and $R^c(t) = [r_i^c(t)]_{N_c \times 1}$ be the electrical and communication nodes risk vectors at time t respectively. We take discrete time steps to describe the evolution of the system.

Let $S^l = [s_{ij}^l]_{N_c \times N_c}$ be the l-th power of the matrix S. We are interested in computing the maximum impact of an attack on communication equipment to reach communication nodes during time t_c. Let the matrix $S^{max} = [s_{ij}^{max}]_{N_c \times N_c}$ represent this maximum impact, where $s_{ij}^{max} = \max_{l=1,\dots,\lfloor t_c \rfloor} \gamma_c^l s_{ij}^l$. The overall impact on node j, given a specific attack path, depends on the number of equipment the attacker needs to compromise to impact node j. At attack step r, the payoff is decreased by a factor of γ_c^r. In fact, we consider that each action of the attacker in the system increases the probability of him being detected. Therefore, γ_c^r represents the uncertainty for the attacker of getting the payoff of the r^{th} future attack step. Let S_n^{max} be the normalized matrices of S^{max} with respect to their rows s.t. $\forall j,\ \sum_i {s_n^{max}}_{ij} = 1$.

Therefore, the system of equations for inter- and intra-infrastructure risk diffusion is given by:

$$\begin{cases} R^c(t+1) = S_n^{max} R^c(t) \\ R^c(t+1) = F R^e(t) \\ R^e(t+1) = B R^e(t) \\ R^e(t+1) = D R^c(t) \end{cases} \tag{3.1}$$

Solving the system of equations in 3.1, we will have:

$R^c(t+4) = S_n^{max} FBD R^c(t) = H R^c(t)$ where $H = [h_{ij}]_{N_c \times N_c} = S_n^{max} FBD$.

Lemma 3.1. *Matrix $H = S_n^{max} FBD$ is a left stochastic matrix.*

Proof. Let $Z = [z_{ij}]_{m\times n}$ and $Y = [y_{ij}]_{n\times m}$ s.t $\forall j,\ \sum_i z_{ij} = 1$ and $\sum_i y_{ij} = 1$. Let $X = [x_{ij}]_{m\times m} = ZY$. Therefore:

$$\sum_i x_{ij} = \sum_i \sum_m z_{im} y_{mj} = (\sum_m y_{mj})(\sum_i z_{im}) = \sum_m y_{mj} = 1$$

Similarly, we can prove that matrix H, which is the product of matrices S_n^{max}, F, B and D, is a left stochastic matrix. □

We take a similar approach to [AB09] by balancing the immediate risk and the future induced one. The value of risk on communication equipment at a given time is defined as:

$$R^c(t+4) = \delta H R^c(t) + \beta R^c(0) + \theta D^T R^e(0) \tag{3.2}$$

In Equation 3.2, β, θ and δ are nonnegative real numbers and $\beta + \theta + \delta = 1$. β and θ represent the weight of the initial risk on communication nodes and the weight of the diffused risk from electric equipment to communication equipment at time $t = 0$ respectively. Finally, δ reflects the weight of future cascading risk w.r.t. the value of the total risk on communication equipment. In fact, in the power system, different safety and control measures ensure that failures in the electric system do not propagate through the entire grid. In our model, by balancing the immediate risk and the future induced one, we can take into account this assumption. We can notice that at each iteration of Equation 3.2, the weight of future risk decreases w.r.t. the value of risk on communication equipment at $t = 0$.

Theorem 3.1. *The iterative system of the cascading risk converges. An equilibrium solution exists whenever $\delta < 1$ and is given by:*

$$R^{c*} = (I - \delta H)^{-1}(\beta R^c(0) + \theta D^T R^e(0)) \quad where\ H = S_n^{max} F B D \tag{3.3}$$

Proof. From Lemma 3.1, we know that H is a left stochastic matrix. The spectral radius of any matrix is less than or equal to the norm of the matrix. The 1-norm of the matrix $H = [h_{ij}]_{N_c\times N_c}$ is defined as $\|H\|_1 = \max_{0\le j\le N_c} \{\sum_{i=1}^{N_c} |h_{ij}|\}$. The matrix H is a left stochastic matrix. Therefore, $\|H\|_1 = 1$ and the spectral radius $\rho(H) \le 1$. The matrix H has at least one eigenvalue equals to 1 since (1,e) is an eigenpair of H^T (where $e = [1...1]^T$). Since the matrix H is multiplied by $\delta < 1$, so as the eigenvalues of H. Therefore, the sequence converges. The equation of the cascading risk $R^c(t+4) = \delta H R^c(t) + \beta R^c(0) + \theta D^T R^e(0)$ converges to the value R^{c*} given by $R^{c*} = \delta H R^{c*} + \beta R^c(0) + \theta D^T R^e(0)$.

The solution of the problem is given by: $\lim_{t\to+\infty} R^c(t) = (I - \delta H)^{-1}(\beta R^c(0) + \theta D^T R^e(0))$. The existence of the solution depends on the existence of the inverse of the matrix $(I - \delta H)$.

However, we can notice that: $\mid 1 - \delta h_{ii} \mid > \mid \delta \sum_{i \neq j} h_{ij} \mid = \mid \delta - \delta h_{ii} \mid \forall i$ is true whenever $\delta < 1$. In this case, the matrix $(I - \delta H)$ is a strictly column diagonally dominant matrix, and therefore nonsingular. As a result, $(I - \delta H)^{-1}$ exists. □

From Theorem 3.1, we can predict how the risk on communication equipment diffuses between nodes of the communication and electric systems. If an attacker has access to H, he can choose his targets in the communication system intelligently to maximize the impact of his attacks on the power system. In the next section, we propose a security game between an attacker and a defender and analyze the behavior of both players in this scenario.

3.6 Security Game

The use of communication equipment has the potential to increase the attack surface of the power system. Attacks on the communication system could have severe impact on the power grid. It is conceivable that an attacker could exploit vulnerabilities in the strategy of the defender to compromise communication equipment that control electric equipment. In this section, we analyze the expected behavior of a rational attacker and derive the optimal strategy of the defender. We formulate the problem as a non-cooperative game and analyze the behavior of the attacker and the defender at the Nash equilibrium. The attacker's/defender's objective is to distribute attack/defense resources on the communication nodes in order to maximize/minimize the impact of attacks on the power system. We consider a perfect information game. In addition, we consider the worst-case scenario where both players have complete knowledge of the architecture of the system.

The attacker's strategy is a vector $p = [p_i]_{1 \times N_c}$, where each $0 \leq p_i \leq 1$ is the attack resource allocated to target $i \in \mathcal{T}^c$. The defender's strategy is a vector $q = [q_i]_{1 \times N_c}$, where each $0 \leq q_i \leq 1$ is the defense resource allocated to target $i \in \mathcal{T}^c$. We can interpret p_i (resp. q_i) as the probability that the attacker (resp. defender) attacks (resp. defends) communication node i. We assume that the cost of attacking and defending a communication node i are proportional to the risk on node i and are given by $c_i^a r_i^c(0)$ and $c_i^d r_i^c(0)$ respectively, where $0 \leq c_i^a, c_i^d \leq 1$.

We associate for each communication equipment a load l_i that represents the amount of computational work the equipment performs. Let $L = diag(l_i)_{N_c \times N_c}$ be the load matrix. In general, the power utility assigns a set K_i of communication nodes to be the backup of another set K_j if equipment in K_j were compromised or became unreachable. The existence of redundant equipment in the communication system increases the resilience of the power grid against cyber attacks. Let $W = [w_{ij}]_{N_c \times N_c}$ be the redundancy matrix where $\forall i, w_{ii} = -1$ and $\sum_{j, j \neq i} w_{ij} \leq 1$. If $i \neq j$, w_{ij} represents the fraction of the load of node i, node j will be responsible of processing when node i is compromised. In fact, control centers rely on a telecommunication infrastructure to communicate. A telecommunication carrier often

manages this infrastructure. A failure in the power system could impact communications between control centers, therefore affecting the possibility that redundant equipment take charge of the load of compromised communication equipment. This scenario should be taken into account when evaluating the impact of the existence of redundant equipment on the utilities of the attacker and the defender.

The utilities U_a and U_d of the attacker and the defender respectively are as follows:

$$U_a(p,q) = pR_D^{c*}(e^T - q^T) - pR_D^c(0)C^a p^T - \psi pL(Wq^T - I(e^T - 2q^T))$$

$$U_d(p,q) = -pR_D^{c*}(e^T - q^T) - qR_D^c(0)C^d q^T + \psi pL(Wq^T - I(e^T - 2q^T))$$

$R_D^c(0) = diag(r_i^c(0))_{N_c \times N_c}$, $R_D^{c*} = diag(r_i^{c*})_{N_c \times N_c}$, $C^a = diag(c_i^a)_{N_c \times N_c}$, and $C^d = diag(c_i^d)_{N_c \times N_c}$ are diagonal matrices, I is the identity matrix and $e = (1, ..., 1)_{1 \times N_c}$. The players' utilities are composed of three parts:

- Payoff of an attack taking into account both players' actions and the cascading impact of the attack in the communication and electric systems
- Cost of attacking/defending
- Impact of redundant equipment in ensuring the control of the power system when a set of communication nodes is compromised. $\psi \in [0, 1]$ is a parameter that represents the impact of the existence of backup equipment in computing players' utility functions. ψ is a function of the probability that backup equipment are able to take charge of the load of compromised communication equipment.

In the context of non-cooperative games, we are interested in the concept of Nash equilibrium (NE), in which none of the players has an incentive to deviate unilaterally [OR94]. The Nash equilibrium is considered as the most profitable strategy profile that maximizes each player's utility given the actions of other players. Let $p = (p_1, ..., p_{N_c}) \in \mathscr{P}$ and $q = (q_1, ..., q_{N_c}) \in \mathscr{Q}$ be the strategy profiles of the attacker and the defender respectively, where $\mathscr{P}$ and $\mathscr{Q}$ refer to the strategy spaces of each player. We define the Nash equilibrium as follows:

Definition 3.1 (Nash equilibrium)**.** *A* Nash equilibrium *is a strategy profile (p^*, q^*) in which each player cannot improve his utility by altering his decision unilaterally.*

3.6.1 One-shot Game

We investigate the case where both the attacker and the defender take their decisions at the same time while taking into account each other's strategies. This type of interactions falls under the one-shot game category [OR94].

Let p^* and s^* denote the attacker and defender strategies at the Nash equilibrium respectively. Therefore, we have:

$$U_A(p^*, q^*) > U_A(p, q^*)\ \forall p \in \mathscr{P}$$
$$U_D(p^*, q^*) > U_D(p^*, q)\ \forall q \in \mathscr{Q}$$

Theorem 3.2. *A unique Nash equilibrium of the game exists and is given by:*

$$q^* = \frac{1}{2}e(R_D^{c*} + \psi L)(R_D^c(0)C^a)^{-1}M[\frac{1}{2}M^T(R_D^c(0)C^a)^{-1}M + 2R_D^c(0)C^d]^{-1} \tag{3.4}$$

$$p^* = e(R_D^{c*} + \psi L)[\frac{1}{2}M(R_D^c(0)C^d)^{-1}M^T + 2R_D^c(0)C^a]^{-1} \tag{3.5}$$

$$where \quad M = R_D^{c*} + \psi L(W + 2I)$$

Proof. Let $\overline{\nabla}$ be the pseudogradient operator of $U = U_a(u) + U_d(u)$ where $u = [p\ q]$.

$$g(u) = \overline{\nabla}U = \begin{bmatrix} \nabla_p U_a(u) \\ \nabla_q U_d(u) \end{bmatrix}$$

Let $G(u)$ be the Jacobian of $g(u)$.

$$G(u) = \left(\begin{array}{c|c} -diag(2r_i^c(0)c_i^a) & -R_D^{c*} - \psi(W^T + 2I)L \\ \hline R_D^{c*} + \psi L(W + 2I) & -diag(2r_i^c(0)c_i^d) \end{array}\right)$$

We suppose that $r_i^c(0)c_i^a \neq 0$ and $r_i^c(0)c_i^d \neq 0\ \forall i$. Therefore $(G(u) + G(u)^T)$ is a negative definite matrix. As a result, U is diagonally strictly concave. Based on [Ros65], an equilibrium of the game in pure strategy exists and is unique.

To characterize the equilibrium, we need to find vectors p^* and q^* in which the gradients ∇U_a and ∇U_d are equal to 0. Solving these equations, we find q^* and p^* given in Equations 3.4 and 3.5 respectively.

Let $M = R_D^{c*} + \psi L(W + 2I)$. The existence of p^* and q^* depend on the existence of the inverses of matrices ξ and ζ, where:

$$\xi = \tfrac{1}{2}[M(R_D^c(0)C^d)^{-1}M^T + 4R_D^c(0)C^a]$$
$$and \quad \zeta = \tfrac{1}{2}[M^T(R_D^c(0)C^a)^{-1}M + 4R_D^c(0)C^d]$$

The diagonal matrix $4R_D^c(0)C^a$ is a positive definite matrix (diagonal matrix with strictly positive elements). To prove that $M(R_D^c(0)C^d)^{-1}M^T$ is a positive definite matrix, we need to show that:

$$\forall x \neq 0,\ x^T M(R_D^c(0)C^d)^{-1}M^T x > 0$$

Let $y = M^T x$. Therefore, we need to prove that:

$$\forall y \neq 0,\ y^T(R_D^c(0)C^d)^{-1}y > 0 \tag{3.6}$$

However, $(R_D^c(0)C^d)^{-1}$ is a positive definite matrix, and Equation 3.6 holds. Therefore, the matrix $M(R_D^c(0)C^d)^{-1}M^T$ is a positive definite matrix. Finally, the matrix ξ is a positive definite matrix because it is the sum of two positive definite matrices. Since ξ is a positive definite matrix, the inverse ξ^{-1} exists. Similarly, we prove that ζ^{-1} exists. □

The analytical solution has multiple advantages. From a scalability point of view, the complexity resides in evaluating the input parameters of the model. In fact, by proving the existence and uniqueness of the Nash Equilibrium, and characterizing the solution analytically, we avoided the complexity of searching the set of all possible strategies to find the NE. Using an analytical solution, we can compute the optimal strategies of both players directly and be able to assess the sensitivity of players' strategies to estimation errors on the values of parameters used in the model.

3.6.2 Stackelberg Game

In most cases, the attacker chooses his attack strategy based on the deployed security measures in the system. In this section, we analyze the interactions between the attacker and the defender as a Stackelberg game [OR94]. In this type of games, a leader chooses his strategy first. Then, the follower, informed by the leader's choice, chooses his strategy. The leader tries to anticipate the follower's response. In our case, the defender is the leader who tries to secure communication equipment in order to best protect the power system.

Stackelberg games are generally solved by backward induction. The solution is known as Stackelberg Equilibrium (SE). We start by computing the best response strategy of the follower as a function of the leader's strategy. Then, according to the follower's best response, we derive the optimal strategy of the leader.

The attacker solves the following optimization problem:

$$p(q) = \underset{p\in[0;1]^{N_c}}{\operatorname{argmax}} U_A(p,q)$$

On the other hand, the defender solves the following optimization problem:

$$q(p) = \underset{q\in[0;1]^{N_c}}{\operatorname{argmax}} U_D(p(q),q)$$

Theorem 3.3. *The game admits a unique Stackelberg equilibrium* (p^S, q^S) *given by:*

$$q^S = e(R_D^{c*} + \psi L)(R_D^c(0)C^a)^{-1}M(Q + 2R_D^c(0)C^d)^{-1} \tag{3.7}$$

$$p^S = \frac{1}{2}e(R_D^{c*} + \psi L)(R_D^c(0)C^a)^{-1}[I - M(Q + 2R_D^c(0)C^d)^{-1}M^T(R_D^c(0)C^a)^{-1}] \tag{3.8}$$

$$where \quad Q = M^T(R_D^c(0)C^a)^{-1}M$$

Proof. The solution can be found by solving the system by backward induction. We start by finding p^S by setting $\nabla U_a(p,q) = 0$. Then we solve the equation $\nabla U_d(p^S,q) = 0$ to find q^S.

Similarly to the proof of Theorem 3.2, we can prove that $(Q + 2R_D^c(0)C^d)^{-1}$ exists. □

In the next section, we validate the analytical model on a case study based on the polish electric transmission system, and analyze the behavior of the attacker and the defender at the NE.

3.7 Case Study

In this section, we validate our model on a case study based on the publicly available dataset of the polish electric transmission system at a peak load in the summer of 2004 provided in the MATPOWER computational package [ZMST11]. The dataset consists of 420 generators and 3504 transmission lines. In the electric system, we analyze the impact of tripping transmission lines or loosing generators on the power grid. The analysis of an electric system at a peak load is important, as it allows us to assess the maximum impact on the power grid as a result of a cyber attack. In fact, in such scenarios, constraints on possible containment strategies in the power grid to avoid blackouts increase. Therefore, the impact of a cyber attack is amplified by the inability of the utility to produce fast and effective control strategies to stop cascading failures from propagating in the power grid.

3.7.1 Impact Assessment

In our case study, we rely on experts' knowledge to assess the impact of attacks in the communication infrastructure and evaluate matrices F and D. However, at the end of this section, we conduct a sensitivity analysis to evaluate errors in the outputs of our model to estimation errors on the values of the elements of some of these matrices. In Chapter 4, we will present a method that allows us to generate the attack graph of a control system. Using the output of such attack graph, it becomes possible to assess the impact of attacks on the communication system and evaluate the elements of matrix S. In order to quantify the values of matrix B, we follow a similar approach to [PTC11]. We assess the impact of cascading failures in the power grid by solving power flow equations using the DC power flow approximation [Zhu09]. To achieve this objective, we simulate individual failures and assess their impact on the power grid such as identifying generators with insufficient capacities to meet the demand and overloaded lines.

In our model, we analyze the impact of tripping transmission lines or loosing generators on the power grid. In general, tripping specific transmission lines could have a significant impact on the power grid and could lead to the formation of islands in the electric system.

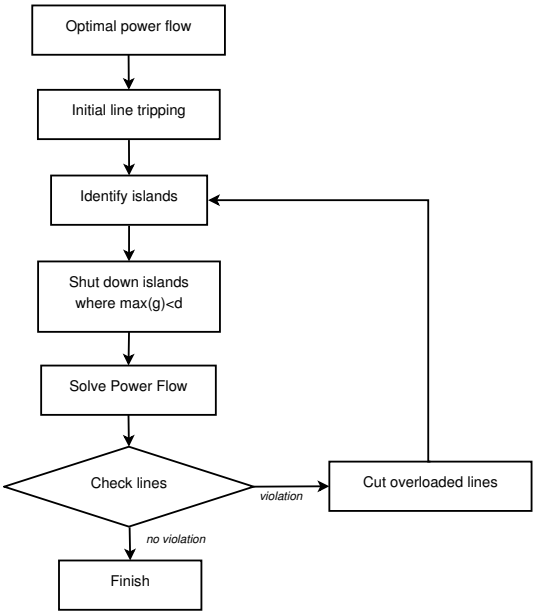


FIGURE 3.1: Flowchart of the cascade algorithm in the case of tripped transmission lines

Our objective is to identify the parts of the grid that will be impacted following a simulated blackout.

The flowchart diagram in Fig. 3.1 shows the cascading algorithm used in our model to analyze the impact of tripping transmission lines. In our algorithm, we proceed by tripping a transmission line and identifying the resulting newly formed islands. We shut down islands where the demand exceeds the maximum generation capacity in the island. We then solve the DC power flow problem in the electric transmission system using MATPOWER [ZMST11] and check the existence of overloaded lines. These lines are tripped and the process is repeated until a balanced solution emerges. Similarly, we assess the impact of loosing generators on the power grid.

In our approach, we consider the worst-case scenario where load shedding is not an option when we conduct our analysis of the impact of cascading failures on the power grid. Our objective is to identify the parts of the grid that will be impacted following a simulated blackout. Further work taking into account more fine grained analysis of the behavior of the power grid will allow us to quantify more precisely the values of the elements of matrix B. However, in this chapter, we present the cascade algorithm as a proof of the feasibility of quantifying the values of the elements of matrix B.

Even though we were able to evaluate the impact of failures in the power grid, evaluating the impact of cyber attacks on the communication infrastructure remains a challenging task. However, as we will see in Chapter 4, attack graphs are a promising solution to generate all possible attack steps to compromise a target node. These graphs could be used in

conjunction with risk assessment methods to evaluate the impact of each attacker action on the communication infrastructure.

3.7.2 System Architecture

The communication infrastructure used to control the polish electric transmission system was not available in the dataset provided in MATPOWER. Therefore, with the absence of publicly available information, we made a number of assumptions on the architecture of the communication infrastructure that we use in our case study to assess the impact of attacks on the power grid. In addition, to simplify our analysis, we combined a set of communication equipment in a single communication node depending on their functions, thus reducing the number of nodes to be represented in each electric transmission system control center. Let $\mathcal{E}$ represent the polish electric transmission system. We assume that $\mathcal{E}$ is controlled by 10 TSO (Transmission System Operator) control centers. Each center controls a set of electric nodes in a specific area of the power grid. In particular, each center has under its control a set of generators and transmission lines. In our case study, each center controls 42 generators and about 350 transmission lines. We assume that communication equipment in control centers are vulnerable to attacks. In our analysis, we assume that the attacker has enough resources and both players know the architecture of the system. As we study the impact of attacks on the power grid in the worst-case scenario, this assumption holds.

A unique TSO ICT control center is introduced to manage all communication equipment in TSO control centers. Therefore, our case study is composed of twelve building blocks: a TSO ICT control center, 10 TSO area control centers and the polish electric transmission system. The communication architecture of the electric transmission system is represented in Fig. 3.2.

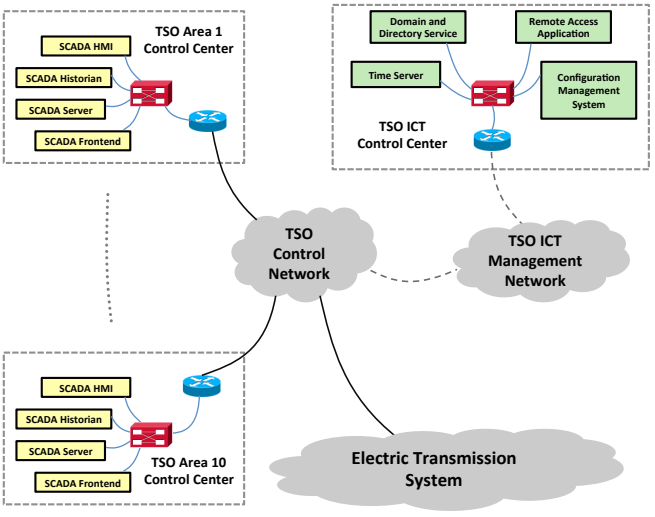


FIGURE 3.2: Example of a control network of an electric transmission system

TSO ICT Control Center. In the TSO ICT control center, four types of communication equipment are represented. A Time Server synchronizes the clocks in all communication equipment. A Domain and Directory Service manages access controls on communication equipment. The Remote Access Application is used by ICT administrators to access equipment remotely via secured connections. Finally, the Configuration Management System is responsible of pushing OS and software updates to equipment. Updates can be installed automatically or require specific authorizations on equipment performing critical operations.

TSO Area Control Centers. TSO centers control the electric transmission system. For example, commands can be sent to stop power generators or to route electricity through electric buses. In this case study, we represent four types of communication equipment in each TSO area control center: a SCADA HMI, a SCADA server, a SCADA frontend and a SCADA historian. The SCADA HMI is a human-machine interface that provides a graphics-based visualization of the controlled area of the power system. The SCADA server is responsible of processing data collected from sensors in the power grid and sending appropriate control commands back to electric nodes. In our case study, we assume that the different electric control functions performed at a TSO control center are provided by the SCADA server. The SCADA frontend is an interface between the SCADA server and electric nodes control equipment. It formats data in order to be sent through communication channels and to be interpreted when received by control equipment and vice versa. Local communication equipment responsible of controlling electric nodes are not represented in our case study, as their main task is to execute control commands received from control centers. Finally, the SCADA historian is a database that records power state events. Data can be retrieved to analyze the origins of failures or for statistical inference of the behavior of the system.

Impact Matrix. We use the algorithm in Fig. 3.1 presented in the previous section to assess the impact of stopping generators or tripping transmission lines on the electric transmission system. We rely on experts' knowledge to assess the impact of attacks in the communication infrastructure and evaluate matrices F and D. Fig. 3.3 depicts an example of possible impacts between electric and communication equipment. In the communication infrastructure, we consider that each equipment in a TSO control center is also the backup of an equipment in another TSO control center. Therefore, if a communication equipment i fails, another communication equipment j takes charge of processing the load of equipment i. In fact, backup equipment taking charge of the load of equipment of a compromised TSO area control center assumes that communications between TSO control centers can be established. However, TSO control and ICT management networks are generally managed nowadays by third parties. In the worst-case scenario, failures in the power system can lead to failures of telecommunication equipment. Therefore, a failure in the power system can affect the availability of communication links used by control centers. This should be taken into account when assessing the weight of the existence of backup equipment in the attacker and defender's utility functions.

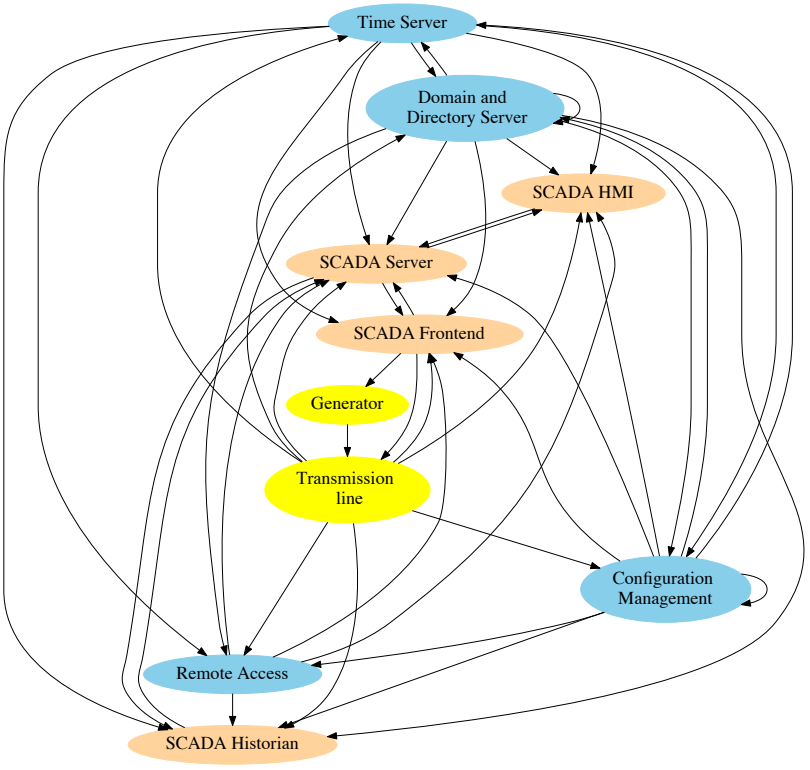


FIGURE 3.3: Example of impacts between communication and electric equipment

In this case study, we assume that the values of the initial risk on communication equipment have been computed, and we focus on the diffused risk in the system and the behavior of the attacker and the defender. In addition, for each communication equipment, we assume that the cost to defend is always greater than the cost to attack. Finally, we fix $\beta = 0.4$, $\theta = 0$, $\delta = 0.6$ and $\psi = 0.5$. Therefore, the future cascading risk has more weight than the initial risk w.r.t. the value of the total risk on communication equipment.

3.7.3 Numerical Analysis

Fig. 3.4 shows the value of risk on communication equipment in each TSO area control center after the impact of attacks propagates in the interdependent communication and electric infrastructures.

We can notice in Fig. 3.4 that the highest risk values in TSO control centers are on SCADA servers. In particular, risk values on SCADA servers in TSO 1 and TSO 2 control centers are significantly higher than risk values on SCADA servers in the other TSO control centers. In addition, in each TSO control center, the SCADA HMI has the second highest risk value.

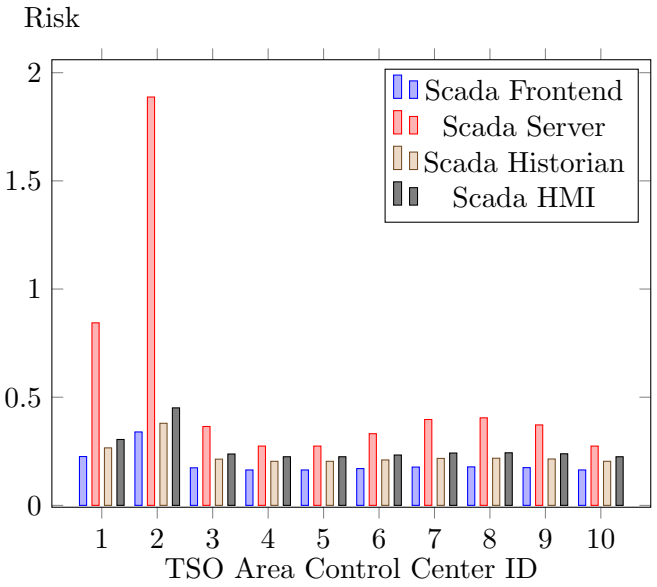


FIGURE 3.4: Risk on communication equipment in TSO area control centers

In order to understand the values of risk on communication equipment, we introduce the *impact betweenness centrality* Λ_t^c of communication node t. Λ_t^c represents the weight of node t to propagate the impact of attacks originating from the communication infrastructure on other communication equipment. Before giving the expression of Λ_t^c, we introduce $\lambda_{ij} = \sum_{r=0}^{+\infty} \delta^r (H^r)_{ij}$ as the impact metric of a communication node i on a communication node j. λ_{ij} represents the possible impact of an attack on the communication node i to affect another communication node j taking into account the interdependent electric and communication infrastructures. λ_{ij} is a measure of the cumulated impact on communication node j of an attack originating from node i. This measure takes into account all possible cascading impact paths that could exist between nodes i and j. At each step r, the weight of the payoff of the future impact is multiplied by δ, which represents in a sense the uncertainty for the attacker of getting the payoff of the r^{th} future step. Similarly to the proof of Theorem 3.1, we can prove that $\lambda_{ij} = (I - \delta H)^{-1}_{ij}$. Let $H(l)$ be the matrix identical to H while removing elements relative to the edges of node l. The importance λ_{ilj} of a communication node l in diffusing the impact of an attack on communication node i to reach communication node j can be computed as follows $\lambda_{ilj} = (I - \delta H)^{-1}_{ij} - (I - \delta H(l))^{-1}_{ij}$. Therefore, the impact betweenness centrality of a communication node t is given by $\Lambda_t^c = \sum_{r \neq t} \sum_{s \neq \{t,r\}} \frac{\lambda_{rts}}{\lambda_{rs}}$ where $\{r, s, t\} \in \mathcal{T}^c$.

In our analysis, the values of risk on a communication node i are highly correlated to $\sum_j h_{ij} \Lambda_j^c$ (correlation coefficient of 99.76% between R^{c*} and $H\Lambda^c$). In fact, the risk on communication node i depends on the identities of the nodes it will eventually impact following an attack. The more critical these nodes are in propagating the impact of attacks

TABLE 3.1: Nash and Stackelberg equilibriums of interdependent ICT and electrical infrastructures security risk management games

		$\mathbf{r_i^{c*}}$	**One-Shot game**		**Stackelberg game**	
			p^*	q^*	p^S	q^S
TSO ICT	**Time Server**	2.547	0.287	0.972	0.146	0.986
	Domain Server	2.885	0.183	0.972	0.093	0.986
	Remote App.	2.089	0.202	0.966	0.103	0.9823
	Config. Manag.	3.073	0.21	0.985	0.106	0.992
TSO 1	**SCADA Fontend**	0.226	0.275	0.537	0.15	0.591
	SCADA Server	0.844	0.295	0.688	0.156	0.744
	SCADA Historian	0.266	0.315	0.515	0.177	0.584
	SCADA HMI	0.305	0.329	0.51	0.187	0.586
TSO 2	**SCADA Fontend**	0.339	0.302	0.648	0.162	0.697
	SCADA Server	1.888	0.213	0.895	0.108	0.909
	SCADA Historian	0.379	0.344	0.618	0.189	0.684
	SCADA HMI	0.451	0.358	0.631	0.197	0.7

in the interdependent electric and communication infrastructures, the higher the risk value is on node i.

In the rest of this section, we analyze the results of the game between the attacker and the defender and evaluate the sensitivity of these results to model parameters. Table 3.1 presents the results of the one-shot and Stackelberg games between the attacker and the defender for the TSO ICT and TSO area 1 and area 2 control centers.

3.7.3.1 One-Shot Game

We study the behavior of the attacker and the defender in the one-shot game where both players choose their strategies at the same time. From Fig. 3.4 and Table 3.1, we notice that the Time, Configuration and Domain Servers have the highest risk values. These equipment are often connected to the internet, which significantly increases their attack surface. In addition, given their functions, compromising these equipment could lead to important disruptions in the communication infrastructure. As a result, at equilibrium, the defender allocates a large amount of defense resources to protect these equipment. However, this does not prevent the attacker from allocating attack resources on these equipment considering their potential impact on the power grid in the case of a successful attack.

The utilities of the attacker and the defender in the one-shot game are $U_a = 0.941$ and $U_d = -6.151$ respectively. In addition to the risk on communication equipment, the cost to attack and defend and the existence of a backup play an important role in the strategy of both players. In our case study, we noticed that in the case where the values of risk on equipment in two different TSO control centers are similar, the attacker/defender

allocate more resources to attack/defend backup equipment. Therefore, by attacking backup equipment, the attacker improves the efficiency of his attacks and increases the probability of succeeding in his attempts to disrupt the power system. On the other hand, the defender responds by allocating more defense resources to protect backup equipment. Under the assumption of the rationality of both players, the strategy at the Nash equilibrium yields the best payoff for the defender that reduces the impact of attacks on the power system taking into account attacker's actions.

3.7.3.2 Stackelberg Game

In the Stackelberg game, the defender is the leader who tries to anticipate attacker's actions and secure communication equipment to reduce the potential impact on the power system. The utilities of the attacker and the defender in the Stackelberg game are $U_a^S = 0.307$ and $U_d^S = -5.746$ respectively. Compared to the one-shot game, we notice that the defender's utility has improved. In addition, the defender increased his defense resources on each communication equipment. By choosing to allocate security investments first, the defender forced the attacker to reduce his attack resources on every communication equipment. Therefore, compared to the one-shot game, an additional security investment by the defender by 2.908 reduced the attacker's allocated resources by 6.082. By increasing his defense resources by 11.51%, the defender increased his utility by 6.58%. However, the attacker was forced to decrease his attack resources by 43.36%, which reduced his utility by 67.42%. As a result, from the point of view of the defender, the benefits of operating at the Stackelberg equilibrium outweigh the additional cost of increasing security investments on communication equipment.

3.7.3.3 Impact of Redundancies

We studied the impact of the weight of the existence of redundancies in players' utility functions ψ on attack and defense resource allocations. Fig. 3.5 shows the variation of total attack and defense resources w.r.t. ψ. We notice that ψ has a negative effect on the total amount of resources allocated by the attacker. This is consistent with the fact that increasing the weight of redundancies in player's utilities leaves the attacker with fewer choices to achieve a better payoff since the defender will increase the protection of backup equipment. In addition, we notice that when the value of ψ increases, the difference between the one-shot and Stackelberg games total defense resource allocations decreases. However, we do not notice any significant change in the difference of the total attack resource allocations between the two games. When ψ approaches 1, the total amount of defense resources in the one-shot game approaches those allocated in the Stackelberg game. In this case, the defender is better off playing a Stackelberg game, thus reducing the total amount of attack resources allocated on communication equipment.

In addition to ψ, we studied the impact of the redundancy matrix W on attack and defense strategies. Fig. 3.6 shows the variation of the attacker and the defender strategies on communication equipment in TSO area 2 control center w.r.t. variation of elements

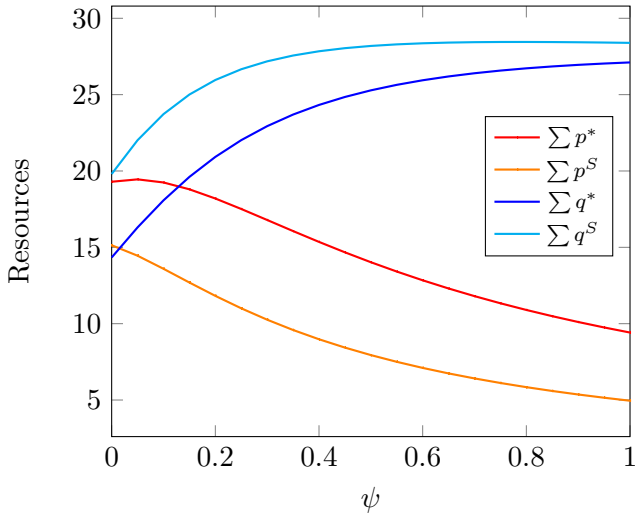


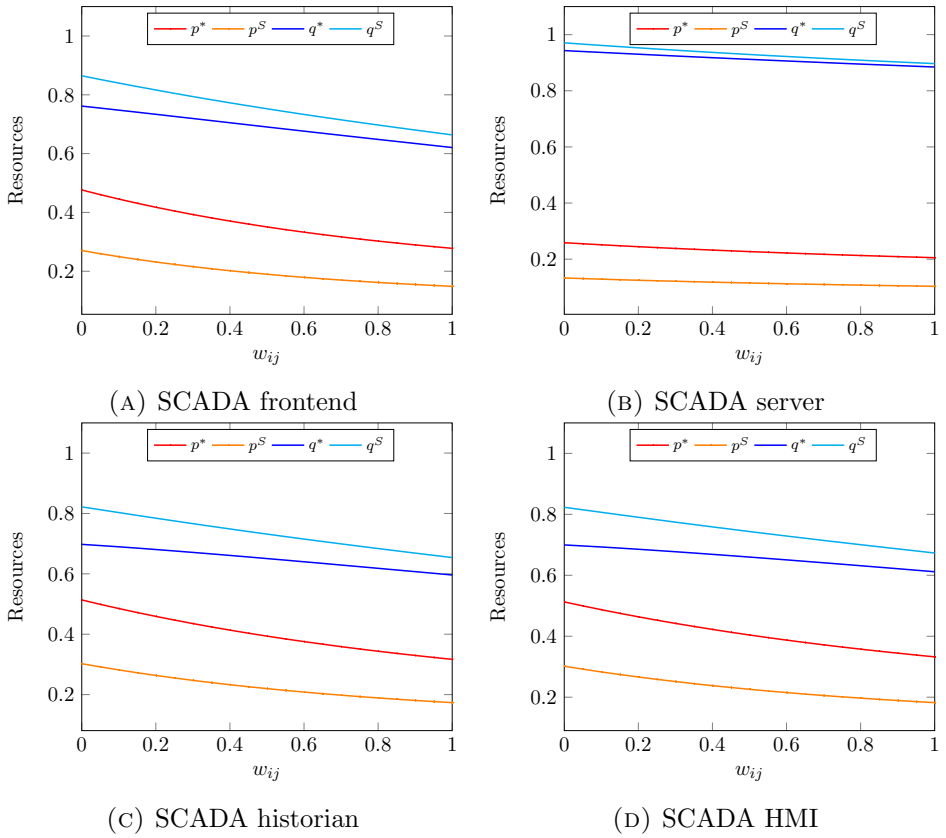
FIGURE 3.5: Variation of attack and defense resources w.r.t. ψ

of the redundancy matrix W. We analyze the behavior of the attacker and the defender when varying elements w_{ij}, the fraction of the load of node i, node j will be responsible of processing when node i is compromised. We notice that the behavior of the attacker and the defender depends on the type of the communication equipment. For example, the behavior of both players does not change significantly with respect to W for critical equipment such as the SCADA server. However, the behavior is different for the other equipment in TSO area 2 control center. Finally, increasing w_{ij} leads both the attacker and the defender to decrease their attack and defense resources on communication equipment. However, the allocated resources by both players increase on the backup of equipment of TSO area 2 control center.

3.7.3.4 Sensitivity Analysis

We rely on experts' knowledge to assess the values of a set of parameters used in our case study. However, we performed a sensitivity analysis to evaluate the consequences of estimation errors in one or a set of these parameters. We conducted a sensitivity analysis of the diffused risk R^{c*} and the players' NE strategies in the one-shot and Stackelberg games w.r.t. the values of the initial risk $R^c(0)$ and the elements of matrices S and F. We averaged the results of 10000 iterations. At each iteration, we assume that a random number of elements of $R^c(0)$ deviate from their correct values by $\pm 10\%$ (sign of the deviation is chosen randomly). We repeat the same experiment taking into account errors in a random number of elements in matrices S and F.

Sensitivity to $\mathbf{R^c(0)}$. The maximum error on the values of R^{c*} was around 4%. The attacker strategy seems more sensitive than the defender strategy with respect to errors in $R^c(0)$ at equilibrium. In the one-shot game, the maximum error on the attacker strategy was about 4.1% whereas the error on the defender strategy was about 2.1%. However, in the Stackelberg game, we noticed that the maximum error on the attacker strategy has



(A) SCADA frontend

(B) SCADA server

(C) SCADA historian

(D) SCADA HMI

FIGURE 3.6: Variation of attack and defense resources on TSO 2 w.r.t. redundancy matrix W

increased compared to the one-shot game and was about 5.1%. In the case of the defender, the maximum error has decreased and was about 1.2%.

Sensitivity to matrices S and F. The maximum error on the values of R^{c*} was around 3.4%. We do not note a significant change in the maximum errors on the attacker and defender strategies in the case of the one-shot game compared to the Stackelberg game. The maximum error on the attacker and defender strategies was about 2.1% and 1.3% respectively.

3.8 Conclusion

The smart grid is an enhanced power grid providing additional services and improving the control of the electric system, further relying on the communication infrastructure. The increased interdependency between the two infrastructures results in an additional exposure to threats. In this chapter, we presented a quantitative model, based on game-theoretic analysis, to assess the risk associated with the interdependency between the cyber and physical components in the power grid and derive the optimal strategy of the defender. In this model, we compute the total risk on an equipment as a combination of the initial risk and

the diffused/future risk. We presented a security game between the attacker and the defender and analyzed the behavior of both players in two different game settings. The objective of the attacker is to compromise communication equipment to cause the maximum impact on the power system, whereas the defender tries to protect the power system by hardening the security on communication equipment. Based on game-theoretic analysis of the behavior of the attacker and the defender, we compute the optimal strategy of the defender that reduces the impact of cyber attacks on the power grid. In our analysis, we take into account the impact of the existence of backups in the communication system. These backup equipment are enabled when a set of equipment are compromised or became unreachable. Finally, we validate our model via a case study based on the polish electric transmission system.

Part II

Optimal Security Policies Based on Attack Graphs

In Part I of the thesis, we focused on optimizing the distribution of defense resources on equipment to protect the targeted system. This is based on the assumption that the attacker is capable of compromising these equipment. In the first part, we did not discuss the possible methods that can be used that allow the attacker to achieve his objectives. In this part of the thesis, we address this issue by presenting a model for generating the attack graph of the system. The attack graph is composed of a set of attack paths and aims at identifying the sequence of actions that an attacker can execute in the system. Using the information in the attack graph, it becomes possible to identify which equipment can be compromised, assess the risk of attacks on the system, and generate a security policy better adapted to thwart potential threats.

In Chapter 4, we address the challenge of generating attack graphs in the context of industrial control systems. Given a profile of an attacker that includes his skills, preferences, and initial knowledge about the system, we generate the sequence of actions that he can execute in the system. We are interested in particular in the sequence of actions that can be executed before the next maintenance period. The output of the model is improved by taking into account the impact of attacks on the services provided by equipment. Using the information in the attack graph, we present in Chapter 5 an approach to compute the optimal security policy that guarantees that the defender's security objectives are satisfied. The solution can be used to assist asset owners to efficiently respond to intrusions, prioritize the deployment of security countermeasures, and compare the relative security of two architectures or security configurations. Finally, we validate our approach in Chapter 6 on an AMI case study.

Chapter 4

An Attack Execution Model for ICS Security Assessment

The improved communication and remote control capabilities of industrial control systems equipment have increased their attack surface. As a result, managing the security risk became a challenging task. In particular, in order to assess the impact of an attack in the industrial control system, the interdependencies between the different system components must be taken into account. In addition, the success likelihood of an attack is highly correlated to the attacker profile and his knowledge of the architecture of the system. In general, in order to express all potential actions an attacker can carry out in the system, an attack graph is generated. The attack graph represents all attack paths leading to compromise a specific target in the system. However, human analysis of network security flaws is often limited by the complexity of networks; it could therefore be error prone and impractical in general. Therefore, a tool capable of generating attack graphs automatically is needed. In addition, the consequences of attacks in an industrial control system can go beyond targeted equipment to impact services in the industrial process. In this chapter, we present the Attack Execution Model (AEM), which is an attack graph representing the evolution of the adversary's state in the system after each attack step. We are interested in assessing the risk of cyber attacks on an industrial control system before the next maintenance period. Given a specific attacker profile, we generate all potential attacker actions that could be executed in the system.

4.1 Introduction

Cyber threats are considered to be one of the main challenges to the smart grid. Attacks on the power grid could cause direct human injuries and lead to potential loss of human lives [FF05]. In the Industrial Control Systems (ICSs) of critical infrastructures, unpatched vulnerabilities continue to pose a serious threat to the security and safety of these systems.

A security vulnerability could allow attackers to launch DoS attacks, or infiltrate and potentially control networks of industrial systems [ICS12]. In 2014, most incidents reported by ICS-CERT targeted the energy sector [ICS15]. In general, the same types of operating systems found in IT networks can be found running on equipment in a small plant or even in critical infrastructures such as water treatment facilities or the power grid. However, the impact of cyber attacks on industrial control systems extends in scope, severity, and damage than their counterparts in traditional IT systems.

The recent technological advancements have benefited the industry by providing more powerful computation and communication capabilities. These new capabilities allow utilities to optimize their processes to reduce costs and increase revenues. As a result, the number of equipment used in the industrial environment that can be accessed remotely has significantly increased. The introduction of new communication mediums between the system operator and industrial equipment, and the use of off-the-shelf operating systems have increased the attack surface of these systems. In order to assess the potential impact of a cyber attack, utilities need to identify all possible actions that can be undertook by an adversary to compromise critical equipment and services in the control system.

In this chapter, our main objective is to assess the risk of cyber attacks on industrial control systems before the next maintenance period. In general, given the constrained defense budget and operational constraints (e.g. critical services need to be protected for safety reasons, difficulty to stop part of the system without impacting the industrial process), the asset owner has a decision to make about hardening security on vulnerable equipment. The operator can choose whether to wait until the next maintenance period or stop some parts or the entire system to harden security. In the latter case, he must choose the best timing to perform this task. In order to make the best decision, the asset owner has to quantify the risk unpatched vulnerabilities pose to the system. In addition, a good assessment of the probability of successfully exploiting vulnerabilities in industrial control systems should take into account attackers' profiles that include their skills, access levels on machines, and their knowledge of the topology of the control system. In some cases, depending on the profile of the attacker, certain vulnerabilities will not be exploitable. Nicholson et al. provide in [NWD+12] a detailed classification of attackers on SCADA systems. Attackers are classified depending on their motivation, capability, and operational objectives.

In order to identify the critical vulnerabilities on equipment, we need to be able to assess the impact of exploiting these vulnerabilities on the industrial process. Therefore, we need to model the different ways an attacker can proceed to compromise equipment. Attack graphs are a promising solution to this problem. In this type of graphs, attack paths represent the sequence of actions an attacker has to execute in the system in order to compromise a specific target. However, traditional representations of attack paths do not give us the necessary information to conduct a detailed analysis of the security of an ICS. Three main concepts need to be added to the traditional attack graph: the concept of time, the concept of probable attack paths, and the interdependence between the network and service layers in the system. At the end, the asset owner will be able to evaluate the probability that an

attacker with a certain profile compromises critical equipment in the system within a certain time frame (in general, the asset owner will be interested to know if a target machine is exploitable before the next maintenance period), and harden security on these equipment accordingly.

With the increased complexity of interconnections between industrial equipment, a manual assessment of the impact of compromising a particular vulnerability on the control system became a challenging task. Therefore, an automated tool able to assess the impact of attacks on control system equipment and their associated control processes is needed. To achieve this objective, the tool needs to satisfy the following requirements: i) model interdependencies that may exist between physical equipment and the services they offer to the system, ii) model interdependencies between services (to evaluate the impact of each attack step on the integrity and availability of system services), iii) model the time required to execute each attack action, and finally iv) take into account the attacker profile (knowledge of the architecture of the system, skill level, etc.) and the accumulated knowledge an attacker acquires while compromising equipment (additional credentials and knowledge of the topology of the network that can be leveraged to perform targeted attacks against equipment offering critical services).

The rest of this chapter is organized as follows. We discuss related work in Section 4.2. In Section 4.3, we present the motivations behind some key aspects of our attack graph model. Section 4.4 defines the components of the network and service layers of our model. We present the attack execution model in Section 4.5. In Section 4.6, we present the main algorithms for generating the attack graph. Finally, we conclude the chapter in Section 4.7.

4.2 Related Work

In the industrial domain, operators often do not stop the system each time a patch for a vulnerability is available, as any downtime could have an impact on the company's profits. In addition, in general, the operator needs to recertify the safety of control equipment each time a patch is applied. As a result, patching vulnerable equipment in the ICS becomes a challenging task. Therefore, it is important from the point of view of the system administrator to assess the potential impact of exploiting a set of vulnerabilities on the system. Attack graphs were proposed as a potential solution to this problem. An attack graph is composed of attack paths. Each attack path represents the consecutive actions of the attacker whose objective is to compromise one or multiple target equipment.

One of the earlier works on generating attack graphs was carried out by Dacier et al. [DD94, DDK96]. They proposed the notion of the privilege graph, which leverages the use of graph analysis techniques for network security evaluation. In this type of graphs, the nodes represent privileges and the edges represent vulnerabilities. In 2001, Swiler et al. [SPEC01] proposed another type of attack graphs. Their attack graph generation algorithm matches information about attack requirements to information about the network

configuration and assumed attacker capabilities. The graph is generated by matching the current state of the system against a library of templates, choosing only the templates that apply to the current state. Therefore, a node in the attack graph refers to the state of the system, which is the initial configuration except for the changes explicitly written in the node. However, one of the main drawbacks of this approach is its poor scalability. Adopting another strategy, Sheyner et al. [SHJ+02] later proposed automatic construction of attack graphs based on symbolic model checking. The network is modeled as a finite state machine, where the state of the network specifies the services, vulnerabilities, connectivity between hosts, and a remote login trust relation. State transitions correspond to atomic attacks launched by the intruder whose goal corresponds to violating a security property. Unfortunately, as with the work of Swiler et al. [SPEC01], this approach scales poorly. In addition, it is difficult to create inputs for the model and interpret the output.

In order to overcome the scalability limitations, a novel model to represent and reduce the complexity of modeling chains of network attacks was needed. The *requires/provides* model, proposed by Templeton and Levitt [TL00], attempted to address this issue. This model later became one of the basic components of modern attack graphs generation techniques. The *requires* part of the model lists the necessary preconditions to execute an attack. The *provides* part lists the set of postconditions or the effects that result after a successful execution of the attack. Researchers also recognized that assuming that attacker's actions are monotone would improve the generation process of attack graphs while having limited impact on its accuracy. The monotonicity assumption states that an attacker's actions will never remove a precondition of a future attack. For example, the precondition of a given exploit is never invalidated by a successful application of a previous exploit. This assumption was introduced by Ammann et al. [AWK02], which reduces the computational cost of constructing the attack graph to a polynomial complexity. In their model, an exploit is defined as an atomic transformation that, given a set of preconditions, establishes a set of postconditions. The model groups vulnerabilities, attacker access privileges and network connectivity into generic attributes. In the attack graph, the nodes refer to attributes while the edges refer to exploits.

In the modern attack graph generation techniques, we find the work of Jajodia et al. [JNO03]. Their attack graph is constructed based on a directed graph of the dependencies, via preconditions and postconditions, among exploits and conditions. The attack graph nodes represent both conditions and exploits. Therefore, edge labels become unnecessary, with directed edges simply representing generic dependency. The first versions of their tool TVA (Topological Vulnerability Analysis) scaled poorly to large networks. However, an updated version can generate attack graphs for networks of thousands of hosts, but using aggregation techniques such as protection domains [JN10]. Another tool by Ou et al. [OGA05], MulVal, uses a logic-based approach for network vulnerability analysis. The reasoning engine of MulVal consists of a collection of Datalog rules. These rules capture the system behavior and model the interactions of the different components in the network. MulVAL original version produces counterexamples for a given security policy. Ou et al. [OBM06] later proposed an approach, built upon MulVAL, to construct logical attack

graphs. This type of graphs illustrates logical dependencies among attack goals and configuration information. There are two types of nodes in the graph: a *derivation node* and a *fact node*. Each derivation node is labeled with an interaction rule. An edge in the attack graph represents a *depends on* relationship. A fact depends on one or multiple derivation nodes and becomes satisfied when the interactions rules on these nodes are applied. A derivation node i is dependent on one or multiple fact nodes, which together satisfy the preconditions of the interaction rule associated with the node i.

As we have seen so far, most attack graphs generation techniques focused on model scalability. To achieve this objective, each attack graph generation technique uses a certain abstract representation of the system. In addition to abstracting system components, optimizing the generation process of an attack graph was studied. In this category of work, Lippmann et al. propose Predictive graphs [LIS+06]. The authors focus on identifying redundant structures in the attack graph that need to be explored only once. Ingols et al. [ILP06] later improved upon Predictive graphs and proposed the notion of the Multiple-prerequisite (MP) graph. In an MP graph, the maximum number of nodes is linearly related to the source data. Their tool NetSPA that implemented this approach was later extended to model client-side attacks [ICL+09].

Another set of works combine Bayesian networks with attack graphs. The result of this combination, referred to as *Bayesian Attack Graph (BAG)*, was first proposed by Liu and Man [LM05]. Their approach attempts to model potential attack paths in a system using Bayesian networks. Frigault et al. in [FWSJ08] later proposed a Dynamic Bayesian Networks (DBNs)-based model as a solution to the problem of computing security metrics in a dynamic environment. Xie et al. [XLO+10] use Bayesian networks for real-time security risk analysis using runtime observations such as IDS alerts. Using a similar model for the Bayesian attack graph, Poolsappasit et al. [PDR12] enhance the analysis by taking into account mitigation strategies and the selection of the optimal set of security defenses. Along this line of research, Sommestad et al. present a tool CySeMoL [SEH13] that uses a probabilistic relational model (PRM) [GT11] for cyber security risk analysis. CySeMoL computes the success likelihood of different attacks based on expert knowledge and historical and empirical data. The main difference between these approaches lies in the way quantitative metrics are computed. For example, Poolsappasit et al. [PDR12] compute the local conditional probability distribution of nodes. Liu and Man [LM05] assign probabilities to the edges of the Bayesian attack graph while Frigault et al. [FWSJ08] assign probabilities to exploits.

When generating attack graphs, it is important to assess the impact of an attack on the services running in the system. These services could be the target of attacks and must therefore be protected. In addition, it is important to model the interdependencies between the services as an attack on a particular service could lead to a cascading impact on the rest of dependent services. Kheir et al. [KCBCD10] propose a privilege-based service dependency model. Even though the proposed model offers a good representation of services dependencies, it does not show how to generate attack paths that can be used to compromise services, and how each choice of attack paths impacts the attack success probability. To conduct this

type of analysis, service dependency models need to be combined with attack graphs. A first attempt was conducted by Albanese et al. [AJPS11]. However, the authors combine services dependencies with other types of dependencies that can exist in the system. The nodes in their *generalized dependency graph*, referred to as network entities, can represent hosts, subnets, applications, or services. In addition, the state of a service depends on the state of a set of network entities represented as a numerical value. In some cases where multiple vulnerabilities exist on an equipment, it is difficult, using a single node to represent an equipment in the generalized dependency model, to assess the impact of compromising these vulnerabilities if each vulnerability impacts a different set of services in the system.

In the state of the art, there are a number of works that integrate the notion of time in the attack process. Leversage and Byres [LB08] propose a state-space model in order to compute the Mean Time-to-Compromise (MTTC) for a given system. A state-space model represents all attack paths from a launch node to a target node. However, these paths are not automatically generated and need to be manually entered into the model. LeMay et al. propose a tool ADVISE [LFK+11] based on an attack execution graph to check whether a given state or an event in the system can be reached by the attacker within a certain time frame. The tool uses a very abstract view to represent potential attack steps against the system, where each attack step takes a certain time to be executed. These attack steps need to be manually defined and the security of the system is evaluated against a certain attacker profile. However, it is impractical to manually define potential attack steps in real systems. The number of attack steps can exceed the capacity of human analysis and the interest lies in general in identifying non-trivial sequence of attack steps to compromise the system. ADVISE focuses on architectural-level vulnerabilities whereas we are interested in system and machine-level vulnerabilities that need to be patched to prevent the compromise of critical equipment and services.

Finally, in his anticipation games [Bur08], E. Bursztein presents a dual-layer structure timed game in which he models the interactions between an attacker and a defender trying to compromise and defend a set of dependent services respectively. The attacker and the defender adopt a no-memory strategy where they don't have prior knowledge of the state of the system or their adversary. The notion of time is associated to rules that are executed by each player and is linked through penalties to costs. In this framework, a strategy objective is defined for each player. The tool implementing anticipation games uses heuristics to generate the optimal strategy for each player when the number of states explodes. However, in our case, we are interested in all attack executions that lead to the compromise of a machine or service within a specific time frame.

Table 4.1 exhibits a comparison of the main state of the art approaches with our model. As we have laid out in the introduction, in order for an attack graph to be useful and efficient in the assessment of the security of a control system, a number of assumptions and hypotheses must be taken into account. First, the network and service layers of the system must be modeled. The importance of exposing the different methods than can be used to compromise a service is crucial to assess the impact of attacks on the control system.

Paper	Node	Edge	Goals	Scaling	Requirements
Swiler at al. [SPEC01]	System state	Attacker action	One	Poor	Attack templates, configuration files, attacker profile
Ammann et al. [AWK02]	Set of attributes	Exploit	One	$O(N^6E)$	Reachability information, vulnerabilities, exploits, attacker privilege levels on hosts
Sheyner et al. [SHJ+02]	Network state	Atomic attack	One	Poor	Network topology, vulnerabilities, trust relations
Jajodia et al. [JNO03]	An exploit/condition	Dependency relation	One	$O(N^6)$	Reachability information, vulnerabilities, exploits
Ingols et al. [ILP06]	Attacker Access level/ Prerequisite/ Vulnerability	Dependency relation	One	$O(N)$	Network topology, vulnerabilities, credentials
MulVAL [OBM06]	Derivation node/ Fact node	Dependency relation	One	$O(N^2)$	Configuration information, attack techniques, security policy
AEM	Attacker state	Attacker action	Multiple	Poor	Reachability information, vulnerabilities, services, access control policy

Framework	Network layer	Service layer	Attacker profile	G.C.	T.D.	F.D.	Comments
Attack Graphs [AWK02, SHJ+02] [JNO03, ILP06]	Yes	No	No	Auto	No	No	-
Swiler at al. [SPEC01]	Yes	No	Yes	Auto	Yes	Yes	Defining attack templates is impractical
MulVAL [OBM06]	Yes	No	No	Auto	No	No	Constructs a logical attack graph
ADVISE [LFK+11]	No	No	Yes	Manual (AEG)	Yes	Yes	Focuses on computing a set of security metrics
Attack Scenario Graph [AJPS11]	Imprecise	Incomplete	No	Auto	Yes	No	Time used to evaluate exploit probabilities, imprecise service and network interdependencies
Anticipation games [Bur08]	No	Yes	No	N/A	Yes	Yes	Analyze behavior of an attacker and a defender, verify if a security property holds
AEM	Yes	Yes	Yes	Auto	Yes	Yes	-

G.C.: Graph construction T.D.: Temporal dimension F.D.: Financial Dimension N/A: Not applicable AEG: Attack Execution Graph

TABLE 4.1: Comparison of the state of the art approaches with AEM

However, in the state of the art, most of the related works do not clearly model the service layer and its interaction with the network layer. In addition, it is important to evaluate the dependencies between the services. In fact, in control systems, the industrial process relies on a set of interdependent services. The failure of one or multiple services can affect other services and renders the industrial process unavailable, which can have significant impact on the system.

Another issue when generating attack graphs for industrial control systems is taking into account the time the attacker needs to execute each of his actions. For example, given the limited computational resources of equipment in such environment, an aggressive scan of the system can affect their availability. Therefore, a strategic attacker will choose a scanning strategy that minimizes the probability of being detected to achieve his objectives. Finally, we should note that the strategy of the attacker and the overall potential impact on the system as a result of his attack depends on his profile and skill set.

4.3 Towards a Time-based Stochastic Attack Behavior

In general, traditional attack graphs generation techniques focus on model scalability and a certain abstraction of the components of the system is usually performed [AWK02, ILP06, JN10]. As a result, in some cases, the generated attack graph can miss important set of executions that could be in practice the most interesting. The accumulated set of knowledge items and credentials an attacker acquires while compromising equipment along an attack path can eventually determine whether his objectives can be achieved. For example, let C, D, E, F, G, and S refer to a set of equipment in the control system. Let us assume that given an initial access on equipment S, we have the following two paths: $S \rightarrow vul(C) \rightarrow vul(D) \rightarrow vul(E)$ and $S \rightarrow vul(F) \rightarrow vul(D) \rightarrow vul(G)$, where $vul(i)$ refers to a vulnerability on equipment i. In both paths, the vulnerability on equipment D was compromised. However, only the second path allowed the attacker to exploit the vulnerability on equipment G, which can later be used to compromise further equipment in the control network. In order to exploit the vulnerability on equipment G, the attacker could have needed a set of credentials that was only obtained by exploiting the vulnerabilities on equipment F then D successfully.

In a control system, the integrity and availability of the industrial processes need to be guaranteed. Let us suppose the attacker can choose between a number of attack paths as in Fig. 4.1. There are two vulnerabilities on node B. The objective of the attacker is to compromise node B and services s_1 and s_5. These two services depend on service s_3. Each vulnerability γ_i takes a certain time t to be exploited. From Fig. 4.1, if $t_4 < t_2 + t_3$, the attacker can achieve his objective with a minimum time by compromising vul 2 (compromising service s_3 impacts services s_1 and s_5). However, in practice, the choice of an attack path is more challenging. It does not only depend on the number of actions that need to be executed to compromise target equipment, but also on the attacker profile. For example, an attacker's objective could be to compromise a target machine without being

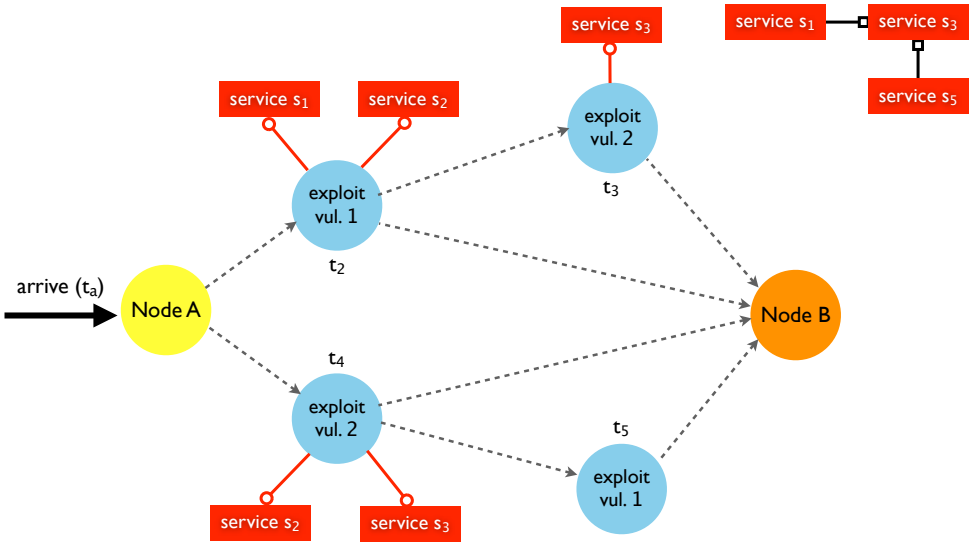


FIGURE 4.1: Example of attack paths

detected. Therefore, he will choose the set of actions that minimize the detection probability even though such behavior may require an additional set of actions and time to achieve his objective. The profile of the attacker will eventually determine which attack paths are the most probable. All this information needs to be taken into account to generate the attack graph.

In the remaining of this section, we present our motivations to model events that represent key elements to consider to derive the time required to compromise a target machine. The time it takes an attacker to compromise an equipment depends on four main atomic events: arrival time of the attacker, probing time, access time, and time to exploit vulnerabilities. In addition to the characteristic of a vulnerability, the time needed to execute an attack depends on the profile of the attacker. We are interested in evaluating the impact of different types of attackers on the ICS. Given a time period $[0, \kappa_t^M]$, where κ_t^M refers to the time of the next maintenance period, we are trying to find which equipment an attacker can compromise and how deep in the network he can successfully infiltrate.

In the attack process, the first event to consider is the arrival time of the attacker. This metric relates to the arrival rates of attackers and depends among other factors on their types, motivations, and skills. For example, a skilled attacker that is capable of developing exploits for vulnerabilities is more probable to show earlier than a script kiddie who must wait until an exploit is publicly available before launching attacks. Arrival rates can be statistically inferred from historical data that depend on the frequency of attacks targeting the system and their severities. Important insights for the most adequate distribution for the arrival rates of attackers can be found in the work of Hannes Holm [Hol14], who analyzed cyber intrusions that have been detected across more than 260000 computer systems over a period of almost three years between 2009 and 2012.

Another important issue to consider is the attacker's knowledge of the topology of the network. Critical infrastructures control system topologies and configurations are regarded as highly classified information. For example, in its framework for control systems cyber security [Dep09a], the Department of Homeland Security (DHS) introduced the *attack group knowledge* cyber security dimension. It includes any attributes of the system or actions that provide potential attackers with the means to gain information about the system. In fact, any information leak can leverage attackers' position but it is often difficult to obtain. Therefore, a good assessment model of the attack time should take into account the time required to probe the network to discover the different types of services and connections between machines. The objectives and motivations of attackers will dictate how long, how often, and how deep they try to scan the network to discover connected equipment. Eventually, the probing strategy will have a direct effect on the time it takes to compromise a given target in the network. The longer it takes an attacker to find his target, the more probable his actions will be detected by intrusion detection systems and network administrators. Therefore, the success of the attack depends, among other factors, on the activity of the attacker carried out to map connections in the system. We will refer by $t_s(\tau_i, z, K^a(t))$, the time elapsed to scan part of the network from equipment τ_i, which depends on the profile of the attacker z and his knowledge $K^a(t)$ of the architecture of the system at time t.

Once a vulnerable machine is discovered, the attacker tries to exploit it. For each type of vulnerabilities, the effort and time needed to develop and execute an exploit are different. The attacker's skills and knowledge play an important role in defining the time required to exploit a given vulnerability. It is therefore important to define the essential parameters in attackers' profiles and vulnerabilities characteristics that are critical to develop and execute exploits. We refer by $t_e(\gamma)$, the time required to exploit vulnerability γ.

Finally, when there are multiple vulnerable services on the same machine, important questions arise: Which one the attacker will decide to compromise first? What are the required conditions to execute the exploit? etc. In practice, we can focus on leveraging our knowledge of the preferences of the attacker (given his profile) in order to build a stochastic behavior of the evolution of the state of the attacker in the system. This can have an important impact on reducing the complexity of building the model. In the next sections, we will give a set of definitions that will allow us to simplify the presentation of our model. However, we do not intend to present a new language to express atomic attacks, which is beyond the scope of this chapter. For generic attack description languages, the reader can refer to LAMBDA [CM02] and ADeLe [MM01].

4.4 Control System Architecture

In our model, we represent the control system in two layers: the network layer and the service layer. The network layer consists of the physical equipment and their interconnections. The service layer consists of services used to execute industrial processes. In this section, we will

give formal definitions of the principal components of each of these layers. Table 4.2 lists the main symbols used throughout this chapter.

TABLE 4.2: List of main symbols in Chapter 4

$\mathcal{T}$	set of vertices in $\mathcal{H}$ representing system equipment
$\mathcal{D}$	service dependency graph
Δ	set of services in the system
Γ	set of vulnerabilities in the system
$\mathcal{K}$	set of knowledge items
$\mathcal{K}(z)$	set of attacker z knowledge items
K^a	information about nodes and their interconnections
K^c	set of credentials on machines
K^t	set of specific tools to exploit vulnerabilities
Π	set of attackers' types
S	skill level of the attacker
A	set of attacker's possible actions
R	set of attacker's preferences
Ξ	set of rules
$\mathcal{L}$	set of all access levels in the system
$\mathcal{W}_{\tau_i}$	impact of an attack on node τ_i
$\mathcal{W}_{\delta_i}$	impact of compromising service δ_i
$\mu_i^{\{h,n\}}$	detection rate of the i^{th} IDS
$\alpha_i^{\{h,n\}}$	false positives rate of the i^{th} IDS
$\beta_i^{\{h,n\}}$	false negatives rate of the i^{th} IDS
γ_i	vulnerability i
t_a	arrival time of the attacker
$t_e(\gamma)$	time required to exploit vulnerability γ
t_s	time required to perform a scan
κ_t^M	time until the next maintenance period
κ_b	maximum attack budget
κ_a	maximum number of attacker actions in an attack path
$\mathcal{C}_v$	set of critical nodes
$\mathcal{C}_s$	set of critical services
P	set of attack executions between $[0, \kappa_t^M]$
p_i	i^{th} attack execution in P

4.4.1 Network Layer

The network layer represents configuration and topological information about the system. Each node in this layer refers to a machine in the network. The state of the attacker on each machine depends on his actions. For example, after scanning the network, the existence of an equipment can be discovered by the adversary and the vulnerabilities on that equipment can be exploited. We will represent the network as a directed graph $\mathcal{H} = \langle \mathcal{T}, \mathcal{Y}, l_{\mathcal{Y}} \rangle$. $\mathcal{T}$ represents the set of vertices and $\mathcal{Y}$ is a subset of $\mathcal{T}^2$ and is referred to as the edges of $\mathcal{H}$.

Each vertex of the graph represents a physical equipment. The communication from a vertex τ_i to a vertex τ_j is represented by the edge y_{ij}. $l_{\mathcal{Y}} : \mathcal{Y} \rightarrow \{n, m\}$ is an edge labeling function where n refers to a network-based communication and m refers to a manual human-based communication. A network-based communication can be established between vertices τ_i and τ_j if they can communicate through the network. However, in some cases, a human operator manually intervenes to transmit information and configuration files between two machines in the system. This scenario occurs in general when a machine needs files from another machine and there are no network-based communications between the two equipment. All vertices $\tau_i \in \mathcal{T}$ will be referred to as nodes in the remaining of this chapter. More formally:

Definition 4.1 (Node)**.** *A* node τ_i *in the graph* $\mathcal{H}$ *represents a machine in the network and is represented by the tuple* $\langle \Gamma_i, \Delta_i \rangle$ *where:*

- $\Gamma_i = \{\gamma_k\}$ *represents the set of vulnerabilities* γ_k *on node* τ_i
- $\Delta_i = \{\delta_k\}$ *refers to the set of services running on node* τ_i*.*

For each node τ_i, we associate a tuple $\langle (l, k^c)_r \rangle$ where $(l, k^c)_r$ refers to the set of credentials k^c required to access τ_i with the access level l. For example, we may have two access levels *user* and *root* on a node, where each one requires a different set of credentials to get access to that node with the corresponding access level. In our model, a vulnerability represents a weakness in the hardware or software on a node. Multiple services can run on a machine (depending on its role, computation capacity, etc.). In addition, multiple machines can be used to run a particular service. In this case, in order to compromise the service, the attacker needs to compromise it on one of the supporting machines.

4.4.2 Service Layer

The service layer represents dependencies between the services in the system. Each node in the system is responsible for providing a service or a set of services. Multiple nodes can interact to provide a particular service. Fig. 4.2 depicts an example of equipment and services dependencies. For example, equipment 1 provides service a, while equipment 2 and 3 must interact to provide service b that depends also on service a. Therefore, service b cannot be provided if service a was compromised or one of the equipment 2 or 3 were compromised.

The dependency graph $\mathcal{D}$ between the services is represented by a tuple $\mathcal{D} = \langle \Delta, \rightarrow \rangle$ where Δ refers to the set of services in the system, and $\rightarrow$ is a binary relation representing a dependency between two services. In general, we can have disjoint dependent set of services and cyclic dependencies between services could exist.

After each action executed by the adversary, the state of the services in this layer is synchronized with the state of the nodes in the network layer. Therefore, at each attack

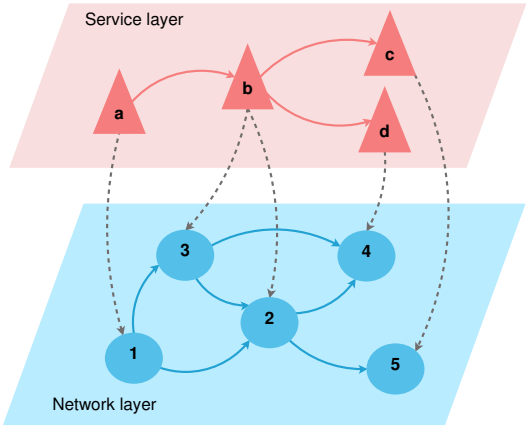


FIGURE 4.2: Example of equipment and services dependencies

step, the defender can identify the set of services that were compromised and as a result, check whether his security objectives are still satisfied. We define a service in the system as follows:

Definition 4.2 (Service)**.** *A* service δ_i *is an atomic function required to execute a specific industrial process.*

For each service δ_i, we associate a tuple $\langle (l, k^c)_j \rangle$ where $(l, k^c)_j$ refers to the access level l and the set of credentials k^c required to use/access service δ_i. In fact, different persons, each with a different role in the network, can access a service. For each role, the access level and the set of required credentials to access the service can be different (i.e. depending on the permissions granted to each role). In addition, if the service requires interactions between multiple machines, using/accessing the service on each of the supporting machines may require different access levels and credentials.

4.4.3 Security Mechanisms

Generating an attack graph aims at assessing the security of a system, which should take into account the deployed security mechanisms. These mechanisms include measures to enforce the access control policy, traffic filtering through network firewalls, deploying host-based and network-based IDSs for attack detection, etc. In this chapter, we suppose that the access control policy and firewall rules were analyzed in a preprocessing phase. The output of this analysis is the directed graph $\mathcal{H}$ representing authorized interactions between network equipment.

In our attack graph, we are interested in the success likelihood of attacks. Therefore, we assume that a host-based IDS could be installed on certain equipment to detect malicious activities. This host-based IDS is characterized by its detection rate μ_i^h, its false positives rate α_i^h, and false negatives rate β_i^h. In addition, we may have a network-based IDS deployed in the system. In this case, we associate for each edge y_{ij} monitored by the IDS the

parameters μ_{ij}^n, α_{ij}^n, and β_{ij}^n representing the detection rate, the false positives rate, and the false negatives rate of the network-based IDS respectively.

4.5 Attack Execution Model

As we have seen so far, we assume that certain information about the system is available (e.g. services running in the system, vulnerabilities, access control policy). In this chapter, we restrict our analysis on constructing an attack graph without discussing how that information can be collected. Even though several methods exist to get most of the data that is needed, we acknowledge that some information is harder to get than others. For example, the time required to compromise vulnerabilities is harder to assess than mapping the network connections between equipment. Therefore, in this chapter, we make the following assumption:

Assumption 4.1. *The required information about the system needed to build the attack graph is available.*

4.5.1 Attacker State

The profile of the attacker plays an important role in assessing the success likelihood of attacks and their potential impact on the system. We formally define an attacker as follows:

Definition 4.3 (Attacker)**.** *An* attacker *is any type of adversaries targeting the system and is represented by the tuple* $\langle \pi, S, A, R \rangle$ *where:*

- π *refers to the type of the attacker*
- S *refers to the skill level of the attacker*
- A *refers to the possible actions that can be executed by the attacker where* $A \subset \{$*scan, access, exploit*$\}$
- R *refers to the attack preferences.*

π is one among the set $\Pi = \{$*script kiddie, hacker, hacktivist, nation state, insider attacker*$\}$. The adversary's attack preferences R is an ordered set of priorities (depth of a scan, cost of an attack, its payoff, etc.) taken into account when executing an attack. This information is used, in conjunction with other parameters, to assist in the attacker decision-making process regarding the choice of the next attack step when multiple possibilities exist.

In order to execute each attack action, the attacker should have acquired a set of knowledge items. For example, the attacker must know the existence of a vulnerability on a machine and acquired or developed the necessary tools in order to exploit that vulnerability. In general, we define the set of knowledge items as follows:

Definition 4.4 (Knowledge items)**.** *The set of* knowledge items $\mathcal{K}$ *is represented by the tuple* $\langle K^a, K^c, K^t \rangle$ *where:*

- $K^a = \langle \mathcal{T}, \mathcal{Y} \rangle$ *refers to information about machines (running services, etc.) and their interconnections*
- $K^c = \{(\tau_i, k_i^c) | \tau_i \in \mathcal{T}\}$ *refers to the set of credentials on machines*
- K^t *refers to the set of tools needed to exploit specific vulnerabilities in the network.*

K^c refers to the set of credentials on machines. For example, the knowledge of a password on a given machine can provide the attacker with the required credentials to gain access to that machine, which he can leverage to compromise additional equipment and services in the system.

We define an attacker state as follows:

Definition 4.5 (Attacker state)**.** *At a given instant, an* attacker state *refers to the set of access levels acquired on equipment and the set of knowledge items at the disposal of the attacker.*

The state of the attacker in the system evolves depending on his actions. This evolution is described using a set of rules Ξ.

4.5.2 Rule-based Attack Execution

We refer by Ξ the set of rules used to describe the evolution of the state of the attacker in the system. There are four types of rules: scan, network access, human-based access, and exploit. Each rule needs a set of preconditions *Pre* to be executed and its execution results in a set of postconditions *Post*. In the rest of this section, we formally define the rules governing the evolution of the state of the attacker in the system.

4.5.2.1 Scan

In order to compromise a target equipment, an attacker needs to identify the possible paths that could lead to his target. In critical systems, aggressive scanning of the network can set off alarms and affect services on equipment, and therefore increases the probability of detecting intrusions. The attacker can choose to compromise equipment located on specific attack paths, leveraging his knowledge of critical assets' locations while decreasing the probability of being detected. We define the scan rule ξ_{scan} as follows:

Definition 4.6 (Scan rule)**.** *The* scan rule ξ_{scan} *is defined as* Pre $\langle \mathcal{T}, R, \mathcal{K} \rangle \xrightarrow{t_s, c_s}$ Post $\langle \Gamma, K^a \rangle$*, where* t_s *refers to the time the attacker spends to perform the scan and* c_s *to its associated cost.*

Taking into account the attacker's preferences (how often and how deep to scan, the probability of being detected, etc.) and his set of acquired knowledge items, scanning the network from a node $\tau_i \in \mathcal{T}$ results in the discovery of new equipment and their vulnerabilities. For this rule, we associate a time t_s representing the time the attacker spends to perform the scan and a cost c_s for executing this rule (e.g. effort to perform the scan, the cost to the attacker if he was detected).

4.5.2.2 Network Access

Depending on the role of employees in an industrial environment, different access levels are granted. These access levels can be used as access tokens on equipment. Let $\mathcal{L}$ be the set of all possible access levels in the system. We refer by l_i, the access level of type i. With respect to the access levels, we make the following assumption:

Assumption 4.2. *A partial ordering $\leq$ between access levels exists. However, the order is total between access levels on a particular node.*

We define an access from an equipment to another connected equipment in the network as follows:

Definition 4.7 (Network access rule)**.** *The* network access rule $\xi_{\text{network\_access}}$ *is defined as* Pre $\langle \mathcal{T}, \mathcal{L}, \mathcal{K}, \mathcal{T} \rangle \xrightarrow{c_n, \eta_n}$ Post $\langle \mathcal{L}, K^a, K^c \rangle$*, where c_n and η_n refer to the cost and the payoff associated to the execution of this rule respectively.*

The attacker tries to access a remote equipment in the system using his access level $l_i \in \mathcal{L}$ on a compromised machine and a set of knowledge items. If this type of access is allowed in the access control policy, the result of the execution of this rule is the set of access levels granted to the attacker on the remote equipment and an access to system configuration files and credentials located at that equipment using the granted set of access levels. A cost c_n and a payoff η_n are associated to the execution of this rule.

4.5.2.3 Human-based Access

In an industrial environment, for various reasons, different equipment could not be connected together. However, for operational requirements, specific information needs to be transmitted between these equipment. For example, an equipment could require configuration files that exist on another equipment. In general, the operator transmits these files manually using USB flash drives or other storage mediums. We formally define this type of access rules as follows:

Definition 4.8 (Human-based access rule)**.** *The* human-based access rule $\xi_{\text{human-based\_access}}$ *is defined as* Pre $\langle \mathcal{T}, \mathcal{L}, \mathcal{K}, \mathcal{T} \rangle \xrightarrow{t_h, c_h, \eta_h}$ Post $\langle \mathcal{L}, K^a, K^c \rangle$*, where t_h, c_h, and η_h refer to the time, the cost, and the payoff associated to the execution of this rule respectively.*

The key difference of executing this rule compared to the network access rule (Definition 4.7) is taking into account the time t_h corresponding to the average time elapsed between two consecutive manual human intervention to transmit information from the compromised to the remote equipment. In order to take into account the worst-case scenario, we can assume that a malware installed on the remote equipment continues to explore the network after the execution of this rule. This assumption depends on the profile of the attacker and his capability to develop an intelligent malware able to execute actions that reflect his preferences and achieve his objectives.

4.5.2.4 Exploit

We use the pre/post-conditions model to represent the prerequisites and the consequences of exploiting a vulnerability. In particular, a vulnerability γ is represented by the tuple $\langle \varphi_{pre}, \varphi_{post} \rangle$. $\varphi_{pre} = \{b, S, l, k^c, k^t\}$ refers to the set of preconditions required to exploit the vulnerability. $b = \{0, 1\}$ is a binary value referring whether the vulnerability can be exploited locally (0) or remotely (1). S refers to the minimum attacker skill level required to exploit γ. l refers to the required access level to exploit γ. $k^c \subset K^c$ and $k^t \subset K^t$ refer to the set of credentials and tools the attacker needs to possess in order to exploit vulnerability γ successfully. $\varphi_{post} = \{l', k^{a\prime}, k^{c\prime}\}$ refers to the set of postconditions representing the consequences of successfully exploiting the vulnerability. l' refers to the acquired access level after exploiting γ. $k^{a\prime}$ and $k^{c\prime}$ refer to the additional knowledge of the topology and the configuration of the network and the set of credentials acquired after exploiting the vulnerability respectively. For example, exploiting a vulnerability could increase the privilege level of the attacker on the vulnerable machine and allow him to access sensitive data such as configuration files and credentials needed to access critical equipment in the system.

The rule to exploit a vulnerability $\gamma = \langle \varphi_{pre}, \varphi_{post} \rangle$ is defined as follows:

Definition 4.9 (Exploit rule)**.** *The* exploit rule ξ_{exploit} *is defined as* Pre $\varphi_{pre} \xrightarrow{t_e, c_e, \eta_e}$ Post φ_{post}*, where* t_e*,* c_e*, and* η_e *refer to the time, the cost, and the payoff associated to the execution of this rule respectively.*

We associate to the exploit rule of a vulnerability the time t_e needed to develop and execute the exploit and its development and execution cost c_e. The payoff η_e represents the gain the attacker gets if the exploit was successfully executed.

4.5.3 Formal Model

In general, in an ICS, equipment have limited resources. In addition, processes executed within the system are mission critical and a failure could result in a severe impact on the infrastructure's equipment, operations, and personnel. Therefore, any action executed by an attacker in the system should take into account these constraints. In order to represent the

evolution of the attacker in the system, we adopt the notion of atomic attack executions. An atomic attack execution is a single action of the attacker that given a set of preconditions, the state of the attacker in the system changes. More formally:

Definition 4.10 (Atomic attack execution)**.** *An* atomic attack execution *is a couple* (ξ_i, τ_i) *where* ξ_i *is a rule executed on node* $\tau_i \in \mathcal{T}$*.*

Before an attack begins, an attacker z possesses a set of initial knowledge items $K^0(z)$ and could have an initial access to certain equipment. $K^0(z)$ contains information about the topology and the configuration of the network, credentials at the attacker's disposal, and a set of offensive tools.

In our analysis, we are interested in assessing the risk of cyber attacks on an industrial control system before the next maintenance period. An attack execution is a sequence of atomic attack executions corresponding to rules executed by the attacker. Let P be the set of all attack executions within $[0, \kappa_t^M]$, where κ_t^M refers to the time of the next maintenance period. An attack execution is defined as follows:

Definition 4.11 (Attack execution)**.** *An* attack execution $p_i \in P$ *is a tuple* $\langle(\Xi_i, \tau_i, q_i), >\rangle$ *where:*

- Ξ_i *refers to a rule executed on node* τ_i
- q_i *refers to the probability of executing* Ξ_i *successfully*
- $>$ *is a strict order on atomic attack executions.*

The rules Ξ have localities that determine on which nodes they are executed. We associate a probability q_i to the successful execution of rule Ξ_i. Finally, we define a strict order $>$ on atomic attack executions. Therefore, we note $z_1 > z_2$ if the atomic attack execution z_1 is executed before z_2.

Let $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ refer to the set of supervertices where each supervertex v_i represents a state of the system. The system state is defined as the state of connections and interaction rules between the different components of the control network (i.e. the state of connections and the rules that govern the communications between equipment). For example, changing the access control policy modifies the state of the system by denying or granting access on existing equipment. Moreover, in critical control systems, compromising a set of services can trigger safety features that modify the existing connections and change the rules that govern the interactions between equipment. In addition to defining these states, we should note that we rely on the system operator to define the specific parameters or actions that trigger any change in the state of the system.

In each supervertex v_i, let X_i represent the set of subvertices $\{x_1^i, x_2^i, ..., x_{N_i}^i\}$, where each x_j^i represents an attacker state when the system state is v_i. Attacker states evolve

depending on the actions of the attacker. This evolution is described using the set of rules Ξ. At any given time t, the attacker state represents the adversary's access levels on equipment, acquired credentials, and knowledge about the topology and configuration of the control system. $\mathcal{E}$ is a subset of $\{\bigcup_i X_i\}^2$. An element of the set of ordered pairs of subvertices $\mathcal{E}$ is defined as $e_{ij} = (i, j)$. Let $\Sigma_\mathcal{V}$ and $\Sigma_\mathcal{E}$ be two finite alphabets of supervertex and edge labels respectively. $\Sigma_\mathcal{V}$ represents a description of the set of states the system can transition to following an action by the adversary. $\Sigma_\mathcal{E} = \{$*scan, network access, human-based access, exploit*$\}$. $l_\mathcal{V} : \mathcal{V} \to \Sigma_\mathcal{V}$ and $l_\mathcal{E} : \mathcal{E} \to \Sigma_\mathcal{E}$ are two mapping functions for supervertex and edge labeling respectively.

Given all this information, we define our attack execution model, which is an attack graph, as follows:

Definition 4.12 (Attack execution model)**.** *An* attack execution model (AEM) *is a labeled supergraph represented by the tuple* $\langle \mathcal{V}, \mathcal{E}, \Sigma_\mathcal{V}, \Sigma_\mathcal{E}, l_\mathcal{V}, l_\mathcal{E} \rangle$.

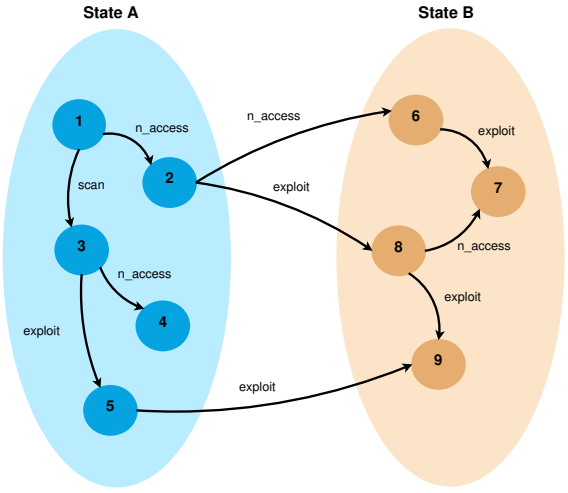


FIGURE 4.3: Example of a sequence of attack steps

Fig. 4.3 shows an example of the execution of a sequence of attack steps. In this example, the system can be in two different states: state A and state B. Subvertices 1 to 9 represent the state of the attacker in these two system states. In each state of the system, the state of the attacker changes as a result of the execution of an action by the attacker. For example, the state of the attacker can change after accessing a node in the network (transition $1 \to 2$) without changing the state of the system. Exploiting a vulnerability on a node can change the state of the attacker (transition $5 \to 9$) and can be accompanied by a change in the state of the system (state A $\to$ state B). In some cases, getting access to a node can be used to change the state of the system (transition $2 \to 6$). For example, accessing a particular node can provide the attacker with sufficient privilege to change the state of the system. We note that in practice, as in most attack graphs generation techniques, we can simplify our attack paths generation process by assuming that the system state does not change. For example, attacker's actions do not change the access control policy and the state of the connections

between equipment in the network. However, in this case, it is important to evaluate the impact of this assumption on the accuracy of the generated attack graph.

4.6 Implementation

In this section, we present the main algorithms for the generation of our attack graph. For our implementation, we make a number of assumptions. Some of the assumptions that we make aim at reducing the complexity of constructing the AEM, while others enhance its expressivity and accuracy of modeling real-world scenarios.

Assumption 4.3. *If an equipment τ_i was partially compromised and the attack proceeded on other equipment, the attacker will not return to compromise τ_i again.*

In order to reduce the complexity of constructing the attack graph, we assume that the attacker will not return to an equipment that he has already compromised even partially. In general, Assumption 4.3 holds. However, in some cases, we may miss to construct a number of attack paths. This is especially true when we have multiple access levels on an equipment τ_i. For example, let us assume that the attacker managed, with his current set of knowledge items, to get only the lowest access level (in terms of ordering) on τ_i. If the attacker proceeded and managed to compromise other equipment, he may get the additional knowledge required to get back and access τ_i using the remaining access levels.

4.6.1 Algorithms

In this section, to simplify the presentation, we assume that the state of the system does not change. Algorithm 2 presents the pseudocode of the main function CONSTRUCTAEM for generating the attack graph $\mathcal{G}$. The input of this function is an attacker profile z and a node t to which the attacker has initial access.

Before explaining the algorithm, we will start by presenting some notations. Let Γ_t^l and Γ_t^r refer to the set of vulnerabilities on node t that can be exploited locally and remotely respectively. $\mathcal{K}(z)$ refers to the set of knowledge items at the disposal of the attacker. Let $\mathcal{C}_v$ and $\mathcal{C}_s$ refer to the set of critical nodes and services in the system respectively. At each stage of an attack path, let κ_t, κ_b, and κ_a refer to the elapsed time, the consumed budget of the attacker, and the number of attack actions that were executed so far. κ_t^M, κ_b^M, and κ_a^M refer to the time until the next maintenance period, the maximum budget at the disposal of the attacker, and the maximum number of actions allowed for the attacker along each attack path. We assume that these values are provided as an input to the model. κ_a^M will help reduce the complexity of constructing the model by providing an upper limit to the allowed depth of the attack graph.

Algorithm 2

Input: $t \in \mathcal{T}$, z
Result: $\mathcal{G}$

```
1  function constructAEM(t, z)
2      t.passOver ← TRUE
3      if t.compromisedServices ∉ C_s & κ_a < κ_a^M then
4          exploitLocalVulnerabilities(t, z)
5          if t ∉ C_v then
6              k ← Scan(t)                          ▷ k: set of knowledge items
7              if k ≠ ∅ & ¬stateExists then
8                  K(z) ← K(z) ∪ k
9                  x ← createNewAttackState
10                 G ← G ∪ {x}
11                 Update(κ_t, κ_b, κ_a)
12             end if
13             if ((k == ∅||v ≠ ∅) & κ_t ≤ κ_t^M & κ_b ≤ κ_b^M & κ_a < κ_a^M) then
14                 for each m ∈ z.getNetworkAccessibleNodes(t) do
15                     if m.passOver == FALSE then
16                         accessRemoteNode(t, m, z, NETWORK)
17                         exploitRemoteVulnerabilities(t, m, z)
18                     end if
19                 end for
20                 for each m ∈ z.getHumanAccessibleNodes(t) do
21                     if m.passOver == FALSE then
22                         accessRemoteNode(t, m, z, HUMAN)
23                     end if
24                 end for
25             end if
26             if k ≠ ∅ & ¬stateExists then
27                 Restore(κ_t, κ_b, κ_a)
28                 K(z) ← K(z)\k
29             end if
30         end if
31     end if
32     t.passOver ← FALSE
33 end function
```

The attack graph is generated using a depth-first strategy. At first, the profile of the attacker includes, among other factors, his initial knowledge of the architecture of the system and access levels on equipment. We proceed in a modular way to model the evolution of the state of the attacker. After each attack step, the knowledge of the attacker is updated with new information about the architecture of the system or his privilege level increases on an equipment and new information such as credentials is acquired after exploiting a vulnerability. We note that we model information on equipment that can be accessed with a specific access level and a set of required credentials. This data can include information about the topology of the network or a database of credentials that can be later used by the attacker to compromise additional equipment. In our model, we distinguish two types of vulnerabilities: a local vulnerability that requires a local access on the equipment to be exploited and a remotely exploitable vulnerability.

In general, we are not certain that a specific knowledge item such as a password will be useful for the attacker in future attack steps, which will have an impact on the number of states generated in the attack graph. In addition, in order to find the most probable attack path with minimum time to achieve the attacker's objective, all possible combinations of attacker actions are explored. However, this increases the complexity of the attack graph generation algorithm, which quickly limits the number of control system equipment that we can model. This limitation can be overcome using some pruning techniques applied during the process of generating the attack graph. For example, we can limit the number of actions an attacker can execute at any given time depending on his partial knowledge of the architecture of the system by prioritizing among his possible immediate actions. The prioritization decision-making process depends on the profile of the attacker and his objective.

In Algorithm 2, we continue to explore the network in search of new attack opportunities unless one of the following stopping conditions is satisfied: attack time κ_t reached the time until the next maintenance period κ_t^M, the attack budget κ_b that was spent reached κ_b^M, the number of actions κ_a executed in an attack path reached κ_a^M, and finally one of the critical nodes in $\mathcal{C}_v$ or critical services in $\mathcal{C}_s$ has been compromised. We note that if an attacker gained access to a critical node, we nevertheless explore whether any local vulnerability on that node can be exploited before stopping. If the operator did not specify any critical nodes or services, the algorithm will explore all possible actions that can be executed by the attacker in the system.

At the beginning of Algorithm 2, when an attacker z gets access to node t, we start by checking whether any critical service is compromised. This takes into account the interdependencies that exist between the services represented by the graph $\mathcal{D}$. If the attacker failed to compromise any critical service and did not reach the maximum number of actions in an attack path (line 3), he will try to exploit the local vulnerabilities EXPLOITLOCALVULNERABILITIES (line 4) and later stop if $t \in \mathcal{C}_v$. The function EXPLOITLOCALVULNERABILITIES is defined in Algorithm 3.

Algorithm 3

Input: $t \in \mathcal{T}$, z

```
1  function exploitLocalVulnerabilities(t, z)
2      for each γ ∈ Γ_t^l do
3          if z.hasDiscovered(γ)& ¬z.hasExploited(γ) & z.wantsToExploit(γ) then
4              if z.canExploit(γ) then
5                  k ← Exploit(γ)
6                  if k ≠ ∅ then
7                      z.hasExploited(γ) ← TRUE
8                      Update(κ_t, κ_b, κ_a)
9                      if (κ_t ≤ κ_t^M & κ_b ≤ κ_b^M & κ_a < κ_a^M & ¬stateExists) then
10                         K(z) ← K(z) ∪ k
11                         x ← createNewAttackState
12                         G ← G ∪ {x}
13                         constructAEM(t, z)
14                         K(z) ← K(z)\k
15                     end if
16                     Restore(κ_t, κ_b, κ_a)
17                     z.hasExploited(γ) ← FALSE
18                 end if
19             end if
20         end if
21     end for
22 end function
```

If $t \notin \mathcal{C}_v$, we call the function Scan that checks if the attacker z needs to scan the network and returns the additional knowledge k that will be acquired if a scan was needed. If there is at least one additional knowledge item that was acquired and the attacker state after scanning the network does not already exist in the attack graph $\mathcal{G}$, then: (i) we add k to the attacker knowledge $\mathcal{K}(z)$, (ii) we create a new attack state x and add it to $\mathcal{G}$, and (iii) we finally update the values of κ_t, κ_b, and κ_a.

Before adding a new attack state x_i to $\mathcal{G}$, we need to check whether there already exists a state $x_j \in \mathcal{G}$ such that x_i and x_j are equivalent. There are multiple approaches to define the equivalence between two states. The classic approach requires checking if each state contains the same set of knowledge items and acquired access levels. Another approach, which we adopt in our implementation, is to check if the set of elements in x_i and x_j will give the attacker the same leverage in his future attacks. In particular, if the knowledge items and access levels in states x_i and x_j will lead to the same set of attack states. In this case, instead of focusing on what the attacker has already achieved, we focus on what he is capable of achieving with his set of knowledge items. At the end, the choice of the method depends, among other factors, to a large extent on the topology of the network. With respect to performance, in the second approach, we may still have some redundancies when we construct the attack graph. For example, a number of elements k of the set of knowledge items (e.g. acquiring a certain password) in a state x_i will not eventually be

used by the attacker. x_i is created as a new attack state in $\mathcal{G}$ even though another state $x_j \in \mathcal{G}$ may exist with the same knowledge items as x_i with the exception of k. To solve this problem, we run a cleaning algorithm to remove redundancies after the entire attack graph is generated. However, in some attack graph topologies, the performance of this algorithm suffers. We note that for optimizing network defense, requiring the removal of redundancies in the attack graph before analyzing it may not be essential, as we will see in the next chapter.

It is worth mentioning that, even though the AEM represents the evolution of the state of the attacker in the system, we do not always need to save all the elements of the attacker knowledge set in each state. With minor modifications to the function CREATENEWATTACKSTATE, it is possible to save the required information in each attack state depending on the objective of the analysis of the attack graph. If, for example, we are interested in the evolution of a specific set of attacker knowledge items, we only consider that set when we want to create a new attack state. Otherwise, if we are interested, as we will be in the next chapter, in managing the security risk by choosing the optimal set of defense countermeasures that offer the best protection to the system, we only need to have the following information in an attack state: the type, the impact, and the source and target nodes of the attacker action (and the ID of the vulnerability if one was exploited). We note that including both the source and target nodes as identifiers of an attack state depends on the objective of the analysis and the available set of countermeasures that we can deploy. For example, if we want to deploy a network based IDS and we want to identify the optimal set of links to monitor, we should identify the source and target nodes of an attack. Finally, from the information stored in previous and current attack states, we can deduce the set of services that were compromised following an attack action. This information is useful when we want to set reachability constraints. For example, we want to choose defense countermeasures such as the probability of compromising a particular service or being granted a specific access level is below a defined threshold.

Going back to Algorithm 2, we continue exploring the network if the attacker did not acquire new knowledge items after scanning (no need to take the scan action into account in this case) or a new attack state referring to scanning the network was created, and none of the stopping conditions on κ_t, κ_b, and κ_a is satisfied. The attacker then tries to gain access to reachable equipment either by leveraging the access control policy ACCESSREMOTENODE (defined in Algorithm 4) or exploiting vulnerabilities remotely EXPLOITREMOTEVULNERABILITIES (defined in Algorithm 5). Knowing that a remote vulnerability exists, it is exploited by the attacker only if he has the required skill level and the results of the exploit match his preferences. We explore the network using a depth-first search algorithm. Therefore, when backtracking, the values changed at each stage of an attack path need to be restored.

Algorithm 4

Input: $\{t, m\} \in \mathcal{T}$, z, type_of_access

```
1  function accessRemoteNode(t, m, z, type_of_access)
2      l ← getAccessLevelsOnRemoteNode(t, m, z, type_of_access)
3      if l ≠ ∅ then
4          Update(κ_t, κ_a)
5          if (κ_t ≤ κ_t^M & κ_a < κ_a^M & ¬stateExists) then
6              x ← createNewAttackState
7              G ← G ∪ {x}
8              constructAEM(m, z)
9          end if
10         Restore(κ_t, κ_a)
11     end if
12 end function
```

Algorithm 5

Input: $\{t, m\} \in \mathcal{T}$, z

```
1  function exploitRemoteVulnerabilities(t, m, z)
2      for each γ ∈ Γ_m^r do
3          if z.hasDiscovered(γ)& ¬z.hasExploited(γ) & z.wantsToExploit(γ) then
4              if z.canExploit(γ) then
5                  k ← Exploit(γ)
6                  if k ≠ ∅ then
7                      z.hasExploited(γ) ← TRUE
8                      Update(κ_t, κ_b, κ_a)
9                      if (κ_t ≤ κ_t^M & κ_b ≤ κ_b^M & κ_a < κ_a^M & ¬stateExists) then
10                         K(z) ← K(z) ∪ k
11                         x ← createNewAttackState
12                         G ← G ∪ {x}
13                         if m.compromisedServices ∉ C_s then
14                             exploitRemoteVulnerabilities(t, m, z)
15                             if κ_a < κ_a^M then
16                                 constructAEM(m, z)
17                             end if
18                         end if
19                         K(z) ← K(z)\k
20                     end if
21                     Restore(κ_t, κ_b, κ_a)
22                     z.hasExploited(γ) ← FALSE
23                 end if
24             end if
25         end if
26     end for
27 end function
```

4.6.2 Attack Impact Evaluation

We associate a payoff for each action of the attacker in the control system. Depending on the objective of the analysis of the attack graph, payoffs can be associated to impacts on equipment and services running in the system. In this section, we consider impacts on both layers of the control system architecture.

Let $\mathcal{W}_{\tau_i} = \langle W^f_{\tau_i}, W^e_{\tau_i}, W^h_{\tau_i} \rangle$ represents the impact of compromising equipment τ_i, which depends on the access level l_j of the attacker on that equipment. We decompose the impact of an attack into the following three dimensions: the financial impact $W^f_{\tau_i}(l_j)$, the environmental impact $W^e_{\tau_i}(l_j)$, and the human impact $W^h_{\tau_i}(l_j)$. $W^e_{\tau_i}(l_j)$ refers to any consequence on the environment if the attacker achieved an access l_j on equipment τ_i. Furthermore, an attack can cause human injuries for the people located near the compromised equipment, which we quantify by $W^h_{\tau_i}(l_j)$. Similarly, let $\mathcal{W}_{\delta_i} = \langle W^f_{\delta_i}, W^e_{\delta_i}, W^h_{\delta_i} \rangle$ refer to the impact of compromising service δ_i. However, in this case, the impact $\mathcal{W}_{\delta_i}$ does not depend on the access level of the attacker but only on the state of the service (compromised/not compromised). We make the following assumptions:

Assumption 4.4. *On each equipment, we assume a total ordering of attack impacts with respect to access levels.*

Let l_j and l_k be two access levels on equipment τ_i. Following Assumption 4.4, if $l_j \leq l_k$, we have $W^r_{\tau_i}(l_k) = W^r_{\tau_i}(l_j) + \nu^r$, where $\nu^r \geq 0$ and $r = \{f, e, h\}$. For example, let us assume that the attacker has already an access level as *user* on equipment τ_i, and as a result he caused a financial impact $W^f_{\tau_i}(user)$. The attacker will inflict a financial impact $W^f_{\tau_i}(root)$ - $W^f_{\tau_i}(user)$ if he got the *root* access on equipment τ_i, where $W^f_{\tau_i}(root)$ refers to the financial impact of getting only the *root* access on τ_i.

Assumption 4.5. *Interdependencies in terms of impacts could exist between different equipment or between different services.*

A result of Assumption 4.5 is that the order in which an attacker compromises equipment will have an impact on the payoff he will get after each attack. For example, let us suppose that we have two equipment a and b with one access level on each equipment. If we assume that the financial impact W^f between these equipment is interdependent and $W^f_a < W^f_b$, the attacker will get $W^f_b - W^f_a$ as a payoff when compromising b if he had already compromised a. This assumption applies to each type of impact. Equipment a may be interdependent with equipment b with respect to the financial impact, but interdependent with equipment c with respect to the environmental impact. The same analysis can be conducted for the services running in the system. Let $\mathscr{T}^r$ and $\mathscr{D}^r$ refer to the family of sets of $\mathcal{T}$ and Δ respectively that are interdependent w.r.t. impact type r, where $r = \{f, e, h\}$. For example, τ_i and τ_j are financially interdependent in terms of impact if $\exists T \in \mathscr{T}^r$ s.t. τ_i, $\tau_j \in T$.

Assumption 4.4 and 4.5 have important implications as the order in which the attacker proceeds will impact the payoff he will get after each action. This will be useful in the

next chapter when we tackle the problem of managing the security risk by optimizing the distribution of the available defense resources. By computing only the relative increase in each type of impacts after each attack, we optimize our defense strategy by focusing only on the residual risk in each stage of an attack path.

Let $\eta = \langle \eta^f, \eta^e, \eta^h \rangle$ refer to the payoff following an attack action, where η^f, η^e, and η^h refer to the financial, environmental, and human impact. Let $\mathcal{T}^c$ and Δ^c refer to the set of equipment and services that have already been compromised before the execution of an attack action on equipment τ_i. We assume that the action resulted in getting an access level l_j on τ_i and compromising the service δ_i. Let $\tau_i \in T \in \mathscr{T}^f$ and $\delta_i \in D^f \in \mathscr{D}^f$. Therefore, the financial payoff of executing that action can be computed as follows:

$$\eta^f = \max\left(0, W^f_{\tau_i}(l_j) - \max_{\tau_j \in \mathcal{T}^c \cap T} W^f_{\tau_j}(l_t)\right) + \max\left(0, \max_{\substack{j,\ \delta_i \to \delta_j \\ \delta_j \in (\Delta \setminus \Delta^c) \cap D^f}} (W^f_{\delta_i}, W^f_{\delta_j}) - \max_{\delta_k \in \Delta^c \cap D^f} W^f_{\delta_k}\right) + \sum_{\substack{j,\ \delta_i \to \delta_j \\ \delta_j \notin \Delta^c \cup D^f}} W^f_{\delta_j} \quad (4.1)$$

The first part of Equation 4.1 takes into account the financial impact of the attacker action on equipment τ_i. In this part, l_t refers to the highest access level acquired by the attacker on equipment τ_j. The second part of the equation computes the financial impact as a result of compromising the service δ_i. In this case, we take into account the cascading impact of compromising a service δ_i on the set of services δ_j that depend on it. The environmental and human impact can be computed in a similar way.

Finally, in our model so far, we do not consider the existence of backups for the services running in the system. The existence of such backups does not have major impact on the process of generating the attack graph except for the assessment of the impact of each attacker action. For example, when a backup service δ_2 exists for the service δ_1, the attacker needs to compromise both services in order to have an impact on the system associated with the fact that this type of services is unavailable.

4.6.3 Vulnerability Dependency Graph

In our model, the process of exploiting the vulnerabilities in the system depends on the profile of the attacker. In addition, we have assumed that the time needed to exploit a vulnerability depends on the intrinsic characteristics of that vulnerability and the attacker skill level. This assumption holds in many practical scenarios. However, in some cases, the time needed to exploit a vulnerability can also depend on the set of vulnerabilities that have already been exploited by the attacker in the system. In this case, the sequence of vulnerabilities that are being exploited plays an important role in identifying the set of actions that can eventually be executed by the attacker in the system before the next maintenance period. When such vulnerability dependency graph exists, it is possible to use

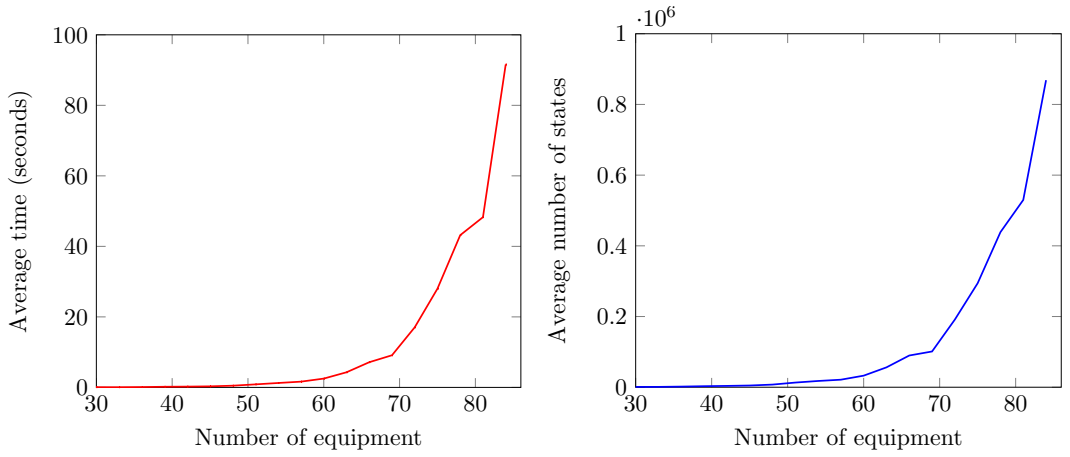


FIGURE 4.4: AEM generation performance for unlimited attack paths lengths

the information in the graph for the assessment of the time needed to exploit a vulnerability in our model.

4.6.4 Performance

We implemented the attack graph generation model in C++. For the performance evaluation, we use a Mac OS X 10.8.5 with a 2.6 GHz Intel Core i5 processor and 8 GB of RAM. As we have mentioned earlier, we are interested in generating all possible combinations of the actions that an attacker can execute in the system. This has important consequences on the number of equipment in the control system that we can model. We evaluate the performance of the tool on an architecture of a control network with four hierarchical levels. Equipment in each level are separated by gateways. In the same hierarchical level, we consider that each equipment can interact with four other equipment. When equipment τ_1 can interact with equipment τ_2, the access control policy is defined such as a user on τ_1 with an access level of *user* can gain access to τ_2. A vulnerability exists on each equipment in the system. We consider the worst case where the attacker is capable of exploiting all vulnerabilities. Since the evaluation of the dependencies between the services is performed prior to the attack graph generation, to simplify the analysis, we omit the existence of a service layer in this scenario. We can immediately realize that we have set the configuration of the architecture such that every equipment and vulnerability can be compromised and exploited by the attacker respectively. This is a strong assumption and constitutes a worst-case scenario, which is often not true in practice.

For a fixed total number of equipment in the system, under the assumptions that we have laid out, we generate 40 random architectures and average the result in terms of the time needed to construct the attack graph and the number of states in this graph. Fig. 4.4 depicts the result when we vary the total number of equipment in the system. As we can

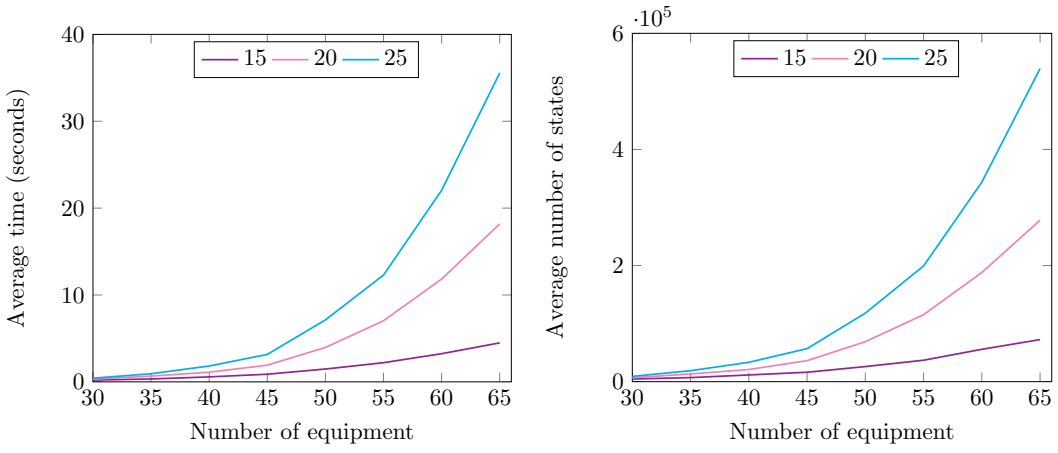


FIGURE 4.5: AEM generation performance for various attack paths lengths

expect, the complexity of generating the attack graph is exponential with respect to the number of equipment in the system. However, within this limit on the total number of equipment, our model can be applied to assess the security of industrial control systems, which have similar restrictions.

In practice, the number of actions an attacker executes in the system will impact the probability of him being detected. Fig. 4.5 shows the average AEM generation time and average number of states in the attack graph for 200 random configurations (under the previous conditions) for various constraints on the length of attack paths. We can notice that the complexity and the time needed to construct the attack graph decrease when the maximum number of attack actions allowed in each attack path decreases. As we have mentioned in Section 4.6.1, when defining equivalency between two states in the attack graph in our implementation, we focus on what the attacker is capable of achieving with his set of knowledge items instead of focusing on what the attacker has already achieved. This approach is highly optimized for the case where there are no constraints on the number of attack actions in an attack path. As a result, we notice that the time needed for generating the attack graph under maximum attack path length constraints of 20 and 25 is greater compared to the unconstrained case in Fig. 4.4.

Finally, we fix a maximum length of an attack path in the attack graph to 10 and average the result of 200 random configurations. Fig. 4.6 depicts the result when we vary the total number of equipment in the system. In this case, for a control system composed of 200 equipment, the average generation time of the attack graph is around 1.2 seconds. Therefore, the model scales well under such constraints on the maximum number of attacker actions allowed in each path in the attack graph.

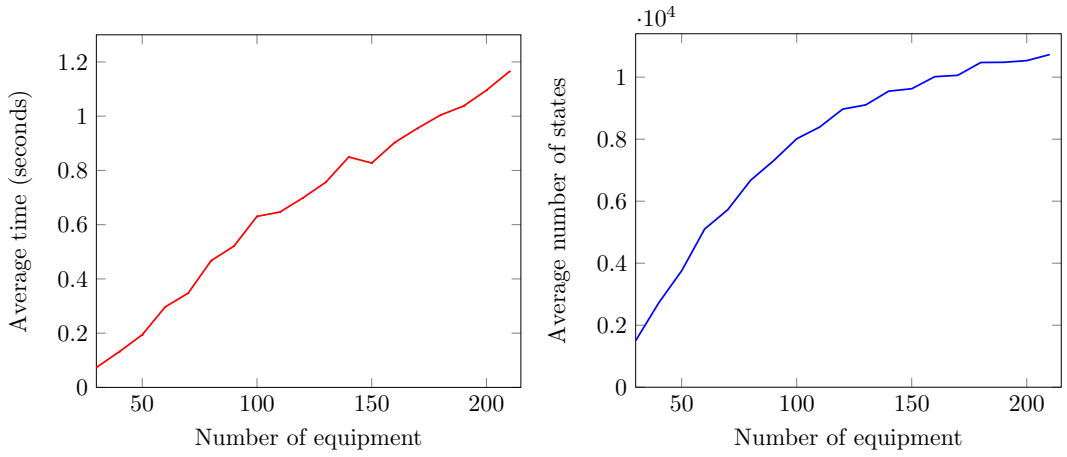


FIGURE 4.6: AEM generation performance for a maximum attack path length of 10

4.7 Conclusion

The life cycle of industrial control systems vary from several years to several decades. With new equipment with increased communication capabilities and a potential exposure to unsecured networks such as the internet, it is important to conceive a model that allows asset owners to quantify the risk that unpatched vulnerabilities and poor security configurations pose to their systems. The type and scope of a cyberattack depends on the profile of the attacker and his knowledge of the architecture and security configuration of the system. An intelligent attacker will leverage his accumulated knowledge acquired after a series of attacks to compromise additional equipment and services in the system. In this chapter, we presented the attack execution model, which is a type of attack graphs that given the architecture of the system and an attacker profile, generates all attack executions that could be carried out by the attacker in the system before the next maintenance period.

In the next chapter, using the output of our attack graph, we present an approach to find optimal security policies that reduce the risk of attacks on the industrial control system. In particular, we propose optimal patch strategies of the vulnerabilities that could be exploited by the attacker. These strategies take into account, among other factors, the interdependencies between industrial processes and the overall cost of patch deployment.

Chapter 5

Optimal ICS Security Policies

Enforcing security in a network always comes with a tradeoff regarding budget constraints, entailing unavoidable choices for the deployment of security equipment over the network. Therefore, finding the optimal distribution of security resources to protect the network is necessary. One of the most significant challenges to secure industrial control systems is managing the process of patching vulnerabilities. In general, in this type of systems, maintenance periods are fixed where operators take advantage of stopping the system to patch critical vulnerabilities. However, the long time between two consecutive maintenance periods could expose the system to additional threats of newly discovered vulnerabilities. In this case, the operator needs to decide whether to stop the system to patch critical vulnerabilities for their significant impact on the system, or wait until the next maintenance period to apply these patches. In the previous chapter, we proposed a model for generating an attack graph that we referred to as the Attack Execution Model (AEM). The AEM represents the set of actions executed by an attacker to compromise equipment and services in the system. In this chapter, we leverage the information present in the AEM to tackle the problem of finding the optimal security policy that offers the maximum level of system protection.

5.1 Introduction

In the last few years, standard bodies, industry groups, and governments started proposing recommendations to secure critical infrastructures. In general, these recommendations are best-known practices to secure this type of systems. In the US, the Department of Homeland Security (DHS) proposed in 2009 the National Infrastructure Protection Plan (NIPP) [Dep09b]. NIPP provides a unifying framework to enhance the security of critical infrastructures in order to prevent or mitigate the effects of attacks, and strengthen the response and recovery in the event of an emergency (e.g. a cyber attack). In addition, the U.S. National Institute of Standards and Technology (NIST) had led the development of

cyber security guidelines for the smart grid [Nat14b] and recently proposed a framework to improve the security of critical infrastructures [Nat14c]. In Europe, the European Network and Information Security Agency (ENISA) published a report [Eur12a] listing a set of minimum security measures to improve the cyber security and resilience of smart grids. One of the objectives of the report is to ensure minimum requirements level on the security and resilience of smart grids across member states, thus improving compliance and reducing operational costs.

This set of recommendations gives general guidelines to protect the smart grid but does not address the problem of optimizing the available defense resources to achieve the best protection. In addition, the smart grid is divided into seven domains [Nat10]. Each domain has its own functional constraints and security requirements. Depending on the result of the security risk assessment, it may not always be necessary to secure all system equipment to satisfy the security objectives and protect the system against a particular type of attacks. For example, it has been shown that it is sufficient to secure a selected meter measurements to prevent attackers from launching false data injection attacks that compromise the power system state estimation [BZ11].

Multiple approaches were proposed to improve the security of the power system. Some approaches [QWT+11] proposed to add intelligent equipment to the grid to have a secure reconfigurable system supported by fault-resilient real-time controls. The design of the system should allow it to quickly respond to natural and intentional attacks on the power grid by coordinating actions at the local and system levels. However, a strategic and optimal deployment of a security policy needs to take into account the different methods that the attacker can use to compromise the system.

The operator is generally interested in identifying the set of security countermeasures that needs to be deployed to secure the smart grid. In particular, the control system of the smart grid, being a critical component, needs to be secured. However, the configuration and deployment of defense countermeasures efficiently to optimize attack detection and mitigation remain a challenging task. In particular, in networks providing critical services, the deployment depends on the types of interdependencies that exist between vulnerable network equipment. In addition, without an assessment and understanding of the attacker capabilities, preferences, and skill set, the deployment of security hardening solutions may not achieve the optimal protection. In this chapter, we leverage the information present in the attack graph generated in Chapter 4 and present an approach to find the optimal security policy that guarantees that the defender security objectives are satisfied.

The remainder of this chapter is organized as follows. We discuss related work in Section 5.2. In Section 5.3, we present a graph theoretic approach to find the optimal set of vulnerabilities that needs to be patched in the system at a minimum cost. However, this approach focuses on only one objective and is in general impractical. Therefore, we present in Section 5.4 an approach based on Constrained Markov Decision Processes (CMDPs) to tackle the problem of finding an optimal security policy given a set of constraints. We show

how to build the CMDP based on the information present in the attack graph generated in Chapter 4. Finally, we conclude the chapter in Section 5.5.

5.2 Related Work

In information systems, optimizing the placement of defense measures has been an active research domain. In particular, in control networks of critical infrastructures, an important issue is the choice of the optimal combination of security hardening schemes to optimize the defense budget in order to secure the network while satisfying system constraints [AMGC09]. In the state of the art, we distinguish two main sets of related work. In these two sets, attack trees and attack graphs are used as frameworks to assess the impact of selecting a set of defense countermeasures on the security of the system, which provide a basis for selecting the optimal set. Before presenting each set of related work, we start by arguing the importance of metrics to the process of searching for an optimal security strategy.

5.2.1 Importance of Security Metrics

To find the optimal solution, one needs to be able to compare the different available options. This is not possible without defining criteria, parameters, or metrics for making the comparison. One the most important elements for assessing the exposure of the system to attacks is the notion of risk. D. Hubbard define the risk in general as being the probability and magnitude of a loss, disaster, or other undesirable event [Hub09]. Therefore, in the security domain, the efficiency of a security strategy can be measured by the amount of risk on the system that is removed when deploying it. In critical infrastructures, researchers proposed a set of tools and methods to quantitatively reduce the security risk. For example, McQueen et al. [MBFB06] propose a method to calculate a quantitative risk reduction estimate of security enhancements applied to a specific SCADA system. The risk is reduced when the value of the metric *time-to-compromise* a device decreases. The metric is a function of known vulnerabilities and attacker skill level. Granadillo et al. [GJDC12] propose a method to find the optimal combination of countermeasures that needs to be deployed in a system by maximizing the Return on Response Investment (RORI) index, which represents its cost-effectiveness ratio.

Another important benefit of using security metrics is to identify vulnerable components in the system that needs to be hardened. Wei and Ji [WJ10] propose metrics to estimate the resilience of control systems. Such metrics include (i) the identification time, which is the time needed to identify an incident, and (ii) the protection time, which is the time the system can withstand an incident without performance degradation. These metrics could offer some insights on the security state of the system, especially if the incidents include cyber attacks. Wang et al. [WIL+08] propose probabilistic metrics that can be computed using the output of an attack graph and refer to the success likelihood of the attacker at each

stage of his attack. Finally, the notion of *attack surface*, first proposed by Manadhata and Wing [MW04], allows the comparison of the relative security of two versions of a system.

5.2.2 Approaches Based on Attack Trees

An attack tree has a similar structure to a fault tree. In an attack tree, the root node represents the goal of the attacker. The children of a node, referred to as subgoals, are refinements of the goal represented by that node. They are connected with their parent either with an AND or an OR condition. For example, for an AND condition (resp. an OR condition), the goal is achieved if all subgoals are achieved (resp. one of the subgoals is achieved). The original versions of attack trees were expanded to include defense countermeasures. The work of Bistarelli et al. [BFP06] on *defense trees* was the inspiration to include countermeasures in the tree structure. Arcs between the countermeasures and the subgoals they impact are drawn. Therefore, the success probability of the attack will depend on the set of deployed countermeasures. The approach was used to assess the security of Wide Area Networks (WANs) used to operate electric power systems. However, for confidentiality reasons, only the results of the assessment on a fictitious example are presented [SEN09]. Roy et al. [RKT10a, RKT10b, RKT12] propose the notion of attack countermeasure trees (ACTs) to model attacks and countermeasures. Contrary to defense trees in which countermeasures are placed on the leaf nodes, countermeasures in ACTs can be placed on any node. Kordy et al. [KMRS11] propose the notion of an attack–defense tree in which the possible actions of the attacker and the defender are taken into account.

Dewri et al. [DPRW07] investigated the problem of securing a system by choosing an optimal set of security measures within a certain budget while minimizing the residual risk. An attack tree, generated using an in-house tool, is used to derive a solution to the problem. The multi-optimization problem is solved using a genetic algorithm. For each possible solution, sensitivity analysis is conducted with respect to failures of a selected set of deployed security controls. In this approach, the security planning is static. The model does not capture the case where during run time, the system administrator may need to revise the security planning based on emerging security conditions.

The papers in the related work that we have presented so far do not take into account the interactions between the attacker and the defender when searching for a solution to the defense optimization problem. Using a similar framework to [DPRW07], Dewri et al. [DRPW12] analyzed the impact of the interactions between the attacker and the defender on finding the optimal set of security measures. The equilibrium strategies of players are found using competitive co-evolution. This limitation is also addressed by Zonouz et al. [ZKSY09] in RRE, an engine to compute the optimal response to a given action by the attacker. The interactions between the system and the attacker are modeled as a two players Stackelberg stochastic game. Attack response trees (ARTs) are used in the process of computing the optimal strategies, and are based on the attack consequences and incorporate

possible countermeasures actions against attacks. Each node in the tree represents the consequence of an attack. Each ART is transformed into a Partially Observable Competitive Markov Decision Process (POCMDP), where each node represents a security state of the network, and solved to find the optimal response action.

5.2.3 Approaches Based on Attack Graphs

Attack graphs are used to find solutions to a number of security related problems such as correlating and predicting intrusion alert [WLJ06], optimizing the placement of IDS sensors [NJ08], and forensic analysis [CSW12]. In addition, a number of researchers have leveraged information present in attack graphs to identify critical points in the system and provide a set of defense recommendations. For example, Swiler et al. in [SPEC01] use their attack graph to identify the set of near-optimal shortest paths indicating the most exploitable components of the system configuration. The nodes and the edges that appear most frequently in the attack graph are identified as critical, and can be used in subsequent analysis to suggest defense placement. Noel et al. [NJWS10] simulate attack graphs through Monte Carlo methods. The overall security of the network is measured through the propagation of attacks likelihoods, which is used to score risk mitigation options in terms of maximizing security and minimizing cost.

Most approaches of network hardening using attack graphs focus on identifying the minimum set of exploits that needs to be removed to protect the system [AWK02, SHJ+02]. Jha et al. [JSW02] interpret their attack graph as a Markov Decision Process (MDP) to compute the success probability of the attacker. In addition, they analyze the attack graph to find the minimum critical attack set that needs to be thwarted to protect the network. They define the problem as a Minimum Hitting Set problem and propose a greedy algorithm to find a solution, which will not always yield the optimal set. Similarly to Jha et al. [JSW02], Sheyner et al. [SHJ+02] use the information in their attack graph to perform a probabilistic reliability analysis to determine the likelihood that the intruder will succeed. In addition, they propose an algorithm to determine the minimal set of atomic attacks whose prevention would guarantee that the intruder would fail. Based on the information in their Predictive graph, Lippmann et al. [LIS+06] identify the set of vulnerabilities that needs to be patched in order to minimize the percentage of hosts in the network on which the attacker can obtain user or administrator-level access. In NetSPA, Ingols et al. [ILP06] compute for each individual prerequisite (representing either a reachability group or a credential) in their attack graph, which vulnerability instances need to be removed in order to prevent the attacker from reaching the prerequisite, and which states the attacker cannot reach with the absence of that prerequisite.

Based on the information in their attack graph, Wang et al. [WNJ06] focus on hardening measures on initial conditions that are not implied by the exploitation of any vulnerability in the system. Disabling these conditions will prevent the attacker from achieving his goal. Among the sets that satisfy this objective, the solution is the one with the minimum deployment cost. The assumption that such conditions can be disabled independently, which is not

always the case, was later dropped by Albanese et al. [AJN12]. Their hardening strategy focuses on the set of initial conditions that needs to be disabled such that a target set of conditions in the attack graph cannot be reached. An approximation algorithm is proposed to find a hardening solution at a minimum cost.

Almohri et al. [AYWO15] base their analysis on the attack graph generated through MulVal [OBM06]. Based on initial probability values on a set of nodes in the attack graph (representing success probabilities at fact nodes), they define a method to compute the probability that an attack succeeds. Their objective is to find the optimal placement of a security countermeasure in the system in order to minimize the maximum value of the expected chance of a successful attack. The approach does not take into account the impact of attacks in the system and becomes complex when trying to find the optimal placement of multiple countermeasures in the system. Finally, Poolsappasit et al. [PDR12] compute a risk mitigation plan using the information in a Bayesian attack graph. Their objective is to minimize the implementation cost of security controls and maximize the gain of implementing a security plan. They formulate the problem as a multi-objective optimization problem and use a genetic algorithm to find a solution.

5.2.4 Other Approaches

In this section, we present a set of related work on security assessment, optimal defense hardening, and automatic response systems to intrusions. With respect to security assessment, specific formalisms and frameworks were adapted from other domains to assess the security of a system. For example, countermeasures were added to a Bayesian network for cyber security analysis [ES09], and the result is referred to as a Bayesian defense graph [SEJ09]. Boolean logic Driven Markov Processes (BDMPs), invented by Bouissou and Bon in 2003 [BB03] as a modeling formalism for dependability assessment, was later adapted to enable security modeling by Pièrre-Cambacédès and Bouissou [PCB10a, PCB10b]. It combines concepts from fault trees and Markov models and allows the evaluation of quantitative security metrics.

Gupta et al. [GRCC06] study the problem of finding a security policy that covers the maximum set of exploitable vulnerabilities in the system at a minimum cost. They refer to residual vulnerabilities, the vulnerabilities that are not covered by the security policy or the ones introduced after implementing it. The multi-objective optimization problem, solved using a genetic algorithm, aims at minimizing the cost of deploying a security policy while minimizing the set of residual vulnerabilities. The two objectives are weighted and combined to form a single objective function. Durkota et al. [DLBK15b] study the problem of placing honeypots in the network to deceive the attacker and detect attack attempts. The attacker knows the number and the types of honeypots added to the network but not their locations. The interaction between the attacker and the defender is modeled as a Stackelberg game in which the defender is the leader. The attacker's strategy is to select an optimal attack plan given his limited information. The authors use a finite horizon MDP, constructed from the attack graph, to find the attacker's best response. In [DLBK15a], Durkota et al. study a

similar problem where the attacker has imperfect information about the original network before the honeypots are added. In this case, the defender seeks an optimal randomized honeypot deployment.

Intrusion Response Systems (IRSs) in the literature are classified depending on a number of criteria. An IRS can be proactive by anticipating the threat of an attacker or reactive by selecting adequate responses following intrusions. A response action can be statically mapped to a set of intrusion alerts or be selected based on its deployment cost. However, the cost model used to compute the cost of a response can be a static model or a dynamic model that depends on the current state of the network (e.g. number of users, running processes, etc.). Similarly, the risk assessment model for deploying security countermeasures can be a static or a dynamic model. For example, Gehani and Kedem [GK04] present a real-time risk management system, focusing on the access control policy on a single host. Checks are performed before granting access to resources, which reduces the probability of compromising critical resources in the presence of a threat.

Lee et al. [LFM+02] propose a cost-sensitive model to detect intrusions. The model makes a tradeoff between the impact of the attack and the response cost, and use machine learning techniques to produce detection models. In [Car01], C. Carver proposes an adaptive agent-based intrusion response system (AAIRS). The generated response plan depends, among other factors, on the policy constraints, the attacker type, and the time, type, and impact of the attack. Balepin et al. [BMRL03] represent resources in the system as a directed graph. Response actions are associated to nodes that represent system objects (e.g. a file, a process, etc.), which restore their functionalities in case of intrusions. Costs are associated to response actions. The optimal response is the one that yields the maximum benefit at a minimum cost. A possible decision criteria proposed by the authors is the minimax criterion in which the optimal strategy is the one that minimizes risk under the most unfavorable conditions.

Foo et al. [FWM+05] present an automated response mechanism ADEPTS, which uses a graph of intrusions to determine response locations. The nodes in the graph represent individual services in the system and the edges represent intrusion-centric channels (i.e. how the intrusion spreads between two nodes). Based on the received IDS alerts, the nodes in the graph that have been likely compromised are identified. Using this information, the locations of the response actions are computed and the appropriate response is chosen depending on its effectiveness against the attack, and its perceived disruptiveness to legitimate users of the system. In [ML10], Mu and Li present an intrusion response decision-making model based on hierarchical task network planning. The model takes into account the response planning and the time to execute each response. The best response is selected depending on the current risk index of the network. However, the authors do not detail how the effects of a response are computed, which are one of the main criteria to select the best response to an intrusion.

Finally, Miehling et al. [MRT15] formulate the problem of finding the best response to intrusions as a Partially Observable Markov Decision Process (POMDP). The authors use a Bayesian attack graph to describe the progress of the attacker in the system. At a given moment, the defender can only have a partial observation of the state of the network as a result of the attacker actions. The Bayesian attack graph is considered an input to the model. In addition, contrary to our model in which the set of countermeasures at a given state directly reflect the defender's decision to thwart the next potential action of the attacker, the authors' model considers the power set of all possible individual countermeasures at the disposal of the defender at each time step. In practical scenarios, this assumption significantly increases the size of the action space of the defender. In addition, finding a solution to the problem in large state spaces is still a challenge for the application of this model in realistic scenarios.

The models that we have seen so far for security hardening and optimization of the defense response to intrusions do not take full advantage of the additional defender's knowledge about the system under attack. In particular, the information about the capability of the attacker and the potential impact of his actions on the system will determine the nature and the type of the defender's response. This information, coupled with financial constraints such as the defense budget, and technical constraints such as critical equipment and services that need to be protected or security countermeasures that can only be applied on a subset of the system components, will also influence the defender's decision. The information about the type and the impact of the potential attacker's actions can be captured in the attack graph generated in Chapter 4. Contrary to the classic approaches, the defender can use this type of information to improve his decision-making process by choosing defense countermeasures that limit the action space of the attacker depending on his current state in the system. In addition, the defender can weigh the efficiency of deploying a countermeasure in the present with the potential risk of future attack attempts while taking into account the financial and the technical constraints in the system. In the rest of this chapter, we will tackle the problem of security hardening and the optimization of security policies in an industrial control system using the attack graph generated in Chapter 4. First, we present a graph theoretic approach for identifying the set of vulnerabilities that needs to be patched in the system at a minimum cost. Second, we present an approach, based on constrained Markov decision process, to compute an optimal security policy that takes into account the potential actions of the attacker and guarantees that the defender's objectives are satisfied.

5.3 A Graph Theoretic Approach

In an attack graph, each path refers to the sequence of actions executed by the attacker in the system. In this section, we assume that we have an attack graph in which each edge represents the action of exploiting a vulnerability. This type of attack graphs can be constructed from our AEM by keeping the edges in the AEM that correspond to a vulnerability exploitation. In general, there is a cost associated to patching each one of these vulnerabilities. We will be interested in particular in the problem of patching the

optimal set of vulnerabilities with the minimum cost in order to remove all attack paths from a source to a set of target nodes.

Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ be an attack graph where $\mathcal{V}$ and $\mathcal{E}$ refer to the set of nodes and edges in $\mathcal{G}$. We assume that $\mathcal{G}$ can contain cycles where each edge e_i refers to the exploitation of a vulnerability and a node v_i represents an equipment or service in the system. The graph $\mathcal{G}$ is constructed based on the set of all attack paths from a source node s representing the initial starting point of the attacker, to a set of target nodes. We can consider a single final target node in our case by introducing a new dummy node d and connecting the set of target nodes to that dummy node. We will associate to each edge e_i a nonnegative capacity $c(e_i)$. In our case, $c(e_i)$ refers to the cost of patching the vulnerability associated with e_i. When we need to add an edge e_j to the graph that does not refer to a vulnerability exploitation (e.g. when connecting a target node to d), we set the capacity of that edge to infinity. We note that adding e_j will not affect the problem of finding the optimal set of vulnerabilities that needs to be patched at a minimum cost.

The attack graph $\mathcal{G}$ is constructed from the set of attack paths. Therefore, the same vulnerability could be represented with two different edges in $\mathcal{G}$. For example, let us consider two attack paths p_1 and p_2 in which vulnerability *vul* is exploited. Let $\mathbf{Pred}(vul, p_j)$ and $\mathbf{Succ}(vul, p_j)$ be the set of vulnerabilities that are exploited before and after *vul* on path p_j respectively. If $\mathbf{Pred}(vul, p_1) \neq \mathbf{Succ}(vul, p_2)$ or $\mathbf{Pred}(vul, p_2) \neq \mathbf{Succ}(vul, p_1)$, we cannot combine paths p_1 and p_2 in a graph-like manner featuring one instance of an edge referring to *vul*. As a result, multiple edges in the graph $\mathcal{G}$ may refer to the action of exploiting a particular vulnerability.

Let $Y = \bigcup_{i=1}^{M} Y_i$, where each Y_i refers to the set of edges that represent the same vulnerability in $\mathcal{G}$. Let $y : \mathcal{E} \to \mathbb{R}^+$, where $y(e_i) = r$ if $e_i \in Y_r$.

Our problem can be stated as follows:

Problem 5.1. *Given a weighted directed graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ and a source and target nodes s and d respectively, find the set of edges with the minimum weight that needs to be removed to separate the nodes in $\mathcal{G}$ into two disjoint sets A and $\mathcal{V} \backslash A$ where $s \in A$ and $d \in \mathcal{V} \backslash A$.*

The problem of finding the set of vulnerabilities that needs to be patched at a minimum overall cost to prevent the attacker from reaching his targets can be translated to a variation of a known problem, the max-flow min-cut problem.

5.3.1 Max-Flow Min-Cut Problem

We define the max-flow problem as follows:

Definition 5.1 (Max-flow problem)**.** *Let $G = \langle V, E \rangle$ be a directed graph where a nonnegative value $c(e)$, referred to as the capacity, is associated to each edge e of the graph. Let s and*

d refer to the source and sink nodes in G. A flow *f is a function satisfying the following constraints:*

- *Capacity constraint:* $0 \leq f(e) \leq c(e),\ \forall e \in E$
- *Flow conservation:* $\sum_{(v,u)\in E} f(v,u) - \sum_{(u,v)\in E} f(u,v) = 0,\ \forall v \in V \backslash \{s,d\}$

The Max-flow problem is the problem of maximizing $\sum_{(s,v)\in E} f(s,v)$.

Definition 5.2 (Cut)**.** *A* cut *B is a partition of V into two disjoint nonempty sets* $(A, V\backslash A)$.

In particular, an $s-d$ cut is a cut $(A, V\backslash A)$ s.t. $s \in A$ and $d \in V\backslash A$. The capacity of a cut $B = (A, V\backslash A)$ is given by $C(B) = \sum_{\substack{(u,v)\in E \\ u\in A,\ v\in V\backslash A}} c(u,v)$.

Theorem 5.1. *The maximum value of an $s-d$ flow is equal to the minimum capacity of any $s-d$ cut.*

For the proof of Theorem 5.1, the reader can refer to the proof of Theorem 10.3 in [Sch04]. Theorem 5.1 states that the maximum amount of flow passing from the source s to the sink d is equal to the minimum capacity of any cut that partition V into two disjoint nonempty sets $(A, V\backslash A)$, where $s \in A$ and $d \in V\backslash A$. In addition, the maximum flow can be found in polynomial time [Sch04].

5.3.2 Exact Solution

Finding an exact solution to Problem 5.1 efficiently depends on the set Y. We have the following theorem:

Theorem 5.2. *When $Y = \varnothing$, there exists a polynomial time algorithm to find the set of vulnerabilities that needs to be patched at a minimum cost to prevent the attacker from compromising any target node in $\mathcal{G}$.*

Proof. Finding a minimum cut to a graph G as defined in Definition 5.1 can be done in polynomial time [Sch04]. The proof of the theorem follows directly from the fact that we assumed that the capacity of an edge e in $\mathcal{G}$ represents the cost of patching the vulnerability associated to e. □

When $Y \neq \varnothing$, finding an exact solution to Problem 5.1 efficiently becomes more challenging. In this case, at least two edges in $\mathcal{G}$ refer to the same vulnerability. A more general form of this problem was formulated by Jegelka and Bilmes in [JB14]. They introduced the terminology of *minimum cooperative cut* to refer to the problem of finding the minimum cut when

interdependent relations between the weights of the edges in the graph exist. In its general form, the problem can be formulated using submodular functions[1]. Let t be a nonnegative, monotone, and non-decreasing submodular function that computes the cost of choosing a set of edges in the graph $\mathcal{G}$. Given a set of interdependent edges $\mathcal{E}_1 = \{e_1, e_2, ..., e_n\}$, we have $t(\mathcal{E}_1) \leq \sum_{i \in \mathcal{E}_1} t(e_i)$. In particular, in our case, $t(\mathcal{E}_1) = \frac{1}{|\mathcal{E}_1|} \sum_{i \in \mathcal{E}_1} t(e_i)$.

When $Y \neq \varnothing$, we have the following theorem:

Theorem 5.3. *The problem of finding the minimum cooperative cut in a graph $\mathcal{G}$ is NP-hard.*

Proof. The proof consists of a reduction from a graph bisection problem to the problem of finding the minimum cooperative cut. For a weighted graph $G = \langle V, E \rangle$, graph bisection consists of finding two disjoint sets of nodes V_1 and V_2 s.t. $V_1 \cup V_2 = V$ and $|V_1| = |V_2|$ while minimizing the capacity of the cut between these two sets. This problem is known to be NP-hard. For a complete proof, refer to [JB14]. □

It follows directly from Theorem 5.3 that Problem 5.1 is NP-hard when $Y \neq \varnothing$. In [JB14], Jegelka and Bilmes give several approximation algorithms for finding the *minimum cooperative cut* with different results in terms of efficiency. In the remaining of this section, we will be interested in evaluating the complexity of an approach that finds a solution to Problem 5.1 using the result of Theorem 5.1. Let $|\mathcal{V}| = K$, $|\mathcal{E}| = N$, and $Z \leq N$ be the number of edges that are interdependent in $\mathcal{G}$. These edges represent the action of exploiting $M \leq Z$ vulnerabilities in the system. Let Y_i be the set of edges that refer to vulnerability $vul(i)$.

When $Y = \varnothing$, the exact solution for Problem 5.1 can be computed in polynomial time. For example, Dinitz's algorithm [Din70] finds a solution in time $O(K^2N)$. When $Y \neq \varnothing$, finding a solution to the problem is NP-hard in general and requires exploring the space of all possible solutions. The search space is in the order of 2^{N-Z+M}, where $N - Z + M$ is the number of vulnerabilities that were exploited by the attacker in the system at some point during the attack process. Let $\mathcal{P}(I)$ be the power set of $I = \{vul(1), ..., vul(M)\}$. $\mathcal{P}(I)$ represents the set of all possible combinations of the M vulnerabilities exploited in $\mathcal{G}$.

We propose Algorithm 6 to find the exact solution to Problem 5.1. FindMininumCutSet($\mathcal{F}$) returns the minimum cut set of a graph $\mathcal{F}$ using one of the algorithms from the literature that has a polynomial complexity (e.g. [Din70]).

Theorem 5.4. *Given an attack graph $\mathcal{G}$, Algorithm 6 terminates and outputs the optimal solution to Problem 5.1.*

[1]Let F be a finite set. A function $m : 2^F \rightarrow \mathbb{R}$ is a submodular function if for all A, $B \subseteq F$, we have $m(A) + m(B) \geq m(A \cup B) + m(A \cap B)$.

Algorithm 6

Input: graph $\mathcal{G}$, Y
Result: A

```
1  function FINDOPTIMALSOLUTION(G, Y)
2      A = E
3      for each r ∈ P(I) do
4          F = G
5          for each edge e_i in F do
6              if y(e_i) ∈ r then
7                  e_i ← delete
8              else if y(e_i) ∈ I\r then
9                  c(e_i) ← +∞
10             end if
11         end for
12         B ← FINDMININUMCUTSET(F)
13         if Cost(B) + Cost(r) < Cost(A) then
14             A ← B ∪ r
15         end if
16     end for
17 end function
```

Proof. The number of combinations of the set of vulnerabilities $|\mathcal{P}(I)|$ is finite. Therefore, it is easy to show that the algorithm terminates. For the proof of correctness, we suppose that a set of vulnerabilities A_{opt} is the optimal solution to Problem 5.1. We assume the correctness of function FINDMININUMCUTSET($\mathcal{F}$). Let $R = \underset{r \in \mathcal{P}(I)}{\operatorname{argmax}} |r|$ s.t. R is part of the solution A_{opt}. From the hypothesis, we have $Cost(A_{opt}) = \min\limits_{r \in \mathcal{P}(I)} Cost(r) + \min\limits_{z \in \mathcal{P}(\mathcal{E} \setminus Y)} Cost(z)$ s.t. the set $r \cup z$ is the optimal solution to Problem 5.1. Since we test all the sets in $\mathcal{P}(I)$ in Algorithm 6, we will eventually end up testing the set R. The modifications on the graph $\mathcal{F}$ in the *for* loop in line 5 ensures that we will be able to find the set $A_{opt} \setminus R$. Otherwise, we find a contradiction, which is the existence of a minimum cut set for graph $\mathcal{F}$ that was not been returned by FINDMININUMCUTSET($\mathcal{F}$). In fact, after exiting the *for* loop at line 11, all edges e_i in $\mathcal{F}$ where the cost $c(e_i) \neq +\infty$ represent different vulnerabilities ($\nexists e_i, e_j$ in $\mathcal{F}$ and $c(e_i), c(e_j) \neq +\infty$ s.t. e_i and e_j represent the exploitation of the same vulnerability vul). □

In general, Algorithm 6 performs better than the exhaustive search for the optimal solution to Problem 5.1.

Theorem 5.5. *When the number of dependent edges in graph $\mathcal{G}$ exceeds 10 (i.e. $N - Z \geq 10$) and $K \leq N$, Algorithm 6 performs better than the exhaustive search for a solution to Problem 5.1 and has a complexity $O(2^M K^2 N)$.*

Proof. Let us suppose that we are using Dinitz's algorithm [Din70] in the function FINDMININUMCUTSET($\mathcal{F}$). In this case, the complexity at each step of Algorithm 6 is bounded by

$(K-Z)^2(N-Z)$. We consider all the sets in $\mathcal{P}(I)$. Therefore, in the worst-case scenario, we need to repeat the steps in the algorithm $\sum_{s=0}^{M} \frac{s!}{M!(M-s)!}(K-Z)^2(N-Z) = 2^M(K-Z)^2(N-Z)$ times. In the trivial search for the solution, we need to check 2^{N-Z+M} sets to find the optimal solution. We can easily show that when $N - Z \geq 10$, $2^M(K-Z)^2(N-Z) < 2^{N-Z+M}$. □

As we have seen in this section, finding an optimal solution to the problem of finding the set of vulnerabilities that needs to be patched using a graph theoretic approach is impractical in general. In addition, the defender may want to find a security policy in which patching a set of vulnerabilities is only an option among others to protect the system. In this case, other approaches need to be examined. The security policy may also depend on the current state of the attacker in the system and may need to satisfy a set of constraints. Therefore, the defender tries to answer the following question: "*Faced with an ongoing threat posed by an attacker who managed to gain a certain access to the system, which countermeasures I need to deploy to prevent him from compromising critical assets and further advancing in the system?*". In this case, the optimization of the security policy will depend on the uncertainty related to the type of actions that the attacker is attempting or will attempt to execute in the system. In the next section, we present an approach based on Constrained Markov Decision Processes (CMDPs) for finding an optimal security policy to protect the system. We leverage the information in the attack graph generated in Chapter 4 to construct our CMDP. In addition to the cost of deploying a defense countermeasure, we take into account additional constraints when defining our optimal security policy.

5.4 Approach Based on Constrained Markov Decision Processes

The choice of a defense strategy for a control system in a critical infrastructure depends on optimizing the available defense resources. In addition, depending on the context and the nature of the control system, different types of constraints should be taken into account. The defender may be interested to have some guarantees on the probability that an attacker will not be able to compromise a critical equipment. For example, in a wastewater treatment facility, it is important to secure the subsystem responsible of the water disposal for its significant environmental impact if it was the target of a cyber attack. In addition, financial constraints such as the cost to deploy the necessary security countermeasures may add another layer of complexity to the decision of choosing the optimal defense strategy. At the end, the optimal defense strategy must achieve a compromise between all technical, economical, and environmental constraints imposed by the system operator.

In Chapter 4, we proposed an attack graph, the AEM, in which each node represents a state of the attacker in the system. The state of the attacker includes the set of knowledge items (credentials, etc.) acquired by the attacker and his access levels on compromised

machines. In this section, given a set of constraints, we are trying to find the optimal security policy that we need to deploy to protect the system.

5.4.1 Constrained Markov Decision Processes

We associate a set of countermeasures that can be deployed by the defender for each action that can be executed by the attacker in the system. The objective of deploying a countermeasure is to either prevent the execution of a specific attack action or reduce its success probability. Therefore, the transition probabilities between the different states of the attacker in the AEM will be directly affected by the deployed set of security countermeasures. For example, patching a vulnerability targeted by the attacker will prevent him from exploiting it.

The complexity and criticality of the operations of an industrial control system give rise to different types of constraints that need to be satisfied. They range from technical, financial, to environmental constraints. Technical constraints encompass the constraints on equipment that provide critical functions in the system. The services provided by this type of equipment must be protected from any attempt to compromise their integrity and availability. Therefore, a security policy must guarantee tight bounds on the probability of compromising these services. Financial constraints refer to limits on the economical impact of attacks that can be tolerated by the system operator. In addition, the deployment of a security countermeasure will entail a cost that includes the effort needed to deploy this countermeasure and the cost of stopping the system. Given a constrained defense budget, the objective of the operator is to choose the optimal set of countermeasures that provides the best protection to the system. However, contrary to classic MDP problems in which we restrict the analysis on minimizing an objective function, we are trying to find an optimal security policy that takes into account an additional set of constraints. This class of problems is known as Constrained Markov Decision Processes (CMDPs) [Alt99].

We formally define a CMDP as follows:

Definition 5.3 (CMDP)**.** *A* Constrained Markov Decision Process (CMDP) *is a tuple* $\mathcal{J} = \langle \mathcal{X}, \mathcal{A}, \mathcal{P}, \omega, c, \beta \rangle$ *where:*

- $\mathcal{X}$ *is a finite set of states.*
- $\mathcal{A}$ *is a finite set of actions. We refer by* $\mathcal{A}(x)$ *the set of actions available at state* x*. Let* $\mathcal{Q} = \{(x, a) : x \in \mathcal{X}, a \in \mathcal{A}(x)\}$ *be the set of state-action pairs.*
- $\mathcal{P} : \mathcal{Q} \times \mathcal{X} \to [0, 1]$ *is the transition probability function.* $\mathcal{P}(x, a, y)$ *is the probability of moving from state* x *to state* y *if action* a *is chosen. If* $a \in \mathcal{A}(x)$*, we have* $\sum_{y \in \mathcal{X}} \mathcal{P}(x, a, y) = 1$*.*
- $\omega : \mathcal{Q} \to \mathbb{R}^+$ *is the immediate cost.*

- $c : \mathcal{Q} \to \mathbb{R}^{\mathrm{M}}$ *is a* M*-dimensional vector of immediate costs, related to* M *constraints.*
- $\beta \in \mathbb{P}(\mathcal{X})$ *is the initial distribution over the initial state, where* $\mathbb{P}(\mathcal{X})$ *refers to the set of all probability distributions over* $\mathcal{X}$*. For example, initially at* $t = 0$*, the probability of being in state* x *is given by* $\beta(x)$*.*

Definition 5.4 (Labeled CMDP)**.** *A* Labeled Constrained Markov Decision Process (LCMDP) *is a tuple* $\mathcal{J}' = \langle \mathcal{X}, \mathcal{A}, \mathcal{P}, L, \omega, c, \beta \rangle$ *where* $\langle \mathcal{X}, \mathcal{A}, \mathcal{P}, \omega, c, \beta \rangle$ *is a CMDP and* $L : \mathcal{X} \to \{Critical, Not\ critical\}$ *is a labeling function.*

We construct the labeled CMDP based on the information in the AEM generated in Chapter 4. A state x refers to the state of the attacker in the system and is labeled *critical* if he is capable of compromising critical equipment or services, and therefore causes severe impact to the control system. In the labeled CMDP, $\mathcal{A}(x)$ refers to the set of countermeasures available to the defender at state x. In the rest of this chapter, we will only consider labeled CMDPs. We will abuse notations and denote by CMDP the labeled CMDP.

The outcome of the execution of an action by the attacker depends on the countermeasure in place. For example, let us assume that the attacker is attempting to exploit a vulnerability γ to which a patch exists and is effective. If the patch was applied, the attacker will not succeed in his attempt, and therefore there is no security risk on the system. However, if the defender chooses not to apply the patch, the risk on the system will depend on the success likelihood of the exploit and the impact on the system as a result. In this example, we assumed that the patch is effective in thwarting the attack. In other scenarios, a countermeasure for a particular attack may have an *efficiency* measure that represents the probability that it will be able to thwart that attack successfully. If the attack was conducted successfully, the attacker can proceed in his attempt to compromise additional equipment and services in the control system.

We observe the system at times $t = 1, 2, ..., n$ where n refers to the time horizon (it could be finite or infinite). The actions of the defender at each state are chosen according to some decision rule, which we call a policy. The decision rule can be randomized. For example, choosing an action according to some probability distribution. In general, the policy of the defender may depend not only on the current state of the Markov chain and on the current time, but also on previous states and previous chosen actions.

Definition 5.5 (Policy)**.** *Let* $h_t = (x_1, a_1, ..., x_{t-1}, a_{t-1}, x_t)$ *refer to the history at time* t *representing the sequence of previous states and actions. A* policy $u = (u_1, u_2, ..., u_n)$ *is a sequence where* $u_t(a|h_t)$ *refers to the probability of choosing action* a *at time* t *if the observed history was* h_t*. We denote by* $\mathcal{U}$ *the class of all such policies.*

We identify in general three different classes of policies. A *Markov policy* is a policy in which the decision of choosing an action at time t in a state x_t depends only on x_t. A policy is *stationary* if the decision of choosing an action depends only on the state x and does not

depend on the time t. Finally, a *stationary deterministic policy* is a policy in which for each state x, we choose an action $a \in \mathcal{A}(x)$ with probability 1. In this case, we have a mapping from the state space to the action space $\mathcal{X} \rightarrow \mathcal{A}$.

5.4.2 Discounted Cost and Occupation Measure

Given an initial distribution β over the initial state and a policy u, we define in this chapter the finite horizon cost for a horizon n as a discounted cost $\Omega_\alpha(\beta, u)$ as follows:

$$\Omega_\alpha(\beta, u) = (1-\alpha)\sum_{t=1}^{n} \alpha^{t-1} E_\beta^u \omega(\mathcal{X}_t, \mathcal{A}_t) \tag{5.1}$$

E_β^u refers to the expectation operator and $\mathcal{X}_t$ and $\mathcal{A}_t$ refer to stochastic processes of the states and actions respectively. $\alpha \in (0,1)$ refers to the discount factor and reflects the fact that the future is given less importance with respect to the present. For example, the payoff of an attack in the present is greater than the expected payoff of the same attack in the future (i.e. due to uncertainties related to the future).

Similarly, we define the finite horizon cost related to the M constraints $C_\alpha^k(\beta, u)\ \forall k = 1, ..., M$ as follows:

$$C_\alpha^k(\beta, u) = (1-\alpha)\sum_{t=1}^{n} \alpha^{t-1} E_\beta^u c^k(\mathcal{X}_t, \mathcal{A}_t) \tag{5.2}$$

Our objective is to find an optimal security policy while respecting a set of constraints. Therefore, the optimal security policy is the solution of the following constrained optimization problem:

$$\min_{u \in U} \Omega_\alpha(\beta, u) \text{ subject to } C_\alpha(\beta, u) \leq S \tag{5.3}$$

where $S = (s^1, ..., s^M)$ refers to a M-dimensional vector representing constraints on the cost function $C_\alpha(\beta, u)$.

For a given initial distribution β and a policy u, let $\rho(x, a)$ refer to the occupation measure. The occupation measure corresponding to the policy u is the expected discounted time spent in different state-action pairs and is given by:

$$\rho(x, a) = (1-\alpha)\sum_{t=1}^{\infty} \alpha^{t-1} P_\beta^u(\mathcal{X}_t = x, \mathcal{A}_t = a),\ \forall x \in \mathcal{X},\ a \in \mathcal{A}(x) \tag{5.4}$$

Therefore, $\Omega_\alpha(\beta, u)$ and $C^k_\alpha(\beta, u)$ can be written as follows:

$$\Omega_\alpha(\beta, u) = \sum_{x \in \mathcal{X}} \sum_{a \in \mathcal{A}(x)} \rho(x, a)\omega(x, a) \tag{5.5}$$

$$C^k_\alpha(\beta, u) = \sum_{x \in \mathcal{X}} \sum_{a \in \mathcal{A}(x)} \rho(x, a)c^k(x, a) \tag{5.6}$$

In general, defining cost constraints in addition to the main objective function can be sufficient to model a significant number of real-world scenarios. However, in some cases, the system operator may want to have some guarantees that the probability of compromising a critical equipment or service is less than a certain threshold. The threshold may depend on the type and criticality of the equipment or service that needs to be protected. Using the formulation presented in the previous section, we can leverage the definition of cost constraints to define reachability constraints on some critical states in the system. For example, let $o \in \mathcal{X}$ be a state in which the attacker was able to compromise a critical equipment or service and $\mathcal{A}(o)$ the set of actions available at that state. Let O be the set of such states. We define a cost $c^m(z, a)$ $\forall z \in \mathcal{X}$, $a \in \mathcal{A}(z)$ as follows:

$$\begin{cases} c^m(z, a) = 1 & if\ z \in O \\ c^m(z, a) = 0 & otherwise \end{cases}$$

Let r^m be the reachability threshold for any of the states in the set O. This constraint can be defined using the cost constraint c^m as follows:

$$\sum_{o \in O} \sum_{a \in \mathcal{A}(o)} \rho(o, a)c^m(o, a) = \sum_{o \in O} \sum_{a \in \mathcal{A}(o)} \rho(o, a) \leq r^m \tag{5.7}$$

Finally, in a more general context, we can add to the CMDP new states to which we want to direct the attacker. These states could refer to fail safe or safe states that can be reached by an attacker without impacting the security of the system. For example, in an ongoing attack scenario, the defender may have to choose dynamically the set of countermeasures that needs to be deployed in order to have a certain level of assurance that the attacker will reach such states.

5.4.3 Optimization Problem

In order to solve the optimization problem in Equation 5.3, we use the primal linear programming formulation [Alt99]. Therefore, the optimization problem in Equation 5.3 can be formulated as the following linear program:

$$\min_{\rho} \sum_{x \in \mathcal{X}} \sum_{a \in \mathcal{A}(x)} \rho(x,a)\omega(x,a) \tag{5.8}$$

$$\text{subject to } \sum_{x \in \mathcal{X}} \sum_{a \in \mathcal{A}(x)} \rho(x,a)c^i(x,a) \leq s^i \quad \forall i = 1, ..., M$$

$$\text{and } \forall x \in \mathcal{X},\ a \in \mathcal{A}(x),$$

$$\sum_{a \in \mathcal{A}(x)} \rho(x,a) - \alpha \sum_{y \in \mathcal{X}} \sum_{a \in \mathcal{A}(y)} \rho(y,a)\mathcal{P}(y,a,x) = (1-\alpha)\beta(x)$$

$$\rho(x,a) \geq 0$$

For any state $x \in \mathcal{X}$ s.t. $\sum_{a \in \mathcal{A}(x)} \rho(x,a) > 0$, let $b_x(a)$ be a stationary policy defined as follows:

$$b_x(a) = \frac{\rho(x,a)}{\sum_{a' \in \mathcal{A}(x)} \rho(x,a')} \quad \forall a \in \mathcal{A}(x) \tag{5.9}$$

Theorem 5.6. *The optimization problem in Equation 5.3 is feasible if and only if the optimization problem in Equation 5.8 is feasible.*

Theorem 5.6 follows directly from Theorem 3.3 in [Alt99]. In addition, from Theorem 3.3 [Alt99], there exists an optimal solution ρ^* for the problem in Equation 5.8 if the optimization problem in Equation 5.3 is feasible, and in this case, the stationary policy defined in Equation 5.9 is optimal for the optimization problem in Equation 5.3.

When a solution to the optimization problem in Equation 5.8 cannot be found, we try to solve the problem by considering an upper bound on the number of attack actions that the attacker can execute in the system. Let U_b refer to the maximum number of attack actions that can be executed by an attacker in an attack path. U_b is the longest path in the attack graph generated in Chapter 4. We note that for a general graph, the problem of finding the longest path is NP-hard. However, the attack graph generated in Chapter 4 is a directed acyclic graph. In this case, the longest path problem has a linear time solution.

Since we are interested in the generation of all the attack actions executions, U_b can be retrieved during the process of generating the attack graph in Chapter 4. Algorithm 7 presents the function used to compute a general solution to the optimization problem. In Algorithm 7, we assume that the longest path in the attack graph has a length of at least 2. We start by trying to find a solution to the problem in Equation 5.8 when considering the original attack graph. If an optimal solution cannot be found, we keep reducing the length of the longest attack path until either an optimal solution for Equation 5.8 is found or we reach the lower bound of the length of the longest path L_b in the attack graph that we wish to consider.

Algorithm 7

Input: CMDP $\mathcal{J}$
Result: Optimal policy u

```
function findOptimalPolicy(J)
    if ∃ solution to Eq. 5.8 associated to J then
        Break
    else
        G ← AEM associated with J
        I ← U_b − 1
        repeat
            Modify G s.t. maximum path length equals I
            Update J according to G
            I ← I − 1
        until ∃ solution to Eq. 5.8 associated to J || I < L_b
    end if
end function
```

5.4.3.1 Number of Randomizations

In general, the stationary policy b, which is a solution to the optimization problem, can be a stochastic policy. However, we show in this section that there is an upper limit on the number of randomizations that we can have in the stationary policy b.

Definition 5.6 (Randomizations in a state)**.** *We say that under a stationary policy b, there are $\pi(x, b)$ randomizations in state x if there are exactly $\pi(x, b) + 1$ actions in $\mathcal{A}(x)$ for which $b_x(a) > 0$.*

Following Definition 5.6, the total number of randomizations under the stationary policy b is given by $\Pi(b) = \sum_{x \in \mathcal{X}} \pi(x, a)$. Therefore, there are no more that $\Pi(b)$ states in $\mathcal{X}$ in which randomization is used.

Theorem 5.7. *If the constrained optimization problem in Equation 5.3 is feasible, then there exists an optimal stationary policy b such that the total number $\Pi(b)$ of randomizations that it uses is at most M, where M is the number of constraints in the CMDP.*

For the proof of Theorem 5.7, the reader can refer to the proof of Theorem 3.8 in [Alt99]. Theorem 5.7 shows that there exists a stationary policy b that requires at most M randomizations. In this case, the stochastic nature of the policy is limited to only a certain number of states. Nevertheless, we will show in Section 5.4.5 how we can choose an action in such states deterministically.

5.4.3.2 Impact and Assessment of β

Evaluating the attack surface of a system is an important step when conducting a security risk assessment. Eventually, the security policy will depend on the attacker's capability of compromising the system from a given location. While the existence of an optimal policy for a classic MDP problem is independent of the initial probability distribution over the initial state of the system, it is not the case for a CMDP. This constraint has an important implication, as the existence of the solution to the problem is not always guaranteed and depends directly on the initial probability distribution over the initial state β.

In our case, β represents the probability distribution on the state space from which an attack is launched to compromise the system. In general, this information depends on the profile of the attacker and his capabilities. However, in the case where we want to protect the system from external attacks, we can assume that the equipment that are accessible from outside the network are known to the attacker (after scanning the network). In this case, we need to know from which location the attacker will probably launch his offensive and his most likely targets. To solve this problem, we introduce in Appendix C a constrained security game between an attacker and a defender in which both players have limited resources to attack and defend respectively. We analyze the interactions between the attacker and the defender and derive the optimal strategies of both players at the Nash equilibrium (NE). We assume that the attacker and the defender are rational and strategic players. Since the game that we analyze is a one-shot game, in the absence of any observation of the strategy of the defender before choosing an attack strategy, the best payoff the attacker can get given a best response strategy by the defender is when operating at the NE. In this case, the best strategy of the attacker is an optimal distribution of attack resources on system equipment. As we will show, this result can be easily translated to our case to find the probability distribution over the initial state β.

In the next section, we present the algorithms that allow us to construct the CMDP based on the information in the attack graph generated in Chapter 4.

5.4.4 CMDP Construction

In this section, we will present the algorithms that allow us to construct the CMDP using the information in the attack graph generated in Chapter 4. In fact, there are two possible representations of the CMDP depending on the way we interpret each state. A state in the CMDP could refer to a state of the attacker in the system. In this case, the actions associated to that state refer to the set of defense countermeasures that aims at preventing the attacker from further advancing in the system. We will refer to this CMDP as *CMDP Type I*. Another way to look at this problem is to consider that in each state of the CMDP, the attacker is attempting to execute an action. Therefore, the set of countermeasures associated to that state aims at protecting the system against that particular action of the attacker. We will refer to this CMDP as *CMDP Type II*. Each of the two ways of representing a state in

the CMDP can offer some advantages over the other one depending on the available set of defense countermeasures and the number of defenders that are deploying them.

Before presenting the two types of CMDPs, we give the following notations. Let $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ be the attack graph (AEM) where each node $v \in \mathcal{V}$ refers to an attacker state and each edge $e \in \mathcal{E}$ refers to an action executed by the attacker. For presentation reasons, we consider that the state of the system does not change (this assumption has no impact on the construction process). Let $\mathcal{B}$ be the set of available defense countermeasures to the defender. Let $\mathcal{J}$ refer to the labeled CMDP as defined in Definition 5.4. For each state in $\mathcal{J}$, let a_0 be the action that refers to the fact that no countermeasure has been deployed. Let $l_{\mathcal{E}}$ be a labeling function where $l_{\mathcal{E}}(e)$ refers to the attacker action associated with the edge e in the attack graph. Finally, let $\mathcal{L}_{\mathcal{E}}(v_i) = \bigcup_{e=(v_i,v_j)\in\mathcal{E}} \{l_{\mathcal{E}}(e)\}$ be the set of all the actions the attacker can execute in $\mathcal{G}$ after being in state v_i.

5.4.4.1 Labeled CMDP Type I

We start by presenting Algorithm 8, which allows us to construct the *CMDP Type I*. Let $\mathcal{J}_1 = \langle \mathcal{X}_1, \mathcal{A}_1, \mathcal{P}_1, L_1, \omega_1, c_1, \beta_1 \rangle$ refer to this type of CMDP. A state in $\mathcal{J}_1$ refers to a state of the attacker in the system. Our objective is to find the defense countermeasure or set of countermeasures that needs to be deployed at each state in order to minimize the risk of future actions of the attacker on the system. In Algorithm 8, at the beginning, for each state $v \in \mathcal{V}$ in the attack graph $\mathcal{G}$, we create a state in the CMDP $\mathcal{J}_1$ and associate it to v. Let $g : \mathcal{V} \to \mathcal{X}_1$ be a function where $g(v) = x$ refers to the state in $\mathcal{J}_1$ associated with the node $v \in \mathcal{V}$. Once we have created the states in $\mathcal{J}_1$, we need to associate actions to these states and define the probability transition function $\mathcal{P}_1$. If the attacker did not execute any action after being in a state $v_i \in \mathcal{V}$, the attack terminates. It is either that the attacker has achieved his objectives or he does not have the required set of access levels and knowledge items that allows him to execute any additional action in the system. Therefore, the state x_i in $\mathcal{J}_1$ associated with v_i is an absorbing state. We associate the action a_0 to x_i.

We are left with the set of states $x \in \mathcal{X}_1$ associated to states $v \in \mathcal{V}$ after which the attacker has executed at least one action. Given a set of available defense countermeasures $\mathcal{B}$ to the defender, we are interested in the set of countermeasures that can be used to defend against attacker actions in $\mathcal{L}_{\mathcal{E}}(v)$. Each combination of defense countermeasures is treated as an action by the defender. Therefore, all countermeasures combinations that can be deployed in x need to be generated to be available to the defender. During this generation process, we discard the combination in which deploying the corresponding set of countermeasures at the same time is inefficient, conflictory, or infeasible. For example, it may not be possible to deploy two countermeasures for two different attack actions at the same time on the same equipment in the system. In addition, two countermeasures can have the same efficiency against the same set of attack actions. In this case, it is inefficient for the defender to deploy these two countermeasures at the same time.

Algorithm 8

Input: AEM $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, Defense countermeasures $\mathcal{B}$
Result: CMDP Type I $\mathcal{J}$

function CONSTRUCTCMDPTYPEI($\mathcal{G}$, $\mathcal{B}$)
  for each $v \in \mathcal{V}$ **do**
    $x \leftarrow$ CREATESTATE
    $\mathcal{X} \leftarrow \mathcal{X} \cup \{x\}$
  end for
  for each $x \in \mathcal{X}$ **do**
    if $\nexists v' \in \mathcal{V}$ s.t. $e = (g^{-1}(x), v') \in \mathcal{E}$ **then**
      $\mathcal{A}(x) \leftarrow \mathcal{A}(x) \cup \{a_0\}$
      $\mathcal{P}(x, a_0, x) \leftarrow 1$
      $RR(x, a_0) \leftarrow 0$
    else
      for each $b \in \mathcal{B}$ **do**
        if b is a countermeasure for any $r \in \mathcal{L}_{\mathcal{E}}(g^{-1}(x))$ **then**
          $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{b\}$
        end if
      end for
      for each $z \in \mathcal{P}(\mathcal{Z})$ **do** ▷ $\mathcal{P}(\mathcal{Z})$: power set of $\mathcal{Z}$
        if z is valid **then**
          $\mathcal{A}(x) \leftarrow \mathcal{A}(x) \cup z$
        end if
      end for
      $\mathcal{A}(x) \leftarrow \mathcal{A}(x) \cup \{a_0\}$
      for each $a \in \mathcal{A}(x)$ **do**
        $RR(x, a) \leftarrow 0$
        for each $k \in \mathcal{L}_{\mathcal{E}}(g^{-1}(x))$ **do**
          $e = l_{\mathcal{E}}^{-1}(k)$, $e = (g^{-1}(x), v)$
          if a is a countermeasure for k **then**
            $\mathcal{P}(x, a, g(v)) \leftarrow P_c(k) \times P_s(k) \times (1 - \mathit{efficiency}(a))$
            $\mathcal{P}(x, a, x) \leftarrow 1 - P_c(k) \times P_s(k) \times (1 - \mathit{efficiency}(a))$
            $RR(x, a) \leftarrow RR(x, a) + (1 - \mathit{efficiency}(a)) \times P_c(k) \times P_s(k) \times \mathit{Impact}(k)$
          else
            $\mathcal{P}(x, a, g(v)) \leftarrow P_c(k) \times P_s(k)$
            $\mathcal{P}(x, a, x) \leftarrow 1 - P_c(k) \times P_s(k)$
            $RR(x, a) \leftarrow RR(x, a) + P_c(k) \times P_s(k) \times \mathit{Impact}(k)$
          end if
        end for
      end for
    end if
  end for
end function

At this stage, we are left with the need to define the probability transition function $\mathcal{P}_1$ to complete the definition of our *CMDP Type I*. As we have mentioned earlier, we consider that a countermeasure for a particular attack may have an *efficiency* measure that represents the probability that it will be able to thwart that attack successfully. An action a at a state $x \in \mathcal{X}_1$ could refer to the deployment of a set of countermeasures. a could affect the transition probabilities to some states while having no effect on others. For example, from a given attacker state x_i, the attacker can choose to exploit vulnerability γ or γ'. If an efficient patch to γ was applied, the attacker will not be able to exploit γ while maintaining the possibility to exploit γ'.

Let us suppose that the attacker can transition from a state x_i to a state x_j or x_m as a result of two different actions by the attacker. We assume that we have a countermeasure a to the attack action k associated with the edge $e = (g^{-1}(x_i), g^{-1}(x_j))$. If a was deployed, the system will transition to state x_j with a probability $P_c(k) \times P_s(k) \times (1 - \textit{efficiency}(a))$, where $P_c(k)$ and $P_s(k)$ refer to the probability of choosing action k and the probability of succeeding in the execution of attack action k respectively. These values depend on the profile of the attacker and his capabilities. The attack succeeds and the system transition to x_j only if the countermeasure a was not able of thwarting action k. If we suppose that the attacker can repeat his attempt if it fails, we stay in state x_i with probability $1 - P_c(k) \times P_s(k) \times (1 - \textit{efficiency}(a))$. Otherwise, we transition to an absorbing state and stay there (to simplify the presentation, we do not represent this scenario in Algorithm 8). Since the countermeasure is associated with attack action k, it has no effect on the transition probability to state x_m. Therefore, the system transition to state x_m if the attacker choose to execute the action associated with edge $(g^{-1}(x_i), g^{-1}(x_m))$ and the attack execution was successful.

When a countermeasure a is deployed in a given state x, the immediate residual risk $RR(x, a)$ on the system depends on the probability of choosing and successfully executing attack actions after being in state x and their impact on the system in the presence of the chosen countermeasure. Computing the impact of an attack action can be done using Equation 4.1 in Chapter 4. This impact could refer to the financial, economical, or human impact or any combination of the previous ones. Deciding which type of impacts to consider depends on the system operator. Nevertheless, it is possible to consider a different type of residual risk for each type of impact that is taken into account. For example, we can consider a residual risk associated with the financial dimension of an attack and a different residual risk associated with the environmental dimension. In this case, each type of residual risk can be treated as a cost associated with each countermeasure. The objective of the defender is to choose the optimal set of countermeasures to minimize the value of a residual risk or such that the value of each type of residual risks does not exceed a defined tolerated threshold. Finally, we note that defining the labeling function L_1 is straightforward. A state x is labeled critical if in the state $g^{-1}(x) \in \mathcal{V}$, the attacker's set of access levels and knowledge items allows him to compromise critical equipment or services in the system.

Example. Let us take an example of a simple attack graph $\mathcal{G}$ as in Fig. 5.1 and try to construct the associated *CMDP Type I*.

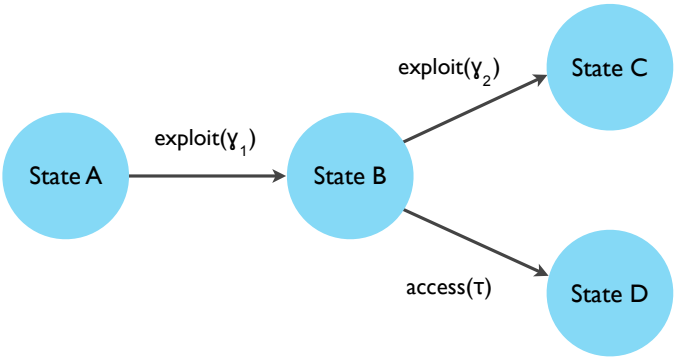


FIGURE 5.1: Example of an AEM

The attack graph $\mathcal{G}$ has four states: *State A*, *State B*, *State C*, and *State D*. Each state represents a state of the attacker in the system. If the attacker is in *State A* or *State B*, he can execute one or two of the following actions: exploit vulnerability γ_1, exploit vulnerability γ_2, and access equipment τ. After the execution of an action, the state of the attacker is updated with a new set of access levels and knowledge items.

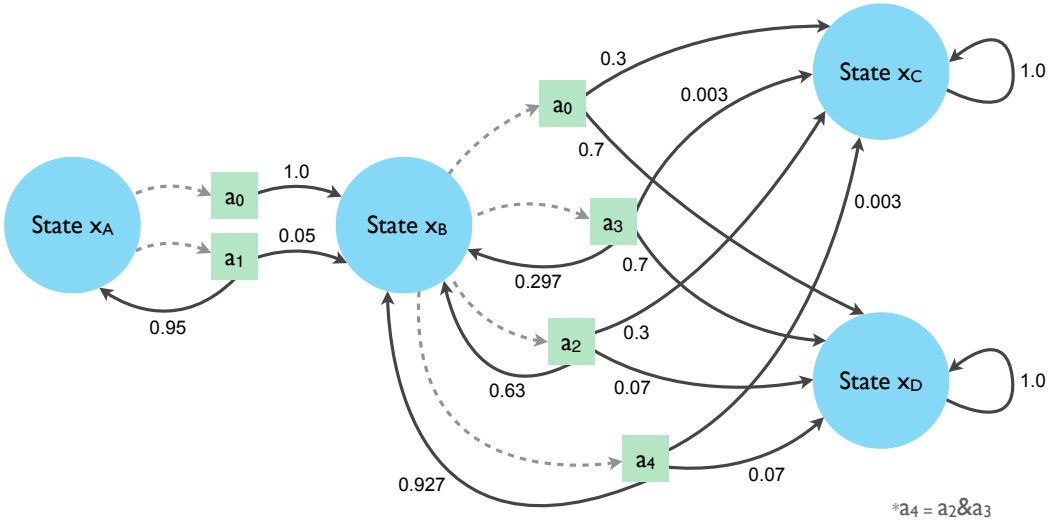


FIGURE 5.2: Example of a CMDP Type I

Let Fig. 5.2 depicts the *CMDP Type I* constructed using the information in the attack graph $\mathcal{G}$. The probability distribution over the initial state β_1 refers to the probability of being in any of these states at $t = 0$. Let us assume that initially, the attacker is in state x_A and will attempt to exploit vulnerability γ_1. Therefore, $\beta_1(state\ x_A) = 1$. We start the construction of the CMDP $\mathcal{J}_1$ by creating a state in $\mathcal{J}_1$ for each state in the attack graph $\mathcal{G}$. For example, for *State A*, we create the state x_A in $\mathcal{J}_1$.

In this example, there are two possible actions in the *CMDP Type I* associated to each action by the attacker. The first action a_0 refers to the fact that no countermeasure will be deployed. The second action refers to a defense countermeasure aiming at protecting the system against the attack attempt. Action a_1 is a countermeasure against exploiting γ_1. However, a_1 has a 95% efficiency. Therefore, an attacker attempting to exploit γ_1 has a 5% chance of succeeding. When an attempt to exploit γ_1 fails, we assume that the attacker will try again using a different method. If, however, only one attempt is possible, the arrow going from a_1 to the state *exploit* γ_1 will be directed to a state called *FAIL* designating the fact that the attacker was not able to achieve his objective. The state *FAIL* is an absorbing state. Once the attacker is in that state, he remains there. Action a_2 is a countermeasure against accessing equipment τ and has an efficiency of 90%. For example, a_2 may refer to an IDS with a detection rate of 90% deployed to detect access attempts to equipment τ. Action a_3 is a countermeasure against exploiting vulnerability γ_2 and has an efficiency of 99%. For example, a_3 may refer to the deployment of a patch to vulnerability γ_2. a_3 is very reliable in thwarting any exploit attempt but does not eliminate the probability that the attacker will leverage any weakness in the deployment of the patch to exploit γ_2 nevertheless. Finally, action a_4 refers to the deployment of countermeasures a_2 and a_3 at the same time. In this example, when the attacker executes an action and no countermeasure is deployed, we assume that he always succeeds ($P_s(k) = 1 \ \forall$ attack action k).

If the attacker succeeded in exploiting γ_1 when countermeasure a_1 was deployed, we will transition to state x_B. Then, we assume that depending on the attacker preferences, there are 70% chance that the attacker will attempt to access equipment τ and 30% chance that he will attempt to exploit γ_2. Let us take the case where the defender chooses to deploy countermeasure a_3 in state x_B. There are three possible outcomes for the attacker. If the attacker chooses to access equipment τ (with a 70% probability), he will always succeed since there is no deployed countermeasure for that action in state x_B. There is a 30% probability that the attacker chooses to exploit γ_2. Since countermeasure a_3 is deployed, the attacker can either succeed or fail in his attack attempt. The efficiency of a_3 is 99%. Therefore, the probability that the attacker fails in his attack attempt is 0.297 ($P_c(\textit{exploit } \gamma_2) \times \textit{efficiency}(a_3)$). Otherwise, he succeeds with a probability of 0.003. Now, if the defender decides to choose action a_4 referring to the deployment of countermeasures against exploiting vulnerability γ_2 and accessing equipment τ, any attack attempt will fail with a probability of 0.927 ($P_c(\textit{exploit } \gamma_2) \times \textit{efficiency}(a_3) + P_c(\textit{access } \tau) \times \textit{efficiency}(a_2)$). According to the attack graph in Fig. 5.1, when the attacker reaches *State C* or *State D*, he does attempt to execute any additional action. Therefore, states x_C and x_D are absorbing states.

We notice that when the number of actions that can be executed by the attacker after a given attacker state increases, the number of combinations of countermeasures that needs to be taken into account increases and thus the complexity of constructing the *CMDP Type I*.

5.4.4.2 Labeled CMDP Type II

In this section, we will be interested in generating the *CMDP Type II* based on the information present in the attack graph in Chapter 4. Let $\mathcal{J}_2 = \langle \mathcal{X}_2, \mathcal{A}_2, \mathcal{P}_2, L_2, \omega_2, c_2, \beta_2 \rangle$ refer to this type of CMDP. In $\mathcal{J}_2$, a state refers to an attempt to execute an action by the attacker given a set of acquired access levels and knowledge items. For example, in a given state, the attacker is attempting to exploit a vulnerability. For that state, we associate a set of defense countermeasures in order to prevent the attacker from executing that action. If the countermeasure chosen by the defender is efficient, the attack fails. Otherwise (i.e. the attack succeeds), we will transition with a certain probability to another state in the system in which the attacker will attempt to execute another action. In some scenarios, this type of CMDP offers a number of advantages to the *CMDP Type I*. We will discuss the implications of choosing each type of CMDP representations in the next section.

Let $h : \mathcal{E} \rightarrow \mathcal{X}_2$ be a function where $h(e) = x_{ij}$ refers to the state in $\mathcal{J}_2$ associated with the edge $e = (v_i, v_j)$ in $\mathcal{E}$. Let h^{-1} be the inverse function. We construct the *CMDP Type II* using Algorithm 9. We start by creating a state x_{ij} in the CMDP for each edge $e = (v_i, v_j) \in \mathcal{E}$. We create an absorbing state x_0 in the CMDP. x_0 refers to the fact that no additional attempt to execute any action will be carried out by the attacker. At state x_{ij}, the attacker is attempting to execute the action associated with the edge $e = (v_i, v_j) \in \mathcal{E}$. In order to defend the system against this threat, the defender has to deploy the required defense countermeasure while satisfying a set of constraints. For example, the condition that the defense budget can cover the cost of deploying that countermeasure. We notice that contrary to the attacker state in the *CMDP Type I* in which we associate defense countermeasures against future attack attempts that can be carried out from that state, we associate countermeasures to a state in the *CMDP Type II* against a defined attack attempt. In this case, we know which action the attacker will attempt to execute and we are trying to deploy the countermeasure that thwarts that threat.

In Algorithm 9, for each state x_{ij} that we created in $\mathcal{J}_2$, we identify the set of countermeasures against the attack action attempt associated to that state from the set of available countermeasures to the defender $\mathcal{B}$. If the attack in x_{ij} was the last attack attempt by the attacker, we will transition to the absorbing state x_0 if it succeeds. This transition takes place with a probability $(1 - \textit{efficiency}(a)) \times P_s(l_{\mathcal{E}}(h^{-1}(x_{ij})))$, where a refers to the deployed countermeasure in x_{ij} and $P_s(l_{\mathcal{E}}(h^{-1}(x_{ij})))$ refers to the probability that the attack attempt in x_{ij} succeeds. If further attack attempts are possible, the probability of transitioning to the next state x_{jk} in $\mathcal{J}_2$ depends on the efficiency of the deployed countermeasure, the success likelihood of the attack in x_{ij}, and the probability that the attacker will attempt to execute the action associated with x_{jk}. The residual risk on the system $RR(x_{ij}, a)$ after deploying a countermeasure a in x_{ij} is given by $(1 - \textit{efficiency}(a)) \times P_s(l_{\mathcal{E}}(h^{-1}(x_{ij}))) \times \textit{Impact}(l_{\mathcal{E}}(h^{-1}(x_{ij})))$, where $\textit{Impact}(l_{\mathcal{E}}(h^{-1}(x_{ij})))$ refers to the impact on the system after executing the attack action associated with x_{ij}. Similarly to the *CMDP Type I*, we add the action a_0 to each state in $\mathcal{J}_2$, where a_0 refers to the fact that no countermeasure was deployed in that state.

Algorithm 9

Input: AEM $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, Defense countermeasures $\mathcal{B}$
Result: CMDP Type II $\mathcal{J}$

function CONSTRUCTCMDPTYPEII($\mathcal{G}$, $\mathcal{B}$)
  for each $e = (v_i, v_j) \in \mathcal{E}$ **do**
    $x_{ij} \leftarrow$ CREATESTATE
    $\mathcal{X} \leftarrow \mathcal{X} \cup \{x_{ij}\}$
  end for
  $x_0 \leftarrow$ CREATESTATE ▷ x_0: sink state
  $\mathcal{X} \leftarrow \mathcal{X} \cup \{x_0\}$
  for each $x_{ij} \in \mathcal{X}$ **do**
    for each $a \in \mathcal{B}$ **do**
      if a is a countermeasure for $l_{\mathcal{E}}(h^{-1}(x_{ij}))$ **then**
        $\mathcal{A}(x_{ij}) \leftarrow \mathcal{A}(x_{ij}) \cup \{a\}$
        if $\nexists v_k \in \mathcal{V}$ s.t. $e = (v_j, v_k) \in \mathcal{E}$ **then**
          $\mathcal{P}(x_{ij}, a, x_0) \leftarrow (1 - \mathit{efficiency}(a)) \times P_s(l_{\mathcal{E}}(h^{-1}(x_{ij})))$
        else
          for each $e \in \mathcal{E}$ s.t. $e = (v_j, v_k)$ **do**
            $\mathcal{P}(x_{ij}, a, x_{jk}) \leftarrow P_c(l_{\mathcal{E}}(e)) \times (1 - \mathit{efficiency}(a)) \times P_s(l_{\mathcal{E}}(h^{-1}(x_{ij})))$
          end for
        end if
        $\mathcal{P}(x_{ij}, a, x_{ij}) \leftarrow 1 - P_s(l_{\mathcal{E}}(h^{-1}(x_{ij}))) \times (1 - \mathit{efficiency}(a))$
        $RR(x_{ij}, a) \leftarrow (1 - \mathit{efficiency}(a)) \times P_s(l_{\mathcal{E}}(h^{-1}(x_{ij}))) \times \mathit{Impact}(l_{\mathcal{E}}(h^{-1}(x_{ij})))$
      end if
    end for
    $\mathcal{A}(x_{ij}) \leftarrow \mathcal{A}(x_{ij}) \cup \{a_0\}$
    if $\nexists v_k \in \mathcal{V}$ s.t. $e = (v_j, v_k) \in \mathcal{E}$ **then**
      $\mathcal{P}(x_{ij}, a_0, x_0) \leftarrow P_s(l_{\mathcal{E}}(h^{-1}(x_{ij})))$
    else
      for each $e \in \mathcal{E}$ s.t. $e = (v_j, v_k)$ **do**
        $\mathcal{P}(x_{ij}, a_0, x_{jk}) \leftarrow P_c(l_{\mathcal{E}}(e)) \times P_s(l_{\mathcal{E}}(h^{-1}(x_{ij})))$
      end for
    end if
    $\mathcal{P}(x_{ij}, a_0, x_{ij}) \leftarrow 1 - P_s(l_{\mathcal{E}}(h^{-1}(x_{ij})))$
    $RR(x_{ij}, a_0) \leftarrow P_s(l_{\mathcal{E}}(h^{-1}(x_{ij}))) \times \mathit{Impact}(l_{\mathcal{E}}(h^{-1}(x_{ij})))$
  end for
  $\mathcal{A}(x_0) \leftarrow \mathcal{A}(x_0) \cup \{a_0\}$
  $\mathcal{P}(x_0, a_0, x_0) \leftarrow 1$
  $RR(x_0, a_0) \leftarrow 0$
end function

Example. To illustrate how we construct the CMDP from the AEM, we take the example in Fig. 5.1. As we have mentioned earlier, there are four states in the attack graph and the attacker executes three actions.

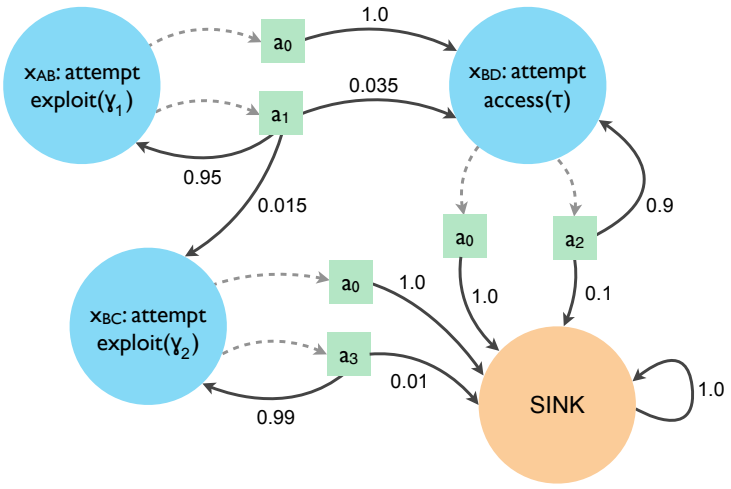


FIGURE 5.3: Example of a CMDP Type II

Fig. 5.3 depicts the corresponding *CMDP Type II*. The probability distribution over the initial state β_2 refers to the probability of being in any of the states in Fig. 5.3 at $t = 0$. We assume that initially, the attacker is attempting to exploit vulnerability γ_1. Therefore, $\beta_2(x_{AB}) = 1$. We notice that we have an absorbing state called $SINK$. We have referred to this state earlier as x_0. When no other attack action is possible or the attacker has achieved his objectives, we will transition to the *SINK* state and remain there.

In this example, there are two possible actions at each state. Action a_0 refers to the fact that no countermeasure will be deployed. In this case, we notice that the transition to the next state after the execution of the action of the attacker takes place with probability 1. Action a_1 is a countermeasure against exploiting γ_1. However, a_1 has a 95% efficiency. Therefore, an attacker attempting to exploit γ_1 has a 5% chance of succeeding. When an attempt to exploit γ_1 fails, we assume that the attacker will try again using a different method. If, however, only one attempt is possible, the arrow going from a_1 to the state *exploit* γ_1 will be directed to a state called *FAIL* designating the fact that the attacker was not able to achieve his objective. Now, if the attacker succeeded in exploiting γ_1 when countermeasure a_1 was deployed, there are 70% probability that he will attempt to access equipment τ and 30% probability that he will attempt to exploit γ_2. These probabilities depend on the profile of the attacker and his objectives. Assuming that the probability of successfully executing an action by the attacker when no countermeasure is deployed is 1 ($P_s(l_{\mathcal{E}}(e)) = 1 \; \forall e \in \mathcal{E}$), when a_1 is deployed, we transition to x_{BD} and x_{BC} with probabilities 0.035 (0.05×0.7) and 0.015 (0.05×0.3) respectively. Similar analysis can be conducted for defining the transition probability function when deploying countermeasures in states x_{BD} and x_{BC}.

5.4.4.3 Semantic of Costs

In order to find the optimal security policy, we solve the constrained optimization problem in Equation 5.3. The definition of the CMDP is generic with respect to the semantic of the immediate cost ω and the cost c related to the constraints in the model. For example, in some cases, the operator is interested in generating a strategy that minimizes the cost of deploying the set of defense countermeasures. In this case, $\omega(x, a)$ will represent the cost of deploying countermeasure a in state x. In other cases, the operator may be interested in minimizing the overall residual security risk of attacks on the system after the deployment of a set of defense countermeasures given constraints on the defense budget. Therefore, $\omega(x, a)$ will refer to the residual security risk that can be a function of the economical, environmental, or human impact, or a combination of the above impacts, and the constraint in the system will refer to the defense budget for countermeasures deployment.

As we have mentioned earlier, a defense countermeasure may not be totally efficient in thwarting an attack. Therefore, there is a probability that the attacker will be able to succeed despite the deployment of that countermeasure. Defining the semantic of the set of costs c^k associated to the M constraints can be done similarly to the immediate cost ω. However, for a given problem, any two elements in the set $\{\omega, c^1, ..., c^M\}$ cannot have the same semantic (e.g. the immediate cost ω and any cost c^k associated to the k^{th} constraint cannot both refer to the cost of deploying a countermeasure at the same time). In addition, as we have presented in Section 5.4.2, the system operator may want to have some guarantees that the probability of compromising a critical equipment or service falls below a defined threshold. This threshold depends on the type and criticality of the equipment or service that needs to be protected. In this case, reachability constraints are associated to costs c^k and are defined as in Section 5.4.2.

5.4.4.4 Discussion

The main difference between the two types of CMDPs that we have presented lies in the semantic of the states in each model, which will have important implications on the result of the security policy optimization. In *CMDP Type I*, a state x refers to an attacker state (i.e. acquired access levels and knowledge items). The transition between the different states in the model depends on the action executed by the attacker and the countermeasure or set of countermeasures deployed by the defender. In that respect, the objective of the defender is to choose the best action at that state in order to protect the system while anticipating the action that will eventually be executed by the attacker. However, in the *CMDP Type II*, having acquired a set of knowledge items and access levels on equipment in a state x, the attacker is attempting to execute a particular action in that state. Therefore, the objective of the defender in this case is to choose the best countermeasure for the attack action in state x. To resume, in the *CMDP Type I*, the defender is trying to answer the question "*At a given moment, the attacker was able to gain certain access to my network, how do I protect the system while anticipating all future attack attempts?*", and in *CMDP Type II*, he

is trying to answer the question "*At a given moment, I know that the attacker is trying or will attempt to execute that action, what should I do?*".

As we have seen in Section 5.4.4, we take into account all valid combinations of countermeasures that can be deployed in a given state in the *CMDP Type I*. This type of CMDP offers an important advantage as we can compute the optimal security policy taking into account the available number of defenders in the system. In particular, the set of security countermeasures that can be deployed at the same time in a given state in the *CMDP Type I* will depend on the available number of defenders. The *CMDP Type I* can be quickly constructed based on the information in the attack graph to take into account changes in the number of defense personnel. Therefore, only the set of countermeasures that can be deployed by the available defenders at a particular state in the system are taken into account.

5.4.5 Optimal Defense Recommendations

In this section, we present and discuss the different interpretations and usage scenarios of the result of the optimization problem in Equation 5.8 for a given CMDP. First, we present the direct interpretation of the result of the optimization problem, which is an optimal security policy to assist the system operator with responding to intrusions. In addition, based on the optimal policy u, we show, for a given state, under which conditions the defender will choose a countermeasure in that state deterministically. Second, we present how the stochastic optimal policy u can offer a useful insight for prioritizing countermeasures deployment. Finally, we discuss how our model can be used to compare the relative security of two architectures.

5.4.5.1 Optimal Response to Intrusions

The solution to the optimization problem in Equation 5.8, if it exists, is a stochastic policy. This policy can be transformed to a stationary policy as in Equation 5.9 in order to be a solution to the original optimization problem in Equation 5.3. Let b refer to this stationary policy, where $b_x(a)$ refers to the probability of choosing action a at state $x \in \mathcal{X}$, $\forall a \in \mathcal{A}(x)$. When it exists, this policy guarantees the security objectives defined by the operator. Therefore, the model can be viewed as a decision support system assisting the system operator with responding to intrusions efficiently. In particular, knowing the current attacker state, the optimal policy defines the best security countermeasure that the defender needs to deploy to thwart the attack and minimize the risk on the system.

In general, the stationary policy b can be a stochastic policy. Being in a given state x, the defender may prefer, if it is possible, to choose an optimal action at that state deterministically. However, this choice comes with a tradeoff with respect to the cost of deploying that countermeasure and the subsequent impact on the system compared to the stationary policy b. We present Algorithm 10 to find such solution.

Algorithm 10

Input: CMDP $\mathcal{J}$, state $x \in \mathcal{X}$
Result: Action a at state $x \in \mathcal{X}$

function findDeterministicAction($\mathcal{J}$, x)
  $u^s \leftarrow$ findOptimalPolicy($\mathcal{J}$) ▷ u^s: optimal policy
  $b^s \leftarrow$ getStationaryPolicy(u^s)
  for each action $a \in \mathcal{A}(x)$ s.t. $b^s_x(a) \geq T_b$ **do**
    $\mathcal{J}^d$ = copyOf($\mathcal{J}$) except that only $a \in \mathcal{X}^d$
    $u^d \leftarrow$ findOptimalPolicy($\mathcal{J}^d$)
    if $u^d \neq$ *null* **then**
      if $|\Omega_\alpha(\beta, u^d) - \Omega_\alpha(\beta, u^s)| \leq T_\Omega$ & $\|C_\alpha(\beta, u^d) - C_\alpha(\beta, u^s)\| \leq T_C$ **then**
        $A^d(x) \leftarrow A^d(x) \cup \{a\}$
      end if
    end if
  end for
  if $A^d(x) \neq \varnothing$ **then**
    $u \leftarrow \underset{u^d \in \mathcal{U}(A^d(x))}{\operatorname{argmin}} \lambda_\Omega \Omega_\alpha(\beta, u^d) + \lambda_C \|C_\alpha(\beta, u^d)\|$
    $a \leftarrow u(x)$
  end if
end function

In Algorithm 10, we assume that a policy u^s to the optimization problem in Equation 5.8 exists. Let b^s refer to the stationary policy associated to u^s. For each action a in state x where the probability of choosing a under b^s is greater than a defined threshold $T_b \geq 0$, we compute an optimal policy u^d to a modified version of the original CMDP $\mathcal{J}$ that we refer to as $\mathcal{J}^d$. $\mathcal{J}^d$ is identical to $\mathcal{J}$ except for the fact that only action a is possible in state x. Let $A^d(x)$ be the set of actions $a \in \mathcal{A}(x)$ when an optimal solution u^d exists and the absolute value of the difference between the discounted costs $\Omega_\alpha(\beta, u^d)$ and $\Omega_\alpha(\beta, u^s)$, and the norm (defined by the defender) of the difference between the vector of discounted costs associated to the M constraints $C_\alpha(\beta, u^d)$ and $C_\alpha(\beta, u^s)$ are within defined tolerated thresholds $T_\Omega \geq 0$ and $T_C \geq 0$ respectively. Let $\mathcal{U}(A^d(x))$ refer to this set of policies u^d. If $A^d(x) = \varnothing$, the decision of which action to choose in state x will be based on the stationary policy b^s. Otherwise, if $A^d(x) \neq \varnothing$, the solution to the problem is the action $a \in A^d(x)$ associated to the policy u s.t. $u = \underset{u^d \in \mathcal{U}(A^d(x))}{\operatorname{argmin}} \lambda_\Omega \Omega_\alpha(\beta, u^d) + \lambda_C \|C_\alpha(\beta, u^d)\|$, where $\lambda_\Omega + \lambda_C = 1$ and $\lambda_\Omega, \lambda_C \geq 0$. Therefore, we choose the action that minimizes the weighted sum of the discounted cost $\Omega_\alpha(\beta, u^d)$ and the norm of the vector of discounted costs associated to the M constraints $\|C_\alpha(\beta, u^d)\|$.

5.4.5.2 Countermeasures Ranking

The solution to the optimization problem stated in Equation 5.8, if it exists, will give us a probability distribution over the occupation measure ρ. In what follows, we refer to a

countermeasure a^i as a countermeasure deployed in a specific location i in the system. For example, even though we can have the same type of countermeasures deployed on equipment τ_1 and τ_2, we refer to the countermeasure deployed on each of these equipment as a^1 and a^2 respectively. For each countermeasure a^i, let $\mathcal{X}(a^i)$ refer to the states in $\mathcal{X}$ in which a^i can be deployed and let $\rho(a^i) = \frac{\sum_{x \in \mathcal{X}(a^i)} \rho(x, a^i)}{1 - \sum_{x \in \mathcal{X}} \rho(x, a_0)}$. Therefore, we have $\sum_{a^i \in \mathcal{A}, a \neq a_0} \rho(a^i) = 1$. The probability distribution over the set of available countermeasures can serve as a ranking system for the deployment of the countermeasures in the system. For example, the countermeasure a^i that has a highest ranking (i.e. highest $\rho(a^i)$) can be viewed as the most urgent to deploy. This ranking is important when anticipating a threat to the system where we have a limited number of defenders. In this case, countermeasures prioritization becomes essential to offer the best protection to the system.

5.4.5.3 Comparing the Relative Security of Two Architectures

It is worth mentioning that when we want to compare two architectures or security configurations of a control system, the attack graph associated to each of these architectures or configurations generated in Chapter 4 and the optimal security policy of the associated CMDPs offer an important insight to evaluate their relative security. For example, the minimum attack budget, attack time, and number of attack actions required to compromise a critical equipment or service in each architecture can serve as basic criteria for comparison. In addition, the security of an architecture can also be evaluated through the minimum cost that is needed to deploy defense countermeasures in order to remain within the tolerated threshold for the residual risk of attacks on the system. On the other hand, given a defense budget, it is possible to compute the minimum residual risk of attacks after the optimal deployment of security countermeasures. In this case, the best result is achieved by the architecture or configuration that most minimizes the residual risk of attacks using the available defense resources.

5.5 Conclusion

An attack graph offers important insights for the assessment of the security of a targeted system. However, the large number of nodes and the complexity of their interconnections render the manual analysis of such graphs challenging. In this chapter, we addressed this challenge by presenting an approach to compute optimal security policies automatically using information in the attack graph. Based on the framework of Constrained Markov Decision Processes (CMDPs), we compute an optimal security policy that satisfies the defender's objective and a set of constraints. In Appendix C, we show how to compute the initial probability distribution over the initial state in the CMDP when facing the threat of an external attacker targeting the system. In our approach based on CMDPs, we answer the following questions: "*Knowing the location and the capability of an attacker, what should a defender do now to reduce the risk of future attack attempts on his system?*", and "*Knowing*

what will be the next move of the attacker, how the defender must react to protect his critical assets?".

The solution of the CMDP problem can be used in multiple ways. First, the optimal security policy can be used as a decision-making support system to assist the defender in responding to intrusions efficiently. Second, the solution can be used to prioritize the deployment of security countermeasures in the system before any attack attempt takes place. Finally, our approach, combined with information in the attack graph generated in Chapter 4, can be used to compare the relative security of two architectures or security configurations. In the next chapter, we validate our approach on an AMI case study.

Chapter 6

Case Study

In this chapter, we validate our approach based on Constrained Markov Decision Processes (CMDPs) introduced in Chapter 5 on a case study.

6.1 Introduction

The Advanced Metering Infrastructure (AMI) is an important component of the smart grid. A smart meter, an intelligent electronic device installed at the customer's premises, is responsible of sending power consumption and executing control commands received from the utility company. The communication infrastructure that enables such services is very important. An attack on a critical equipment in this infrastructure can have undesirable effects on the power grid. In this chapter, we are interested in identifying the set of countermeasures that needs to be deployed on equipment to protect the system from a given attacker. In particular, we investigate the case where the attacker has access to certain networks in the system. The security policy must take into account the initial access of the attacker and satisfy a set of defined objectives.

6.2 System Architecture

We consider an architecture of the AMI as in Fig. 6.1 (inspired from [Nat14b, Nat14a, Nat10]). In this architecture, we have four main layers. In the first layer, we find the smart meters. In the AMI, smart meters offer a number of services to both the electric utility company and the customers. As we have mentioned earlier, one of the smart meters main function is sending the customer power consumption to the utility regularly. This information is used to bill the user on the power consumed and to monitor the power demand in the electric grid. An additional feature of the smart meter is enabling demand response. In this case, the customer gives the utility the right to control his power consumption by

controlling some of his home appliances. For example, the utility can switch the dishwasher on when the energy demand in the grid is low and electricity is cheap. However, one of the most critical functions of the smart meter is executing connect/disconnect command issued by the utility. Given its possible impact on the stability of the electric grid in one hand, and the power supply to the customer in another, this type of commands needs to be issued only by the utility company and protected from any attack attempt. Smart meters communicate with the utility through a set of Data Concentrator Units (DCUs). A DCU, situated in the second layer in our architecture, is a device responsible of collecting data from multiple meters and forwarding it to the utility systems for analysis.

6.2.1 AMI Head-End System

The meter data from DCUs is sent first to the utility AMI head-end system. Situated in the third layer in our architecture, the AMI head-end system is responsible of managing the communication between the utility company and the smart meters. The AMI head-end system requests data and events from smart meters, reports real time meter measurements back to the utility, and can initiate remote connect and disconnect commands to meters. The connection between the AMI head-end and DCUs pass through a firewall. This is important to prevent any unauthorized communication between an equipment in the Wide Are Network (WAN) and another equipment in the AMI head-end system. We consider three main equipment in the AMI head-end: a communication server, an application server, and a database server. The communication server is responsible of managing the communications with the smart meters. The application server performs initial aggregations and analysis of smart meters data received from the communication server before sending it to the Meter Data Management System (MDMS) and the SCADA control center. Meter and aggregated data are also stored locally in the AMI head-end in a database. Finally, an operator workstation is connected to the AMI head-end network for system management.

The aggregated data sent from the AMI head-end system is used, among other things, to monitor the real-time power consumption in the grid, detect outages, and charge customers for the power consumed. These functions are provided by a set of equipment in two different networks, namely the SCADA control center and the enterprise network, which are situated in the fourth layer in our architecture. For additional security and network segmentation, a firewall is added to each AMI head-end, SCADA control center, and enterprise networks to manage the communications between equipment inside these networks and equipment in the outside. These networks can be located in different physical locations and therefore, use the WAN as the communication infrastructure. For that reason, we added the firewalls at the edge of each of the networks. Security administrators manage the firewalls and we assume in our case that they are sufficiently secure such that the access control rules cannot be modified.

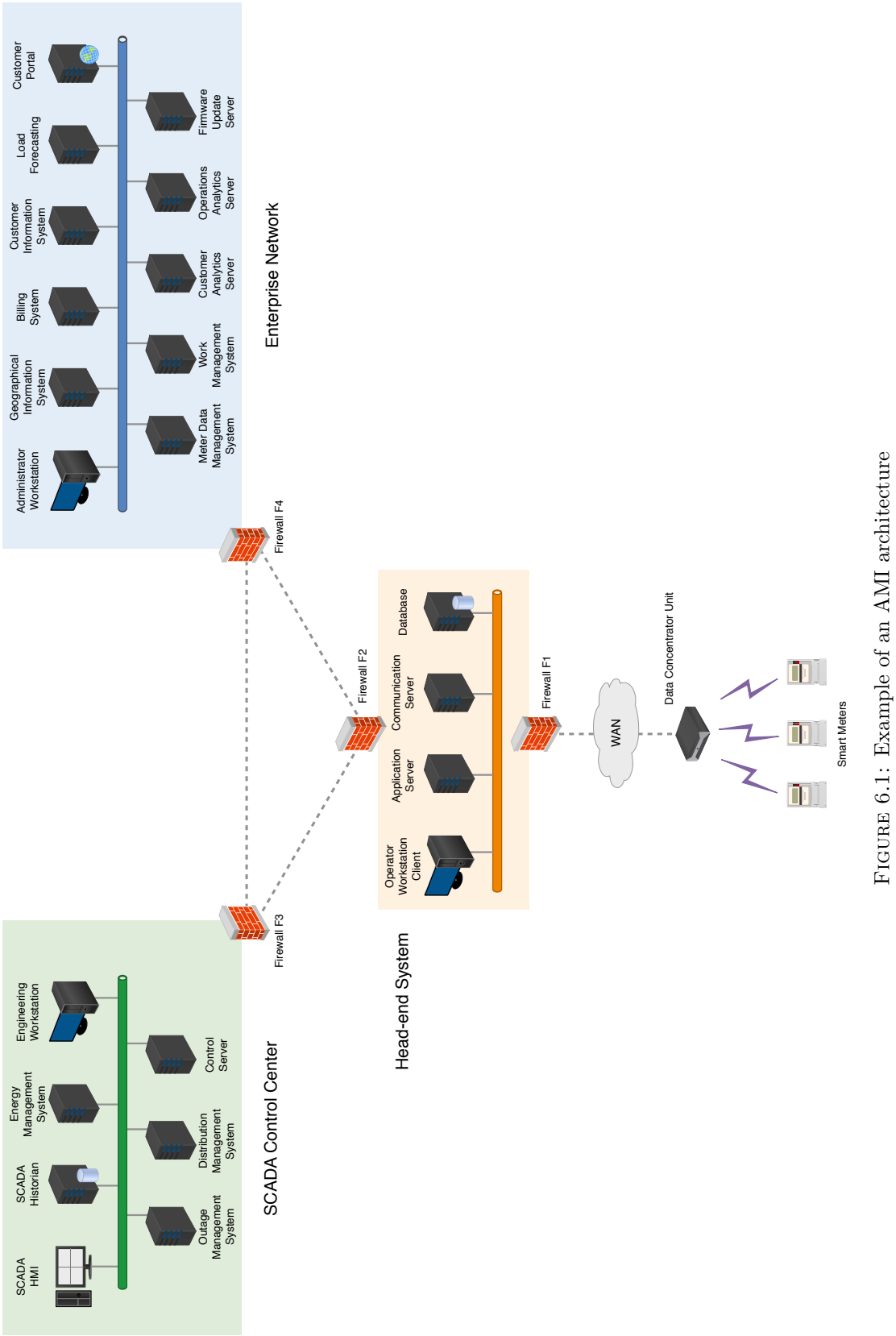


FIGURE 6.1: Example of an AMI architecture

6.2.2 Enterprise Network

In the enterprise network, the data received from smart meters is used in different applications. However, the received data needs to be further aggregated and analyzed before being available to these applications. The Meter Data Management System (MDMS) achieves this function. The MDMS receives meter data from the AMI head-end system. The data is aggregated and its accuracy and completeness is validated through the validation, estimation, and editing (VEE) process. Through this process, data tampering and energy theft can be detected.

Once smart meters data is aggregated and validated in the MDMS, it becomes available to different applications in the enterprise network. For example, the Billing system receives meter data from the MDMS to charge the user for the power consumed. The eventual price the customer needs to pay depends on his energy services subscriptions, and therefore his billing profile. The power consumed and the customer energy bill is then sent to the Customer Information System (CIS). The CIS is a system used to store customer information and manage the relationship between the utility company and the customers. For example, a customer who wants to have online access to his latest energy consumptions and electric bills can connect to the Customer Portal server. This server then interacts with the CIS to request information about that particular customer.

The Geographical Information System (GIS) is an asset management system of the electric grid. It stores information about network components, their interconnections, and maps the location of overhead and underground circuits. Therefore, the confidentiality of the information in the GIS must be guaranteed. An access to this type of information can provide an attacker with some knowledge to identify vulnerable links in the power grid. In order to maintain the power grid infrastructure, the work management system receives information about maintenance schedules and outage information and provides them to work crews. Using this information, field workers can be dispatched to affected areas quickly and efficiently. In the enterprise network, we also find the Load Forecasting System (LFS). Load forecasting is an important component of the smart grid. A forecasting model, which can use various sources of data such as the real-time power consumption, maintenance schedules, and weather forecasts, would allow the utility to take into account the effect of the integration of renewable energy resources in order to strategically manage electric power generation.

In the enterprise network as well, we find the customer and operations analytics servers. The customer analytics server uses the meter data received from the MDMS to produce customers' power demand profiles and forecast demand response for individual customers. The engineering and operation analytics server uses the data from the MDMS, among other sources, to measure the system performance and analyze operational effectiveness.

Finally, the Firmware Update server in the enterprise network is responsible of issuing patches and updates to the smart meters firmware. Therefore, the integrity of these updates must also be guaranteed. Similarly to the AMI head-end system, an administrator workstation is connected to the network to manage the equipment in the enterprise network.

6.2.3 SCADA Control Center

The power grid operator monitors and controls the electric system from the SCADA control center. The operator tracks the state of the power system through a set of sensors. Upon analyzing the data received from the sensors, appropriate control actions are taken. For example, the operator can send commands to route electricity through certain electric buses. In our case study, we consider six main components in the SCADA control center. The SCADA control server manages the communication between a set of equipment in the control center and the different sensors and equipment in the power grid. We assume that the control server is responsible of receiving the collected data from sensors and sending control commands issued by the operator. It formats data in order to be sent through communication channels and to be interpreted when received by control equipment and vice versa. In this sense, we consider that the control server acts as an interface between equipment in the SCADA control center and control equipment in the power grid.

The SCADA HMI is a human-machine interface that provides a graphics-based visualization of the controlled area of the power system. Using the HMI, the operator can also send control commands to equipment in the power grid under the control of the SCADA system. All the power state events that are received are stored in the SCADA historian. From this database, data can be retrieved to analyze the origins of power failures or to perform statistical inference of the behavior of the system.

The Energy Management System (EMS) is used to monitor, control, and optimize the performance of the generation and transmission systems in the power grid [Nat14b]. The EMS enhances the reliability of the grid and optimizes the use of the transmission network. To provide these services, the EMS uses real-time data sent from the different sensors in the power grid. The Distribution Management System (DMS) monitors and controls the distribution network. It provides a number of services such as contingency analysis, short-circuit analysis, switching schedule, and safety management. The Outage Management System (OMS) is an outage detection system that combines information collected from power grid sensors and outage calls from customers to predict the location of power outages. By analyzing the pattern of detected outages, the extent of the outages can be assessed and the number of affected customers can be estimated. Finally, an engineering workstation connected to the SCADA control center network is used to manage the different equipment in the control center.

6.3 Service Layer

In Chapter 4, we presented a model for generating attack graphs that takes into account the impact of attacker actions on the services running in the system. Assessing the impact of attacks on the service layer is an important step in order to better identify the critical components in the system that need to be hardened.

In our case study, to simplify the analysis, we made a set of assumptions to restrict the number of services that we represent in the model. In particular, only important services provided by the equipment are taken into account. The dependencies between 138 services, representing the restricted set of services, is identified. In this case study, we assume that compromising a given service results in a financial impact. Therefore, the environmental and human impact dimensions are not taken into account. In this section, for presentation reasons, we will only present a subset of the services that we have modeled in our case study. Fig. 6.2 depicts the interactions between the main equipment in our case study to provide the different services in the system.

FIGURE 6.2: Interactions between equipment to provide the services in the case study

One of the most important services represented in the case study is the possibility to send connect and disconnect commands to smart meters. The stability of the power grid could be affected if a large number of smart meters are disconnected at the same time without any prior planning. We assume that this type of commands can be issued only by two components in the system: the CIS and the AMI head-end application server. The CIS can issue a command to disconnect the power to a customer if he failed to respect his contract with the utility company. For operational and maintenance reasons, the AMI head-end application server can also issue this command. Both equipment rely on the AMI head-end communication server and the corresponding DCUs to relay the command to smart meters. Therefore, compromising the communication server will prevent the execution of the connect/disconnect command by preventing it from reaching the targeted smart meter.

In the enterprise network, pushing an update or a patch to the current smart meter firmware is a service provided by the Firmware update server. Similarly to the connect/disconnect command, the impact of compromising an updated version of a smart meter firmware could lead to undesirable effects. A malicious version of the firmware can cause permanent damage to the smart meters or be programmed to delay the restoration of the power supply to customers following a blackout. Similarly to the connect/disconnect command, the updated version of the firmware relies on the AMI head-end communication server and the DCU to send the update to smart meters to be installed.

The GIS and CIS systems provide information about the power grid assets and the customers respectively. Other services in the system such as load forecasting and outage management, among others, depend on one or both of these systems. For example, in the absence of information about the energy services that the customer is currently subscribed to, forecasting the demand response plan for that customer becomes challenging. Similarly for mapping outages to geographical locations, with the absence of information from these systems, it becomes challenging to identify the impacted customers and dispatch work crews efficiently to the affected areas. Other services in the enterprise network include data aggregation, validation, estimation, and editing provided by the MDMS.

In the SCADA control center, while the possibility to issue control commands to electric equipment remotely enhances the reliability of the power grid, it has the potential to increase the impact of cyberattacks on the power system. Therefore, this service provided by equipment in the control center must be protected. In addition, modifying the information displayed to the operator in the SCADA HMI could lead him to issue control commands that can impact the stability of the power system. Since the management of the communications between equipment in the SCADA control center and equipment in the power grid relies on the SCADA control server, this function makes it a critical component in the SCADA control center.

6.4 System Security Assumptions

In this section, we highlight the main assumptions that we make in the case study regarding the access control policy, the number and nature of vulnerabilities that exist in the system, and the available set of security countermeasures that can be deployed to protect the system.

6.4.1 Access Control Policy

In our case study, we assume that connections to the different networks are controlled by the edge firewalls. In addition, we assume that only the security administrators are eligible to change the access control rules on these firewalls. Changing the access control rules requires having a special access level on the firewalls and possessing certain credentials to carry out this task.

On each equipment in the three main networks in the case study, with the exception of the operator and administrator workstations, we assume the existence of three accounts: two user accounts and an administration account. Getting access to each of these accounts depends on the role of the user. In general, user accounts are associated with access levels on equipment. For example, a user with the role *viewer* can have access to the displayed data on the SCADA HMI but does not have the required privileges to send control commands to equipment in the power grid. In addition, accessing the services on each equipment depends on the privileges granted to each account. In particular, we assume the existence of an administration account that grants its holder the required privileges to manage the services running on equipment. Using this privileged account on an equipment τ, we assume that an attacker can compromise all the services running on τ.

6.4.2 Vulnerabilities

With respect to vulnerabilities, we assume a worst-case scenario in which each equipment in the network is vulnerable. With the exception of firewalls, we assume the existence of two vulnerabilities on each equipment in the AMI head-end, SCADA control center, and enterprise networks. The first vulnerability is a remotely exploitable vulnerability that grants an attacker the least privileged access level on the targeted equipment. Using the new granted access level, and if the attacker possessed the required skill level and had the required knowledge items, exploiting the second vulnerability on that equipment locally increases his privilege level on that equipment. With respect to firewalls, only a remotely exploitable vulnerability exists that grants the attacker a minimum privilege level on the targeted firewall. This access level does not give the attacker the required privileges to change the access control rules on the firewall. However, equipped with the privileges granted by this access level, the attacker can try to connect or compromise other equipment accessible from the firewall.

6.5 Available Security Countermeasures

In our case study, we assume that there exist two types of security countermeasures that can be applied on vulnerable equipment: patching vulnerabilities and installing a host-based intrusion detection system. We note that we associate two different actions for patching the remotely exploitable vulnerability (REV) and patching the locally exploitable vulnerability (LEV) on equipment. The first type of countermeasures consisting of patching a vulnerability has an efficiency of 99%. We assume that there is always a 1% risk that the patch does not work correctly or the attacker managed to exploit a weakness or a vulnerability introduced by applying the patch on the vulnerable equipment. The cost of patching a vulnerability on an equipment depends on multiple factors including its deployment cost and the criticality of the services provided by that equipment. In fact, a testing and a thorough validation process precede any patch attempt to ensure that the safety and availability of the services on equipment are not impacted by the application of the patch. Given the type of equipment and the criticality of the provided services, we assume in our case study that the cost associated with the application of a patch to a vulnerability on an equipment in the enterprise network is less than the cost of applying a patch to a vulnerability on an equipment in the AMI head-end system. In addition, the cost of applying a patch to a vulnerability on an equipment in the SCADA control center is greater that the cost of applying a patch to a vulnerability on an equipment in the AMI head-end system. In each network, the cost of applying a patch on the operator or administrator workstation is assumed to be considerably less than the cost of applying patches on the other equipment in the same network.

The second type of countermeasures consists of installing a host-based intrusion detection system (HIDS) on vulnerable equipment. We assume that this type of HIDSs is capable of detecting vulnerabilities exploitation attempts and suspicious access attempts on equipment. The detection rate of the HIDS is assumed to be 80%. The cost of installing the HIDS depends on the type of the vulnerable equipment. Similarly for the reasons laid out for the cost of applying patches on equipment, the cost of deploying a HIDS on an equipment in the SCADA control center is assumed to be greater than the cost of deploying a HIDS on an equipment in the AMI head-end system. In addition, the cost of deploying a HIDS on an equipment in the enterprise network is less than the cost of deploying a HIDS on an equipment in the AMI head-end system. We note that with respect to the operator and administrator workstations, the cost of deploying a HIDS is greater that patching a vulnerability on these equipment. This is particularly related to the cost of purchasing the license needed to install the HIDS.

6.6 Numerical Analysis

In this section, we are interested on hardening the security on vulnerable equipment. We simulate different attack scenarios on our system and analyze the results. In particular, we

are interested in identifying the set of countermeasures that needs to be applied to protect the system and in prioritizing their deployment. For that purpose, we restrict the numerical analysis on solving the *CMDP Type II* constructed using the information present in the attack graph generated in each attack scenario.

In this section, we assume a worst-case scenario for the profile of the attacker. Our objective is to protect the system from an attacker that has the required skills to exploit all the types of vulnerabilities that exist on our equipment. In addition, if the first attempt to execute an attack action in the system fails when the corresponding countermeasure to that action is deployed, we assume that the attacker will try to attack again.

6.6.1 Targeting the Enterprise Network

In this section, we consider an attacker who was able to gain access to the enterprise network. We are interested in defining a ranking to the deployment of the optimal set of security countermeasures that guarantees that the operator security objectives are satisfied. In addition, we are interested in evaluating the effect of the choice of the security objective on the optimal set of countermeasures that will eventually be deployed. To that end, we consider the following scenarios.

6.6.1.1 Scenario 1

In this scenario, the objective of the defender is to find an optimal policy in order to minimize the cost of deploying the security countermeasures on the different equipment. Using the equations in Section 5.4.5, we interpret the solution of the CMDP as a ranking system to prioritize countermeasures deployment.

The security objective defined in this scenario is simple and it is used to serve as a baseline for comparison. Therefore, the result is predictable. We note that the result is highly dependent on the different attack paths that an attacker can take to compromise equipment and services in the system. The highest ranking countermeasures are depicted in Table 6.1 (the result is presented in percentages). From the result, we notice that the deployment of a HIDS on most equipment is privileged to the application of patches to the different vulnerabilities that exist on these equipment. The two exceptions are the administrator workstation and customer analytics server where the cost of patching the remotely exploitable vulnerability is assumed to be less than the cost of deploying a HIDS.

6.6.1.2 Scenario 2

In this scenario, the objective of the defender is to find an optimal policy in order to minimize the cost of deploying the security countermeasures on the different equipment while

	Countermeasure	Scenario 1	Scenario 2
Firmware Update	Patch REV	0	10.738
	Deploy a HIDS	6.296	0.21
CIS	Patch REV	0	10.268
	Deploy a HIDS	6.297	0.367
MDMS	Patch REV	0	9.691
	Deploy a HIDS	6.295	0.572
Billing System	Patch REV	0	5.209
	Deploy a HIDS	6.297	2.155
GIS	Patch REV	0	0
	Deploy a HIDS	6.306	4.104
Customer Analytics	Patch REV	15.856	10.855
Operations Analytics	Patch REV	0	9.396
	Deploy a HIDS	6.297	0.687
Load Forecasting	Patch REV	0	9.396
	Deploy a HIDS	6.296	0.648
Work Management	Patch REV	0	0
	Deploy a HIDS	6.296	4.089
Customer Portal	Patch REV	0	0
	Deploy a HIDS	3.084	3.036
Firewall F4	Patch REV	0	0
	Deploy a HIDS	6.391	4.131
Admin Workstation	Patch REV	18.133	11.287

TABLE 6.1: Countermeasures ranking when an attacker has access to the enterprise network

setting a constraint on the tolerated residual risk on the system after the countermeasures are deployed. We take a value of 10000 for this constraint. Table 6.1 depicts the result.

In this scenario, we notice that the priority is now given to applying patches to vulnerabilities on 8 equipment instead of deploying a HIDS. This is in contrast with the result in Scenario 1. In order to satisfy the constraint with respect to the residual security risk, the defender must choose to patch the REV which has an efficiency of 99% instead of deploying a HIDS which has a detection rate of 80%. A compromise is therefore made in order to minimize the deployment cost of the countermeasures while remaining within the tolerated threshold for the residual risk. From Table 6.1, similarly to Scenario 1, applying the patch to the vulnerability on the administrator workstation has the highest priority. We also notice that the ranking for deploying countermeasures on other equipment has changed with respect to Scenario 1. For example, even though the MDMS and the billing system had similar ranking for deploying HIDSs in Scenario 1, the priority in Scenario 2 is given to

patching the vulnerability in the MDMS.

6.6.2 Targeting the AMI Head-end System

In this section, we consider an attacker who was able to gain access to the AMI head-end system. Similar to the previous section, we are interested in defining a ranking to the deployment of the optimal set of security countermeasures that guarantees that the operator security objectives are satisfied. We consider the following scenarios.

6.6.2.1 Scenario 3

In this scenario, the objective of the defender is to find an optimal policy in order to minimize the cost of deploying the security countermeasures on the different equipment while setting a constraint on the tolerated residual risk on the system after the countermeasures are deployed. A value of 100000 is set for the tolerated residual risk. The highest ranking countermeasures are depicted in Table 6.2 (the result is presented in percentages and derived from the equations in Section 5.4.5).

In this scenario, we notice a number of nontrivial results. For example, intervening on the AMI head-end database server has higher priority than intervening on the AMI head-end application server. The ranking of the HIDS on the database server is also similar to the ranking of the HIDSs on firewalls F1 and F2. The highest priority countermeasure that needs to be deployed is patching the REV on the AMI head-end communication server. We also notice that the ranking of patching the LEV on the AMI head-end application server is not 0. This result is interesting given that the attacker cannot try to exploit the LEV without exploiting the REV or gaining access to the AMI head-end application server first. As we will see in the next scenario, one of the reasons for this result is the large value of the tolerated residual risk defined in this scenario.

6.6.2.2 Scenario 4

In this scenario, the objective of the defender is to find an optimal policy in order to minimize the cost of deploying the security countermeasures on the different equipment while setting a constraint on the tolerated residual risk on the system after the countermeasures are deployed. A value of 50000 is set for the tolerated residual risk. Table 6.2 depicts the result.

We notice first that, contrary to Scenario 3, the highest priority in this case is patching the REV on the AMI head-end application server. This result reflects the need to protect the critical services provided by this equipment. The second highest priority is patching the REV on the AMI head-end communication server while the next priority is deploying a HIDS on the AMI head-end database server. We notice that the optimal solution requires the deployment of a HIDS on the AMI head-end database instead of patching the

	Countermeasure	Scenario 3	Scenario 4
HE Comm. Server	Patch REV	23.248	22.818
	Patch LEV	0.794	0.471
	Deploy a HIDS	1.648	0.691
HE App. Server	Patch REV	6.291	23.658
	Patch LEV	3.58	1.526
	Deploy a HIDS	9.679	2.89
HE Database	Patch REV	0	0
	Patch LEV	0	0
	Deploy a HIDS	13.806	12.244
Firewall F1	Patch REV	0	0
	Deploy a HIDS	11.716	10.617
Firewall F2	Patch REV	0	0
	Deploy a HIDS	12.217	10.782
Firewall F3	Patch REV	0	0
	Deploy a HIDS	1.947	1.641
Firewall F4	Patch REV	0	0
	Deploy a HIDS	1.987	1.675

TABLE 6.2: Countermeasures ranking when an attacker has access to the AMI head-end

vulnerabilities on this equipment. In fact, the cost associated with applying the patches outweighs the benefits with respect to reducing the risk of compromising the services provided by the database server. In particular, the value of the services provided by the database server is considerably lower than the value of the services provided by the AMI head-end communication and application servers.

6.7 Conclusion

In this chapter, we validated our approach based on Constrained Markov Decision Processes (CMDPs) on an AMI case study. We assumed a worst-case scenario where each equipment in the system is vulnerable to attacks. The operator's objective was to identify the set of security countermeasures needed to protect the system and prioritize their deployment. We simulate different attack scenarios where the attacker managed to have access to the enterprise and AMI head-end networks. The generated optimal security policy is highly correlated to the security objective defined by the operator. In particular, the overall cost of deploying the set of security countermeasures in the optimal policy depends on the tolerated threshold of the residual risk on the system set by the defender. The attack graph presented in Chapter 4 and the CMDP model presented in Chapter 5 offer the system operator the tools needed to assess the security of his system and generate the optimal security policy that satisfies his security objectives.

Chapter 7

Summary and Conclusions

7.1 Thesis Summary

The smart grid is an improved power grid that enables the emergence of new services. It will improve efficiency, ensure reliability, and offer new potentials both to the utility company and the customers. However, the increased use of information technology in the smart grid has the potential to increase its attack surface.

The process of assessing the security risk of attacks on a critical infrastructure such as the smart grid includes two distinct, though complementary, evaluation stages. In the first stage, the defender is interested in evaluating the threat posed by an attacker. In particular, the defender tries to identify the different methods that can be used by an attacker to compromise a target equipment. In the second stage, the defender focuses on assessing the impact of the attack on the targeted equipment. Each part of this thesis focuses on harnessing the results of one of these evaluation stages with the objective of defining optimal defense strategies to protect the system.

In Part I, we focused on the impact of attacks on the smart grid. We assumed that the attacker can compromise a number of equipment, and we were interested in optimizing the defense resources to protect these equipment. We employed game theoretic techniques to achieve this objective. In Chapter 2, we investigated the optimal choice of security modes that needs to be enabled on equipment in the AMI to protect the confidentiality of customers' data. In the framework of a Stackelberg game, by analyzing the behavior of the attacker and the defender at the Nash equilibrium, we derived the minimum defense resources needed to thwart any attack attempt to compromise customers' data in the AMI. Chapter 3 addressed the challenge of managing the security risk in the smart grid in the presence of interdependencies between the communication and the electrical infrastructures. Using game theory, we analyzed the interactions between the attacker and the defender and computed the optimal distribution of defense resources on communication equipment to minimize the impact of attacks on the power grid.

Part II of the thesis was dedicated to the problem of assessing the threat of attacks on the system and generating optimal security policies to cope with this threat. In Chapter 4, we presented a model for generating an attack graph of a targeted system. Using information about equipment, services, and security mechanisms in the system, the sequence of actions that can be executed by a given attacker is identified. Chapter 5 leveraged the information in the generated attack graph and information about the available defense countermeasures to compute an optimal security policy to protect the system. Finally, Chapter 6 presented a validation of our approach on an AMI case study.

7.2 Open Issues

The problems addressed in this thesis and their resolution resided on a set of assumptions about the system and the key players involved. One of the main assumptions in game theory is the existence of rational decision makers. This assumption plays a pivotal role in our game theoretical analysis of data confidentiality attacks in the AMI and the security risk management in the smart grid. In our case, we consider the worst-case scenario in which the design of the security policy assumes an interaction with an intelligent and strategic attacker. Among a set of available actions, a rational player chooses the action that yields the best payoff. Therefore, the rationality is rooted in the basic set of rules that guide the decision-making process of a player facing the choice between multiple actions, each with a certain value or utility. Irrational players, in that respect, diverge from the most profitable action. In addition, the players in game theory in general are assumed to have unlimited computational capacities, which is a strong assumption that does not hold in practical scenarios. To address this limitation, the concept of bounded rationality has been introduced [Rub98]. In this case, constraints and costs are added to the acquisition of information needed to make rational decisions. Overcoming the limitations of the rationality hypothesis is therefore an important requirement to improve our model and push forward the applicability of the game theoretical results to real-world scenarios.

In Part II of the thesis, based on the information in an attack graph, we presented an approach to derive the optimal security policy in order to reduce the risk of attacks on the system. The generation process of the attack graph is based on the assumption that information about equipment, the services running in the network, and the deployed security mechanisms is available. In practice, the availability of this type of information may not be guaranteed, which will impact the accuracy of the generated attack graph. Therefore, a significant effort must be made in defining methods, tools, and processes to collect the required information about the system. Another issue is the assumptions about the profile of the attacker and his initial knowledge about the system. In this thesis, we propose an approach to find an optimal policy that protects the system from a given attacker. However, how to define the different parameters in the profile of the attacker was beyond the scope of the work in this thesis, and must further be investigated.

Finally, one of the remaining challenges is the assessment of the different parameters used in generating the attack graph and in computing the optimal security policy. The values of these parameters depend on intrinsic characteristics about vulnerabilities, the deployed security mechanisms in the system, and the available set of defense countermeasures. Statistical data is a valuable tool when used in conjunction with experts' knowledge to evaluate the efficiency of security countermeasures against certain types of attackers. Even though sensitivity analysis can be conducted to evaluate the impact of uncertainties on some of these parameters, it is important to formally investigate the impact of uncertainties about the profile of the attacker and the different parameters in the model on the accuracy of the attack graph and the efficiency of the computed optimal security policy.

7.3 Directions for Future Research

In this section, we provide potential extensions and future directions for the work presented in this thesis. In Chapter 2, we presented a game theoretical model for data confidentiality attacks on the smart grid AMI. As future work, it will be interesting to investigate the impact of the false alarm rate for attack detection on players' behaviors. In this case, at the Nash equilibrium, it is important to evaluate the changes in the sensible target set with respect to changes in the attacker behavior. In particular, it will be interesting to find whether the attacker will seek to increase his resources to target more high value assets instead of targeting low value assets. Another research direction will be to extend the model to include additional players' actions. For example, the defender can choose between different encryption algorithms on each device, where each algorithm is characterized by its robustness and cost. Another possible action for the defender is the ability to reconfigure network connections when the system is under attack.

In Chapter 3, we presented a game theoretical model for hardening security on communication equipment to reduce the risk of attacks on the power system. The interdependency model in Chapter 3 was an initial step to formally represent the effects of interdependencies between electric and communication infrastructures. A more fine-grained analysis of the risk of cyber attacks on the power system could be achieved by including specific control functions in the power grid. In Chapter 3, we proposed a method to evaluate the values of parameters used in our model to assess the impact of attacks in the electric system. Investigating methods and tools to assess the other parameters is an important step to the application of the model in realistic scenarios. For example, evaluating the impact of attacks in the communication infrastructure can be achieved using the information in the attack graph for the corresponding system generated in Chapter 4. Investigating the effects of partial knowledge of the parameters and the architecture of the system on the strategies of both the attacker and the defender is also an interesting extension to the presented work. Further explorations would include studying the existence of multiple attackers and the impact of their cooperation on the power system.

Finally, in Chapter 5, we presented an approach to compute an optimal security policy tailored for an industrial control system based on information in the attack graph generated in Chapter 4. The policy takes into account the objective of the defender knowing the potential impact of the attacker's actions on the system. It will be interesting to examine whether the optimization yields the same result when the attacker adapts his response to the deployed security countermeasures by the defender. In this case, the choice of the next attack action to be executed will not only depend on the profile of the attacker and the potential impact of the attack on the system, but also on the action of the defender. Analyzing this type of interdependencies using a game theoretical framework can provide valuable insights on the design of secure control systems and the hardening of security on existing systems. The analysis conducted thus far resided on the actions of a single attacker targeting the system. Introducing more sophisticated cases such as the existence of multiple attackers is an interesting extension to our work. In particular, the analysis of the impact of the cooperation between the different attackers, each with a certain profile, can provide important guidelines to optimize the defender's response to coordinated intrusions.

7.4 Concluding Remark

This thesis addressed a number of security challenges in the smart grid. While each part of the thesis focused on leveraging the results of a risk assessment evaluation stage to improve the security of the system, it would be interesting to combine the results in a unifying framework for smart grid security risk management. In this thesis, we proposed a model for generating the potential attacker's actions in the system. We believe that improving the generation process by using a game theoretical framework to model the interactions between the attacker and the defender is an important extension to the presented work. This approach will further enhance our understanding of the evolution of a rational attacker's behavior in the system and the efficiency of a defender's security policy. Studying the impact of cooperation between multiple attackers, each with a certain profile, on the system is an interesting avenue for future research. In this case, the best protection may not only be achieved based on the efficiency of the available security countermeasures, but also on the deployment time of the optimal set of countermeasures and the number of defenders involved.

Appendix A

Publications

Journal Papers

1. **Z. Ismail**, J. Leneutre, D. Bateman, and L. Chen. A game theoretical analysis of data confidentiality attacks on smart grid AMI. *IEEE Journal on Selected Areas in Communications*, 32(7):1486–1499, July 2014.
2. **Z. Ismail**, C. Kiennert, J. Leneutre, and L. Chen. Auditing a cloud provider's compliance with data backup requirements: A game theoretical analysis. *Accepted in IEEE Transactions on Information Forensics and Security.*
3. **Z. Ismail**, J. Leneutre, D. Bateman, and L. Chen. A quantitative model for security risk management of interdependent ICT and electrical infrastructures in the smart grid. *Preprint.*
4. **Z. Ismail**, J. Leneutre, A. Fourati, and L. Chen. Optimal defense policies to secure critical industrial control systems. *Preprint.*
5. **Z. Ismail**, C. Kiennert, J. Leneutre, and L. Chen. A game theoretical model for optimal distribution of network security resources. *Preprint.*

Conference Papers & Demonstrations

1. **Z. Ismail**, J. Leneutre, D. Bateman, and L. Chen. A game-theoretical model for security risk management of interdependent ICT and electrical infrastructures. *In IEEE 16th International Symposium on High Assurance Systems Engineering (HASE)*, pages 101–109, 2015.
2. **Z. Ismail**, J. Leneutre, and A. Fourati. An attack execution model for industrial control systems security assessment. *In the 1st Conference on Cybersecurity of Industrial Control Systems (CyberICS)*, 2015.
3. **Z. Ismail**, D. Symeonidou, and F. Suchanek. DIVINA: Discovering vulnerabilities of internet accounts. *International World Wide Web Conference (WWW)*, 2015. Demo paper.

Appendix B

Auditing a Cloud Provider's Compliance with Data Backup Requirements: A Game Theoretical Analysis

The new developments in cloud computing have introduced significant security challenges to guarantee the confidentiality, integrity, and availability of outsourced data. A Service Level Agreement (SLA) is usually signed between the cloud provider and the customer. For redundancy purposes, it is important to verify the cloud provider's compliance with data backup requirements in the SLA. There exists a number of security mechanisms to check the integrity and availability of outsourced data. This task can be performed by the customer or be delegated to an independent entity that we will refer to as the verifier. However, checking the availability of data introduces extra costs, which can discourage the customer from performing data verification too often. The interaction between the verifier and the cloud provider can be captured using game theory in order to find an optimal data verification strategy. In this chapter, we formulate this problem as a two player non-cooperative game. We consider the case in which each type of data is replicated a number of times which can depend on a set of parameters including, among others, its size and sensitivity. We analyze the strategies of the cloud provider and the verifier at the Nash Equilibrium and derive the expected behavior of both players. Finally, we validate our model numerically on a case study and provide guidelines on how to evaluate the game parameters.

B.1 Introduction

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage,

applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [MG11]. However, all the benefits brought by the Cloud, such as lower costs and ease of use, come with a tradeoff. In particular, users have to entrust their data to a cloud provider (CP), which can be viewed as a selfish entity aimed at maximizing profits. This could lead the CP to act in ways that are detrimental to users' interests. The new security issues introduced by cloud computing need to be addressed and are of interest to both industry and academia [AFG+10].

One aspect of cloud computing is the ability to buy or lease storage capacity, which introduces security problems related to data integrity and availability. The client often lacks full control over the manner his data is stored, entailing difficulties in ensuring that data stored in the Cloud are indeed left intact. A number of guarantees are given through the *Service Level Agreement* (SLA) which is a contract between the CP and the client that defines the expected level of the service offered by the CP. This includes in particular the overall availability rate, i.e. the expected downtime per year. In addition, an SLA can include other features such as the number of data backups, which may be physically stored at different geographical locations. However, in a worst-case scenario, a CP may not respect the requirements of the backup process of some of the entrusted data to save both money and storage space capacities. By behaving this way, the CP may not directly cause data losses for the client (as the original copy can be left intact), but raises the probability of accidental data loss happening (related to hazards), which impacts the overall data availability rate.

The client may be interested in checking the availability of all data backups using specific protocols such as proofs of data retrievability widely studied in the literature [KW14]. In these works, efforts have been made to design solutions that meet various requirements such as low time complexity, stateless verification, unbounded use of queries, and retrievability of data, etc. In particular, several protocols allow public verifiability from a Third Party Auditor (TPA), to which the client can delegate the verification task through an *Audit Level Agreement*. This assumption is more realistic, since in most cases, a lack of resources or expertise will prevent the client from personally performing these verifications. In this chapter, we will consider that the TPA, which is an independent entity, will be the verifier of the client's data in the CP systems.

In spite of the numerous features of the verification schemes, choosing the efficient set of features to use remains a challenging task. For example, it would be a waste of both time and resources for the verifier to check the client's data all the time in the case of an honest CP. On the other hand, it would be risky if the data is not checked regularly when the CP is acting dishonestly. Therefore, in order to analyze the interactions between the CP and the verifier, and derive their expected behaviors to find the optimal verification strategy for the verifier, we model the data availability verification problem as a two player non-cooperative static game featuring the cloud provider and the TPA. We introduce a number of extensions to the basic model in [DKLC14] to take into account more realistic scenarios. In particular, we consider a model featuring multiple copies of each data stored by the CP and analyze the behavior of both players under different types of strategies.

The remainder of this chapter is organized as follows. In Section B.2, we describe the technical background and related work. In Section B.3, we formulate the untrusted cloud storage game in which we consider the existence of multiple copies of each data on the cloud provider's servers. In Section B.4, we start by analyzing two types of one-shot games related to the dependency of players' strategies on a certain data on other data. Then, we present a second formulation of the problem as a stackelberg game. Section B.5 provides numerical results validating our analysis. Section B.6 provides guidelines on how to evaluate the parameters in the model in a practical scenario, and illustrates it with a numerical example. Finally, we conclude this chapter in Section B.7.

B.2 Related Work

In untrusted cloud storage, it is important to verify the cloud provider's compliance with the security requirements in the SLA. For example, Popa et al. [PLM+11] designed a proof-based system to enable security guarantees in an SLA. In recent years, a significant amount of data integrity schemes were proposed by different researchers, and have been gradually adapted to specific use cases such as outsourced databases and cloud computing, for which works focusing on public verifiability issues, such as [WWRL10], were noticeably helpful and allowed clients to delegate the verification process to third parties. Among these schemes, the two main directions explored by researchers include the Provable Data Possession (PDP) for ensuring possession of data, and the Proof of Retrievability (POR) for data possession and retrievability. The main idea of PDP is that a data owner generates some metadata information for a data file to be used later for verification purposes. Many extensions of this scheme managed to decrease the communication cost and complexity [ADMT08], as well as to allow dynamic operations on data such as insertion, modification, or deletion [EKPT09]. Moreover, Zhu et al. [ZWH+10] and Yang et al. [YWW+11] proposed PDP schemes specific to cloud computing.

The POR scheme is considered as a complementary approach to PDP. [JK07] was among the first papers to consider formal models for POR schemes. In this scheme, disguised blocks (called sentinels) are embedded into the data before outsourcing. The verifier checks randomly picked sentinels, which would be influenced with a certain probability if the data is corrupted. An improved version of the POR approach was achieved with compact proofs of retrievability [SW08a], with the design of a stateless protocol with unbounded audit interactions. Kochumol and Win [KW14] give a detailed survey of the contributions of numerous extensions of the PDP and POR schemes. However, the schemes presented so far focus primarily on a single copy of a data file. Other schemes, such as [CKBA08], allow the verifier to check multiple copies of a data file on multiple cloud servers.

In the cloud domain, game theory has emerged in recent years as an important tool to analyze the interactions between multiple players with the same or conflicting interests. It has been used to study a number of problems including resource allocation and

management [HSH11] and cloud service negotiation [ZMPB10], while some research papers addressed the problem of cloud security [NK12a, NK12b]. To address cloud integrity issues, Nix and Kantarcioglu [NK12a] propose a model in which a client checks the correctness of calculations made on the data by the CP. In [NK12b], Nix and Kantarcioglu study the case of querying one cloud provider, since checking data at multiple CPs is prohibitively expensive. [NK12a] and [NK12b] focused on checking whether the queries sent to the CP are being computed correctly, under the condition that the stored data is intact. On a side note, they did not mention which type of verification protocol (deterministic or probabilistic) they used. In addition to cloud-related problems, game theory has been used in multiple domains including network security [GM12, AB10], intrusion detection [CL09], Botnet defense [BKH10], among others.

B.3 Untrusted Cloud Storage Game Formulation

We consider a client outsourcing a set $\mathcal{D} = \{D_1, D_2, ..., D_N\}$ of N data to a cloud provider (CP). We consider the case in which the client delegates the data availability verification process to a Third Party Auditor (TPA).

We model the data availability problem as a non-cooperative static game with two players, the cloud provider and the TPA. We assume that both players are rational. The CP tries to gain storage space by not backing up correctly the client's data without being caught. On the other hand, the objective of the TPA is to distribute verification resources in order to detect partially of fully unbacked up data. Using the model defined in [DKLC14] as a basis, we introduce an important extension related to the existence of multiple copies of the same data on the CP servers. This assumption has many important implications, as the CP has the possibility to dishonor his backup commitments in a more stealthy way without compromising the original version of the data. On the other hand, the verifier (TPA) will need to improve his verification strategy to check not only the existence of a data file, but the existence of the required number of backups to that file as well. This new extension to the model will allow us to analyze the behavior of both players from which we derive the optimal verification strategy. This scenario is closer to what we might expect to have in a real-world setting. Although this game features interactions between only two players, several users may delegate the verification process to a same TPA. On the other hand, the case of a TPA verifying multiple cloud providers can be regarded as independent occurrences of the two-player game, except in the case where the cloud providers cooperate against the TPA, which leads to an entirely different scenario. Therefore, the proposed model covers a wide range of realistic situations, and can be applied to the case of multiple independent users relying on the same TPA, as well as to the case of auditing multiple independent cloud providers.

The reputation of a cloud provider will rely, among other factors, on the availability rate of clients' data. This can be achieved by keeping a number of copies of the data. This

number can be a function of the importance of the data to the client in addition to its size. The data can be kept on the same server or distributed on multiple geographically dispersed servers. This will offer a higher availability rate and improve the resiliency against attacks, accidents, and hazards targeting specific locations. In this section, we extend the previous model in [DKLC14] with the assumption of the existence of multiple copies of the data on the CP servers. In this case, with the absence of a verification mechanism, the CP can remove the additional copies of a data file without impacting the client's access to that file. However, the CP takes a risk if the remaining copy of the data became unavailable for some reason. The CP will try to weigh the risk of behaving in a malicious way with the possible benefit of increasing the storage space, which translates in practice to additional profits.

We associate to each data D_i the following parameters: the financial storage cost $S^i \geq 0$ of one copy of data D_i by the CP, which is proportional to data D_i's size; the financial value $F^i \geq 0$ of one copy of D_i quantifying how critical data D_i is to the client. The cost of processing the verification query for the TPA and the cost of executing the verification query by the cloud provider are supposed to be proportional to S^i and are given by $C^t S^i$ and $C^s S^i$ respectively, where $0 \leq C^s, C^t \leq 1$. In addition, let R^i refer to the number of backup copies of data $D_i \in \mathcal{D}$ that needs to be stored on the CP servers and ϵF^i the reward (e.g. in reputation) the CP gets if he acts honestly or passes the verification test undetected by the TPA otherwise, where $\epsilon > 0$.

In this chapter, we make the following assumptions:

Assumption B.1. *The costs related to network communications, both on the CP side and between the CP and TPA, are ignored.*

The model presented in this chapter aims only at analyzing whether the CP will behave honestly or dishonestly. The possible storage flaws of an honest CP are out of the scope of this work. Therefore, we make the following assumption:

Assumption B.2. *The way the backup copies of the data are stored on the cloud provider servers is not taken into account.*

Assumption B.3. *The probability of data corruption remaining undetected by the TPA after a check is neglected, even when using a probabilistic protocol.*

The approximation in Assumption B.3 is justified by the fact that a dishonest CP will be likely to entirely omit one or more copies of the data, rather than keep parts of the copies stored on his servers. Nevertheless, such scenario was already taken into account in one of the models presented in [DKLC14]. While it is possible to take into account such scenario in the present work, we believe it will impact the clarity of the presentation of the model and increase the complexity of already complex equations.

Table B.1 presents the payoffs for both players for data D_i in the case where $R^i = 1$. In this case, if the corrupted or unavailable data D_i is not checked, the CP gains a payoff

TABLE B.1: Cloud storage game with deterministic verification for data D_i

CP \ TPA	Check	Not check
Replicate/Available data	ϵF^i , $-C^tS^i - C^sS^i$	0 , 0
Not replicate/Unavailable data	$-C^sS^i - S^i$, $-C^tS^i + F^i$	S^i , $-F^i$

S^i proportional to the size of data D_i while the TPA loses F^i. In addition to the cost of processing the verification query C^tS^i, we consider that the TPA should pay the cost of executing the verification query C^sS^i when he decides to verify D_i in the case where the CP respects the backup process of the data. However, when the CP chooses not to respect the backup process on D_i and the TPA chooses to verify, the TPA will gain F^i while paying for the verification cost C^tS^i, and the CP will lose S^i while paying for the cost of executing the verification query C^sS^i. Finally, neither players will achieve anything when the TPA decides not to verify D_i and the CP respects the backup process. In this chapter, we focus on the number of backup copies of data D_i that can be checked by the TPA. We assume that an original version of the data is present on the CP servers and that version will not be targeted by the CP. Therefore, the TPA will be interested in verifying that the required number of backup copies R^i for each type of data D_i agreed on between the CP and the client is indeed present on the CP servers.

Let $\mathbb{1}$ represent the indicator function. We refer by p_0^m the probability that the CP respects the requirements of the backup process for data D_m. $\forall 1 \leq i \leq R^m$, let p_i^m denote the probability that the CP does not keep i copies of data D_m. Similarly for the verifier, we refer by q_0^m the probability that the TPA does not check the existence of any copy of data D_m, and $\forall 1 \leq j \leq R^m$, q_j^m the probability that the TPA verifies the existence of j copies of data D_m.

The utilities U_A and U_D of the cloud provider and the TPA respectively are as follows:

$$\begin{aligned} U_A(p,q) = \sum_{m=1}^{N} \Big\{ & -\sum_{i=1}^{R^m}\sum_{j=1}^{R^m} p_i^m q_j^m (iS^m + jC^sS^m)\mathbb{1}_{i>R^m-j} + \sum_{i=0}^{R^m}\sum_{j=1}^{R^m} \epsilon p_i^m q_j^m (jF^m)\mathbb{1}_{i\leq R^m-j} \\ & + \sum_{i=1}^{R^m}\sum_{j=0}^{R^m} p_i^m q_j^m (iS^m)\mathbb{1}_{i\leq R^m-j} \Big\} \end{aligned}$$

$$\begin{aligned} U_D(p,q) = \sum_{m=1}^{N} \Big\{ & \sum_{i=1}^{R^m}\sum_{j=1}^{R^m} p_i^m q_j^m (iF^m)\mathbb{1}_{i>R^m-j} - \sum_{i=0}^{R^m}\sum_{j=1}^{R^m} p_i^m q_j^m (jC^sS^m)\mathbb{1}_{i\leq R^m-j} \\ & - \sum_{j=1}^{R^m} q_j^m C^tS^m j - \sum_{i=1}^{R^m}\sum_{j=0}^{R^m} p_i^m q_j^m (iF^m)\mathbb{1}_{i\leq R^m-j} \Big\} \end{aligned}$$

The actions of both the CP and the TPA will determine their utilities. The actions of the CP that result in him getting a positive payoff are limited to the case where the number of copies j that has been checked by the verifier is less than the number of copies that remain on the CP servers after the CP has kept $R^m - i$ copies. In this case, the CP benefits

from the value he gets from the additional storage space that has been freed up in addition to a reward for passing the verification test. The TPA, on the other hand, gets a negative payoff that includes the importance of the i copies that were not kept by the CP and that were undetected and the cost of processing the verification query. Otherwise ($i > R^m - j$), the CP gets a negative payoff that includes the cost of executing the verification query in addition to the value of the storage space that needs to be reallocated to the client's data. In this case, the TPA gets a positive payoff related to the importance of the copies of the data that were not kept by the CP and whose absence was detected by the TPA.

B.4 Solving the Game

B.4.1 Independent Strategies One-shot Game

In this section, we investigate the case where both the CP and the TPA take their decisions at the same time while taking into account each other's strategies. This type of interactions falls under the one-shot game category [OR94]. We suppose that the strategy of each player for a data item D_i is independent from the other data items D_j. We define an independent strategies game as follows:

Definition B.1 (Independent Strategies game)**.** *An Independent Strategies (IS) game is a game in which each player's strategy for each data D_i do not depend on other data D_j, $\forall j \neq i$.*

In this case, we have $\sum_{i=0}^{R^m} p_i^m = 1$ and $\sum_{j=0}^{R^m} q_j^m = 1$, $\forall m \in \{1, ..., N\}$ where m refers to data D_m.

Let $\theta_i^m = 2iF^m + (R^m - i)C^sS^m$ and $\phi_i^m = 2(R^m - i)S^m + i(C^sS^m + \epsilon F^m)$.

Theorem B.1. *The NE of the IS game for the CP and the TPA is expressed as follows, $\forall m \in \{1, ..., N\}$:*

$$\begin{cases} p_0^{m*} = \dfrac{\dfrac{2R^mF^m - C^tS^m}{2R^mF^m + C^sS^m} - C^tS^m \displaystyle\sum_{i=1}^{R^m-1} \frac{1}{\theta_i^m} \prod_{j=1}^{i-1} \left(1 + \frac{C^sS^m}{\theta_j^m}\right)}{1 + C^sS^m \displaystyle\sum_{i=1}^{R^m-1} \frac{1}{\theta_i^m} \prod_{j=1}^{i-1} \left(1 + \frac{C^sS^m}{\theta_j^m}\right)} \\ p_i^{m*} = \dfrac{C^sS^m(p_0^m)^* + C^tS^m}{\theta_i^m} \displaystyle\prod_{j=1}^{i-1} \left(1 + \frac{C^sS^m}{\theta_j^m}\right) \qquad \forall i \in \{1, ..., R^m - 1\} \\ p_{R^m}^{m*} = \dfrac{C^tS^m + C^sS^m}{2R^mF^m + C^sS^m} \end{cases}$$

$$\begin{cases} q_0^{m*} = \dfrac{1 - \dfrac{S^m}{2S^m + \phi_{R^m}^m} + S^m \displaystyle\sum_{i=1}^{R^m-1} \frac{1}{\phi_i^m} \prod_{j=1}^{i-1} \left(1 + \frac{2S^m}{\phi_j^m}\right)}{1 + 2S^m \displaystyle\sum_{i=1}^{R^m-1} \frac{1}{\phi_i^m} \prod_{j=1}^{i-1} \left(1 + \frac{2S^m}{\phi_j^m}\right)} \\ q_i^{m*} = \dfrac{(2q_0^{m*} - 1)S^m}{\phi_i^m} \displaystyle\prod_{j=1}^{i-1} \left(1 + \frac{2S^m}{\phi_j^m}\right) \qquad \forall i \in \{1, ..., R^m - 1\} \\ q_{R^m}^{m*} = \dfrac{S^m}{2S^m + \phi_{R^m}^m} \end{cases}$$

Proof. In this case, considering the data independence hypothesis, we solve the game by focusing on any fixed data D_m independently from the other data. Considering that $p_0^m = 1 - \sum_{i=1}^{R^m} p_i^m$, and $q_0^m = 1 - \sum_{i=1}^{R^m} q_i^m$ and integrating these constraints in the payoff functions, at the optimum we have: $\frac{\partial U_A(p,q)}{\partial p_i^m} = 0$ and $\frac{\partial U_D(p,q)}{\partial q_i^m} = 0$, $\forall i \in \{1, ...R^m\}$.

We have $\forall j \in \{1, ..., R^m\}$:

$$\begin{aligned} \frac{\partial U_D(p,q)}{\partial q_j^m} = &\sum_{i=1}^{R^m} p_i^m (iF^m) \mathbb{1}_{i>R^m-j} - jC^t S^m - \sum_{i=0}^{R^m} p_i^m (jC^s S^m) \mathbb{1}_{i \leq R^m - j} \\ &+ \sum_{i=1}^{R^m} p_i^m (iF^m) - \sum_{i=1}^{R^m} p_i^m (iF^m) \mathbb{1}_{i \leq R^m - j} \end{aligned}$$

We have $\frac{\partial U_D(p,q)}{\partial q_j^m} - \frac{\partial U_D(p,q)}{\partial q_{j-1}^m} = 0$, $\forall j \geq 2$. Therefore, $\forall i \in \{1, ..., R^m\}$, we have:

$$(p_i^m)^* \theta_i^m = C^t S^m + C^s S^m \sum_{j=0}^{i-1} (p_j^m)^* \tag{B.1}$$

From Equation B.1, we prove by induction the following result:

$$\forall i \in \{1, ..., R^m - 1\}, \; (p_i^m)^* = \frac{C^s S^m (p_0^m)^* + C^t S^m}{\theta_i^m} \prod_{j=1}^{i-1} \left(1 + \frac{C^s S^m}{\theta_j^m}\right)$$

Moreover, solving $\frac{\partial U_D(p,q)}{\partial q_1^m} = 0$ gives $(p_{R^m}^m)^* = \dfrac{C^t S^m + C^s S^m}{2R^m F^m + C^s S^m}$

Given the constraint $\sum_{i=0}^{R^m} p_i^m = 1$, we find $(p_0^m)^*$.

Similarly, we find the TPA strategy at the NE $(q_i^m)^*$. □

We can notice that $q_0^{m*} > 0.5$, $\forall m \in \{1, ..., N\}$. This result can be interpreted as the following. When the verifier wants to decide whether to check the existence of a data D_i, he has a choice between performing the verification of a number i of copies of the data or dropping his request. When the TPA prefers not to check over performing any checking ($q_0^m \geq \sum_{j=1}^{R^m} q_j^m$), he allocates nevertheless some resources to execute verification queries. This will ensure that the CP will operate at the NE, and therefore cannot improve his utility by changing his strategy unilaterally.

With respect to the CP's strategy at the NE, we have the following Lemma:

Lemma B.1. *In the case of an IS game,* $\exists!\ x_0^m = F^m/S^m > 0$ *s.t.* $p_0^{m*}(x_0^m) = 0$.

Proof. Let $x = F^m/S^m$. In this case, p_0^{m*} can be written as $p_0^{m*} = \dfrac{1 - \dfrac{C^t + C^s}{2R^m x + C^s} - C^t\Delta}{1 + C^s\Delta}$, where $\Delta = \sum_{i=1}^{R^m-1} \dfrac{1}{2ix + (R^m - i)C^s} \prod_{j=1}^{i-1} \left(1 + \dfrac{C^s}{2jx + (R^m - j)C^s}\right)$.

Therefore, $\dfrac{\partial p_0^{m*}}{\partial x} = \dfrac{\dfrac{2R^m(C^t + C^s)}{2R^m x + C^s}\left(\dfrac{1}{2R^m x + C^s} - x\dfrac{\partial\Delta}{\partial x}\right)}{(1 + C^s\Delta)^2}$.

$$\frac{\partial\Delta}{\partial x} = \sum_{i=1}^{R^m-1} \frac{-2i}{(2ix + (R^m - i)C^s)^2} \prod_{j=1}^{i-1} \left(1 + \frac{C^s}{2jx + (R^m - j)C^s}\right)$$
$$+ \sum_{i=1}^{R^m-1} \frac{1}{2ix + (R^m - i)C^s} \sum_{k=1}^{i-1} \frac{-2kC^s x}{(2ks + (R^m - k)C^s)^2} \prod_{j=1, j\neq k}^{i-1} \left(1 + \frac{C^s}{2jx + (R^m - j)C^s}\right) < 0.$$

As a result, $\dfrac{\partial p_0^{m*}}{\partial x} > 0$ and p_0^{m*} is a strictly increasing function with respect to F^m/S^m.

p_0^{m*} is a continuous function in $[0; +\infty[$. We have for $F^m = 0$, $p_0^{m*} < 0$. For $F^m/S^m \to +\infty$, $p_0^{m*} \to 1$, and as a result $\exists y > 0$ s.t. $\forall F^m/S^m > y$, $p_0^{m*} > 0$. Therefore, by the Intermediate Value Theorem and the fact that p_0^{m*} is a strictly increasing function w.r.t. F^m/S^m, there exists only one value $x_0^m = F^m/S^m$ s.t. $p_0^{m*}(x_0^m) = 0$. □

As a consequence of Lemma B.1, the condition for the existence of the NE in this case is that $F^m/S^m > x_0^m$, $\forall m \in \{1, ..., N\}$. If that condition is not respected for data D_m, the CP is better off deleting at least one copy of the data. However, the TPA will respond by verifying the existence of the maximum number of copies as required in the backup process agreed on between the CP and the client. In this case, the TPA will make sure that he will always catch a dishonest CP and gets rewarded for his actions. Unfortunately, this scenario does not allow the emergence of a NE.

B.4.2 Correlated Strategies One-shot Game

As in the previous section, we investigate the case where both the CP and the TPA take their decisions at the same time. However, we suppose that the strategies for data items D_i are interdependent. In this case, the players' choices for their strategies for data D_i depend on their strategies for data D_j, $\forall j \neq i$. There are two possible scenarios. In the first scenario, we limit the actions of each player on one data item at each instance of the game. For example, at a given moment, the verifier will issue a query to verify only the existence of backups for data D_i. However, this scenario is limiting in practice, as sometimes it is more beneficial for the TPA to issue queries to verify the existence of backups for different types of data at once. In this case, we consider that at each instance of the game, each player can execute an action on each type of available data. For example, the CP can dishonor his backup commitments on a set of data items at once. Nevertheless, in the remaining of this section, we analyze the behavior of the CP and the TPA in both scenarios.

B.4.2.1 Single Targets

We define a correlated strategies single targets game as follows:

Definition B.2 (CSST game)**.** *A Correlated Strategies Single Targets (CSST) game is a game in which each player can target one type of data and execute one action related to that data at each instance of the game.*

In practice, this translates to having $\sum_{m=1}^{N}\sum_{i=0}^{R^m} p_i^m = 1$ and $\sum_{m=1}^{N}\sum_{j=0}^{R^m} q_j^m = 1$. In this case, $\sum_{i=0}^{R^m} p_i^m$ and $\sum_{j=0}^{R^m} q_j^m$ refer to the probability of targeting data D_m for the CP and the TPA respectively.

Let parameters ψ_i^m, ω^m, τ^m, α^m, β^m, γ^m, δ^m, and η^m be defined as in Appendix D.

Theorem B.2. *The NE of the CSST game for the CP and the TPA is expressed as follows,* $\forall m \in \{1, ..., N\}$*:*

$$\begin{cases} p_0^{m*} = \dfrac{\alpha^m}{\beta^m} \\ p_i^{m*} = \dfrac{C^s\alpha^m S^m + C^t\beta^m S^m}{\beta^m \theta_i^m}\psi_i^m & \forall i \in \{1, ..., R^m - 1\} \\ p_{R^m}^{m*} = \dfrac{(C^s\alpha^m S^m + C^t\beta^m S^m)(R^m - 2\omega^m)}{2\beta^m R^m F^m} \end{cases}$$

$$\begin{cases} q_0^{m*} = \dfrac{\eta^m}{\sum\limits_{i\in\mathcal{D}} \eta^i\big(1+\delta^i+\gamma^i\phi_1^i \sum\limits_{j=1}^{R^i-1}\frac{1}{\phi_j^i}\big)} \\ q_i^{m*} = \dfrac{\gamma_m\eta^m\phi_1^m}{\phi_i^m \sum\limits_{i\in\mathcal{D}} \eta^i\big(1+\delta^i+\gamma^i\phi_1^i \sum\limits_{j=1}^{R^i-1}\frac{1}{\phi_j^i}\big)} \quad \forall i \in \{1,...,R^m-1\} \\ q_{R^m}^{m*} = \dfrac{\delta_m\eta^m}{\sum\limits_{i\in\mathcal{D}} \eta^i\big(1+\delta^i+\gamma^i\phi_1^i \sum\limits_{j=1}^{R^i-1}\frac{1}{\phi_j^i}\big)} \end{cases}$$

Proof. The result is found using a similar analysis as in the proof of Theorem B.1. □

Lemma B.2. *In the case of a CSST game,* $\exists!\ S_1^m, S_2^m > 0$ *s.t.* $\forall S^m \in [S_1^m; S_2^m]$*, we have* $p_0^{m*} \in [0;1]$.

Proof. $p_0^{m*} = \dfrac{\alpha^m}{\beta^m} \Rightarrow \dfrac{\partial p_0^{m*}}{\partial S^m} = \dfrac{\beta^m \frac{\partial \alpha^m}{\partial S^m} - \alpha^m \frac{\partial \beta^m}{\partial S^m}}{(\beta^m)^2}$. Let $\Delta^i = \sum\limits_{j=1}^{R^i-1} \dfrac{S^i\psi_j^i}{\theta_j^i}$ and $B^i = \sum\limits_{j=1}^{R^i-1} \dfrac{jS^i}{\theta_j^i}\psi_j^i$, $\forall i \in \{1,...,N\}$.

We have:

$$\begin{aligned} \beta^m\frac{\partial\alpha^m}{\partial S^m} - \alpha^m\frac{\partial\beta^m}{\partial S^m} = \ & \Big(-\frac{\partial\Delta^m}{\partial S^m} - \frac{1}{2F^m} + \frac{1}{R^m}\frac{\partial B^m}{\partial S^m} \\ & -R^m \sum_{i=1, i\neq m}^{N} \frac{1}{S^iR^i}\big(\Delta^i + \frac{1}{C^s} + \frac{S^i(R^i-2\omega^i)}{2R^iF^i}\big)\Big)\Big(C^s\alpha^m + C^t\beta^m\Big) \end{aligned}$$

However, $C^s\alpha^m + C^t\beta^m = C^s + NC^t > 0$, $\Delta^i - \dfrac{S^i\omega^i}{R^iF^i} > 0\ \forall i \in \{1,...,N\}$, and $-\dfrac{\partial\Delta^m}{\partial S^m} + \dfrac{1}{R^m}\dfrac{\partial B^m}{\partial S^m} < 0$. Therefore, we have $\dfrac{\partial p_0^{m*}}{\partial S^m} < 0$ and p_0^{m*} is a strictly decreasing function with respect to S^m.

p_0^{m*} is a continuous function in $[0;+\infty[$. We have for $S^m = 0$, $p_0^{m*} = 1 + (N-1)\dfrac{C^t}{C^s} > 1$. For $S^m \to +\infty$, $p_0^{m*} \to -\dfrac{C^t}{C^s} < 0$, and as a result $\exists y > 0$ s.t. $\forall S^m > y$, we have $p_0^{m*} < 0$. Therefore, by the Intermediate Value Theorem, there exists only one value S_2^m s.t. $p_0^{m*}(S_2^m) = 0$. In addition, given the fact that p_0^{m*} is a strictly decreasing function w.r.t. S^m, and that $p_0^{m*}(0) > 1$ and $p_0^{m*}(S_2^m) = 0$, there exists only one value $S_1^m > 0$ s.t. $p_0^{m*}(S_1^m) = 1$. □

Given the result of Lemma B.2, a necessary condition for the existence of the NE of the game in this case is that we have $S^m \in [S_1^m; S_2^m]$, $\forall m \in \{1,...,N\}$ where S_1^m and S_2^m are the solutions of equations $p_0^{m*}(S_1^m) = 1$ and $p_0^{m*}(S_2^m) = 0$ respectively.

Lemma B.3. *In the case of a CSST game, a necessary condition for the existence of a NE is that:*

$$\max_{i\in\{1,...,N\}}(S^iR^i) < \frac{N+\frac{C^s}{C^t}}{\sum_{m\in\mathcal{D}}\Big(\frac{1}{S^mR^m}+\frac{C^s}{R^m}(\tau^m+\frac{R^m-2\omega^m}{2R^mF^m})\Big)}$$

Proof. Follows directly from Lemma B.2. □

B.4.2.2 Multiple Targets

We define a correlated strategies multiple targets game as follows:

Definition B.3 (CSMT game)**.** *A Correlated Strategies Multiple Targets (CSMT) game is a game in which each player can target multiple types of data at each instance of the game.*

In addition, in this case, we consider that the resources available to each player are limited. Therefore, we have $\sum_{m=1}^{N}\sum_{i=1}^{R^m}p_i^m = P$ and $\sum_{m=1}^{N}\sum_{j=1}^{R^m}q_j^m = Q$, where P and Q represent the resource constraints of the CP and the TPA respectively. We also have $\sum_{i=0}^{R^m}p_i^m = 1$ and $\sum_{j=0}^{R^m}q_j^m = 1$, $\forall m \in \{1,...,N\}$.

Given the limited resources for the CP and the TPA, we can predict that they may be interested to take actions on a subset of the data stored on the CP servers. Let $\mathcal{D}_S$ denote such subset which we find using Algorithm 11. Let parameters E^m, G^m, H^m, W^m, ν and κ be defined as in Appendix D. We have the following theorem:

Theorem B.3. *If* $\max_{D_m\in\mathcal{D}\setminus\mathcal{D}_S} S^mR^m < \kappa$*, the NE of the CSMT game for the CP and the TPA is expressed as follows,* $\forall m \in \mathcal{D}_S$*:*

$$\begin{cases} p_i^{m*} = \dfrac{-\nu C^sS^m + C^sS^mG^m + C^tS^mE^m}{E^m\theta_i^m}\psi_i^m & \forall i \in \{1,...,R^m-1\} \\ p_{R^m}^{m*} = 1 - C^tS^m\tau^m - (1+C^sS^m\tau^m)(\dfrac{-\nu+G^m}{E^m}) \end{cases}$$

$$\begin{cases} q_i^{m*} = \dfrac{\phi_1^mS^m}{\phi_i^m(\phi_1^m-2S^m)}\Big(\dfrac{2\kappa-2S^m}{H^m(2S^m+\phi_{R^m}^m)}+\dfrac{2(1-W^m)}{H^m}-1\Big) & \forall i \in \{1,...,R^m-1\} \\ q_{R^m}^{m*} = \dfrac{S^m-\kappa}{2S^m+\phi_{R^m}^m} \end{cases}$$

Algorithm 11

Input: The set of data items $\mathcal{D}$
Result: The attractive data set $\mathcal{D}_S$

function FindAttractiveDataSet($\mathcal{D}$)
  $S^{i'} \longleftarrow$ SortInDescendingOrder($\frac{S^{\sigma(i)}(\phi_1^{\sigma(i)}+R^{\sigma(i)}C^s S^{\sigma(i)}+\epsilon R^{\sigma(i)}F^{\sigma(i)})}{\phi_1^{\sigma(i)}-2S^{\sigma(i)}}\sum_{j=2}^{R^{\sigma(i)}-1}\frac{\phi_1^{\sigma(i)}}{\phi_j^{\sigma(i)}}$)
  Initialization: $n_S \longleftarrow N$
  while $n_S \geq 1$ **do**
    $z \longleftarrow \frac{n_S - Q - \sum_{i=1}^{n_S}\frac{(1-W^i)(2S^i+\phi^i_{R^i})-S^i}{H^i(2S^i+\phi^i_{R^i})}}{\sum_{i=1}^{n_S}\frac{1}{H^i(2S^i+\phi^i_{R^i})}}$
    if $(S^{n_S{}'} \leq z)$ **then**
      $n_S \longleftarrow n_S - 1$
    else
      BREAK
    end if
  end while
  $\mathcal{D}_S = \{\sigma(i) \in \mathcal{D} : i \in [\![1, n_S]\!]\}$
end function

Proof. Let us suppose that TPA and the CP focus on the attractive set $\mathcal{D}_S$. Therefore, we have $\sum_{m\in\mathcal{D}_S}\sum_{i=1}^{R^m} p_i^m = P$, $\sum_{m\in\mathcal{D}_S}\sum_{j=1}^{R^m} q_j^m = Q$, $\sum_{i=0}^{R^m} p_i^m = 1$ and $\sum_{j=0}^{R^m} q_j^m = 1$ $\forall m \in \mathcal{D}_S$ where m refers to data D_m.

We find the strategy p^m of the CP by solving the following system of equations:

$$\begin{cases} \frac{\partial U_D(p,q)}{\partial q_i^m} = \nu,\ \forall m \in \mathcal{D}_S\ \forall i \in \{1, ..., R^m - 1\} \text{ where } \nu > 0 \\ \sum_{i=0}^{R^m} p_i^m = 1\ \forall m \in \mathcal{D}_S \\ \sum_{m\in\mathcal{D}_S}\sum_{i=1}^{R^m} p_i^m = P \end{cases}$$

Similarly, we find the strategy q^m of the TPA by solving the following system of equations:

$$\begin{cases} \frac{\partial U_A(p,q)}{\partial p_i^m} = \kappa,\ \forall m \in \mathcal{D}_S\ \forall i \in \{1, ..., R^m - 1\} \text{ where } \kappa > 0 \\ \sum_{j=0}^{R^m} q_j^m = 1\ \forall m \in \mathcal{D}_S \\ \sum_{m\in\mathcal{D}_S}\sum_{j=1}^{R^m} q_j^m = Q \end{cases}$$

Let us suppose that the TPA focuses on the attractive set $\mathcal{D}_S$. We want to find out whether the CP will only be interested in targeting data in $\mathcal{D}_S$ or if he will attempt to target any data $D_i \in \mathcal{D}\backslash\mathcal{D}_S$.

We consider a strategy vector p for the CP s.t. $\sum_{m\in\mathcal{D}\backslash\mathcal{D}_S}\sum_{i=1}^{R^m} p_i^m > 0$.

Let $x \in \mathcal{D}_S$ and $r \in \{1, ..., R^x\}$. We define a vector p' based on p as follows:

$$p_i^{m'} = \begin{cases} p_i^m & m \in \mathcal{T}_S \text{ and } m \neq x,\ i \in \{1, ..., R^m\} \\ p_i^x & m = x,\ i \in \{1, .., r-1, r+1, ..., R^x\} \\ p_r^x + \sum_{j\in\mathcal{D}\backslash\mathcal{D}_S}\sum_{i=1}^{R^j} p_i^j & m = x \text{ and } i = r \\ 0 & m \in \mathcal{D}\backslash\mathcal{D}_S \end{cases}$$

Therefore,

$$\begin{aligned} U_A(p, q^*) - U(p', q^*) &= \sum_{m\in\mathcal{D}\backslash\mathcal{D}_S} \big(\sum_{i=1}^{R^m} p_i^m\big)(iS^m) \\ &\quad -\big(\sum_{m\in\mathcal{D}\backslash\mathcal{D}_S}\sum_{i=1}^{R^m} p_i^m\big)\big(-\sum_{j=1}^{R^x} q_j^x(rS^x + jC^sS^x)\mathbb{1}_{r>R^x-j} \\ &\quad +\epsilon F^x\sum_{j=1}^{R^x} q_j^x(j)\mathbb{1}_{r\leq R^x-j} + \sum_{j=0}^{R^x} q_j^x(rS^x)\mathbb{1}_{r\leq R^x-j} - \epsilon F^x\sum_{j=1}^{R^x} q_j^x(j)\big) \\ &= \sum_{m\in\mathcal{D}\backslash\mathcal{D}_S} \big(\sum_{i=1}^{R^m} p_i^m\big)\big(iS^m - \kappa\big) \\ &\leq \sum_{m\in\mathcal{D}\backslash\mathcal{D}_S} \big(\sum_{i=1}^{R^m} p_i^m\big)\big(\max_{D_j\in\mathcal{D}\backslash\mathcal{D}_S}(S^jR^j) - \kappa\big) < 0 \end{aligned}$$

Therefore, when $\max_{D_j\in\mathcal{D}\backslash\mathcal{D}_S}(S^jR^j) < \kappa$ and the TPA chooses to focus on $\mathcal{D}_S$, the CP is better off focusing on this set too.

Finally, the necessary conditions for the solution to be a NE are:

$$\begin{cases} |\mathcal{D}_S| - \sum_{m\in\mathcal{D}_S}\frac{G^m}{E^m} + \max_i\big(G^i - \frac{(1-C^tS^i\tau^i)E^i}{1+C^sS^i\tau^i}\big)\sum_{m\in\mathcal{D}_S}\frac{1}{E^m} \\ \qquad < P < |\mathcal{D}_S| - \sum_{m\in\mathcal{D}_S}\frac{G^m}{E^m} + \min_i G^i\sum_{m\in\mathcal{D}_S}\frac{1}{E^m} \\ |\mathcal{D}_S| - \min_i S^i\sum_{m\in\mathcal{D}_S}\frac{1}{H^m(2S^m+\phi^m_{R^m})} - \sum_{m\in\mathcal{D}_S}\frac{(1-W^m)(2S^m+\phi^m_{R^m})-S^m}{H^m(2S^m+\phi^m_{R^m})} \\ < Q < |\mathcal{D}_S| + \min_i(S^i+\phi^i_{R^i})\sum_{m\in\mathcal{D}_S}\frac{1}{H^m(2S^m+\phi^m_{R^m})} - \sum_{m\in\mathcal{D}_S}\frac{(1-W^m)(2S^m+\phi^m_{R^m})-S^m}{H^m(2S^m+\phi^m_{R^m})} \end{cases}$$

□

An immediate consequence of Theorem B.3 is that the CP has no incentive to dishonor the agreement with the client for any data $D_j \in \mathcal{D}\backslash\mathcal{D}_S$ under the condition that $\max_{D_m \in \mathcal{D}\backslash\mathcal{D}_S} S^m R^m < \kappa$.

B.4.3 Stackelberg Game

In this section, we consider multiple backup copies for each data and analyze the case where the TPA will choose his strategy first. Then, the CP, informed by the TPA's choice, chooses his strategy. This type of interactions between the two players falls under the Stackelberg game category [OR94]. In this type of games, we have a leader and a follower. The objective of the leader in the game is to anticipate the follower's response and to choose his strategy accordingly.

We will the study the interactions between the CP and the TPA when the choice of a strategy for data D_i is independent of the strategy for data D_j, $\forall j \neq i$.

The utility of the CP can be written as follows:

$$U_A(p,q) = \sum_{m=1}^{N}\sum_{i=1}^{R^m} p_i^m \Big(-\sum_{j=1}^{R^m} q_j^m (iS^m + jC^s S^m)\mathbb{1}_{i>R^m-j} + \epsilon F^m \sum_{j=1}^{R^m} q_j^m(j)\mathbb{1}_{i\leq R^m-j} + \sum_{j=0}^{R^m} q_j^m (iS^m)\mathbb{1}_{i\leq R^m-j}\Big) + \epsilon \sum_{m=1}^{N} p_0^m \sum_{j=1}^{R^m} q_j^m (jF^m)$$

The TPA will try to choose a strategy that will deter the CP from dishonoring his backup commitments on any data D_m. This translates to having $p_i^m = 0$, $\forall m \in \{1, ..., N\}$ $\forall i \in \{1, ..., R^m\}$.

In this case, analyzing the CP's utility function, we should have:

$$-\sum_{j=1}^{R^m} q_j^m (iS^m + jC^s S^m)\mathbb{1}_{i>R^m-j} + \epsilon F^m \sum_{j=1}^{R^m} q_j^m(j)\mathbb{1}_{i\leq R^m-j} + \sum_{j=0}^{R^m} q_j^m (iS^m)\mathbb{1}_{i\leq R^m-j} \leq 0 \quad \text{(B.2)}$$

In fact, we can relax the inequality to only require that Equation B.2 equals 0. Therefore, the Stackelberg equilibrium can be expressed as follows, $\forall m \in \{1, ..., N\}$:

$$\begin{cases} p_0^{m*} = 1 \\ p_i^{m*} = 0 \quad \forall i \in \{1, ..., R^m\} \end{cases}$$

$$\begin{cases} q_0^{m*} = \frac{1}{2}\Big(1 + \frac{C^s R^m}{2 + C^s R^m + 2\sum_{i=1}^{R^m-1} \frac{(C^s R^m S^m + 2S^m + i\epsilon F^m)}{\phi_i^m} \prod_{j=1}^{i-1}(1 + \frac{2S^m}{\phi_j^m})}\Big) \\ q_i^{m*} = \frac{(2q_0^{m*} - 1)S^m}{\phi_i^m} \prod_{j=1}^{i-1}(1 + \frac{2S^m}{\phi_j^m}) \qquad \forall i \in \{1, ..., R^m - 1\} \\ q_{R^m}^{m*} = \frac{1}{1 + C^s R^m}\Big(q_0^{m*} + \sum_{i=1}^{R^m-1} \frac{(2q_0^{m*} - 1)(S^m + i\epsilon F^m)}{\phi_i^m} \prod_{j=1}^{i-1}(1 + \frac{2S^m}{\phi_j^m})\Big) \end{cases}$$

We notice that $q_0^{m*} > 0.5$, $\forall m \in \{1, ..., N\}$. Therefore, in order to achieve his objective, the TPA needs the CP to believe that he will more probably not attempt to check the existence of any copy of the data. This can be interpreted as if the TPA will trust the CP to respect the requirements of the backup process for data D_m. However, the TPA does not take the option of checking the existence of at least one copy of the data off the table, even though the probability of such event is lower than the probability of not checking the existence of any copy at all.

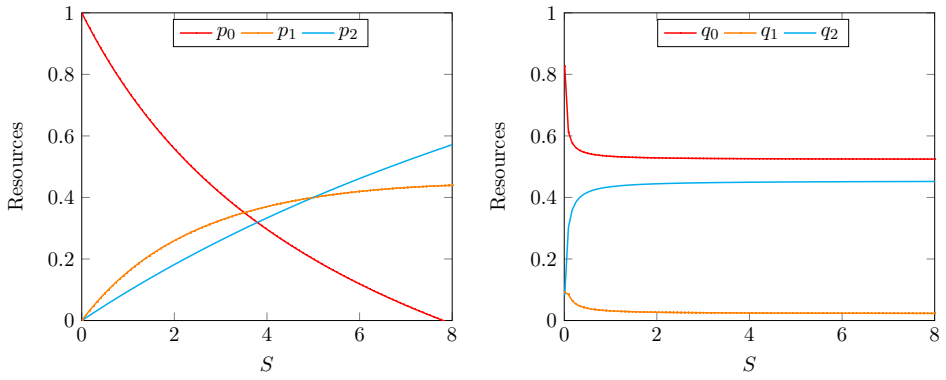
B.5 Numerical Analysis

In this section, unless stated otherwise, we consider the baseline parameters $C^s = 0.1$, $C^t = 0.1$, and $\epsilon = 0.1$ and we analyze players' strategies w.r.t. increasing values of S. Let $\mu = (R, F, C^s, C^t, \epsilon)$.

B.5.1 Independent Strategies One-shot Game

The strategies of the CP and the TPA for data D_j do not depend on their strategies for data D_i, $\forall i \neq j$. Since we focus on each data D_m independently, we will drop the index m in this section. Let $D \in \mathcal{D}$ s.t. $F = 0.5$ and $R = 2$. We will study the impact of the CP storage cost S of the data D on both players' strategies.

In Fig. B.1a, the CP's strategy p_0^* decreases w.r.t. increasing values of S whereas p_2^* increases. We note that there exists a value of S under which $p_1^* > p_2^*$. From Fig. B.1b, when the storage cost S of data D is small, the TPA will privilege not to check the existence of any backup copies. When S increases, the TPA's strategy q_0^* quickly decreases before stabilizing on a value greater than 0.5. On the other hand, q_2^* increases quickly before stabilizing. For small values of S, we observe a peak for q_1^* before decreasing and eventually stabilizing on a value less than 0.5. For small values of C^s and ϵ, when S increases, the values of q_0^* and q_2^* stabilize around 0.5. In this case, it is as if the choice of the TPA is restricted to whether to check all backup copies or none at all. The TPA does not have any interest in checking the existence of a number of backup copies less than the number required in the contract between the TPA and the client. In this case, the cost $C^s S$ paid by



(A) (B)

FIGURE B.1: IS game: $\mu = (2, 0.5, 0.1, 0.1, 0.1)$

the TPA is relatively small when the CP passes the verification test. As a result, the TPA prefers to check the existence of all backup copies stored on the CP servers.

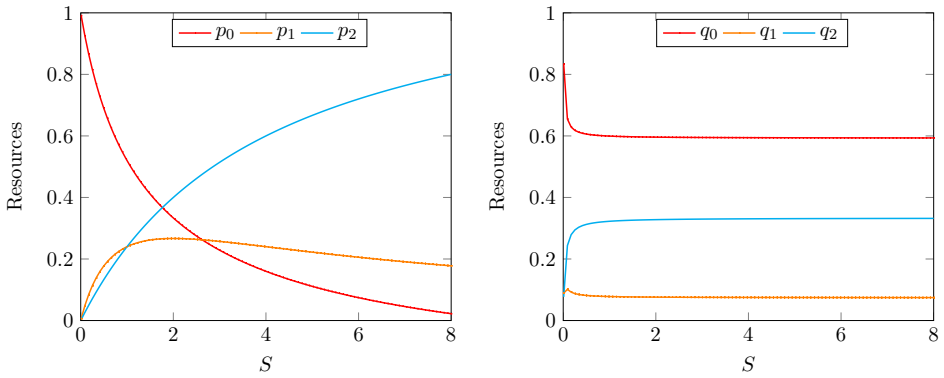
Impact of C^s. From Fig. B.2b, an increase in the cost of executing the verification query C^sS will have no significant impact on the pattern of change of the TPA's NE strategy w.r.t. to S. However, the stable values of q^* for large values of S change. In particular, they increase for q_0^* and q_1^* and decrease for q_2^*. The TPA increases the frequency of checking one copy instead of two copies, since checking either an honest CP or a CP that passes the verification test will entail a higher cost C^sS for the TPA. From Fig. B.2a, with respect to a low cost C^s, a greater cost $C^{s\prime}$ will impact the CP's NE strategy by increasing its rate of change without affecting its pattern of change w.r.t. S.

Impact of C^t. The TPA's strategy at the NE is independent of C^t. In Fig. B.2c, as with greater values of C^s, a similar change is observed in the CP's NE strategy when increasing the cost of processing the verification query for the TPA C^tS. However, in this case, the CP's strategy changes more quickly w.r.t. S.

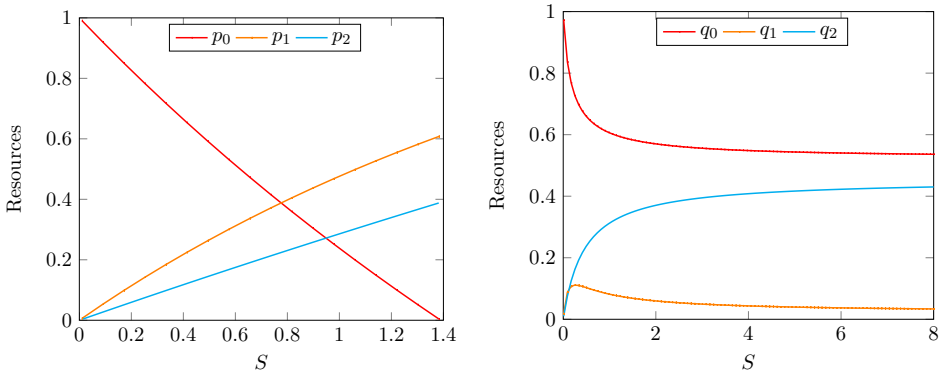
Impact of ϵ. In Fig. B.2d, when ϵ increases, the TPA's NE strategy rate of change decreases. For large values of S, we notice an increase of the stable values for q_0^* and q_1^* and a decrease for q_2^*. The TPA's NE reflects his belief that the CP will more likely behave honestly given the increased incentive given to him when behaving as such. However, this incentive is given to the CP when the TPA fails to detect a malicious act by the CP and therefore, it does not completely prevent such scenario.

Impact of F. In Fig. B.2e, when F increases, the rate of change of the CP's NE strategy decreases. For small values of S/F, the CP has no interest in deleting any copies of the data since such action will entail a small payoff and exposes the CP to the risk of being detected by the TPA.

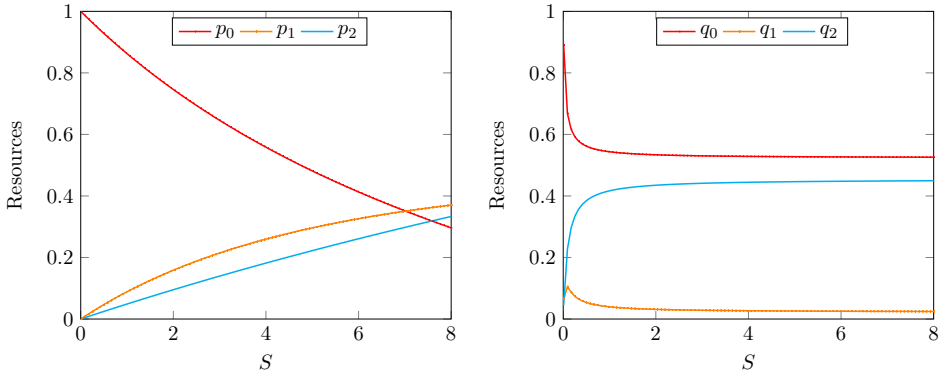
Fig. B.3 depicts an example with the baseline parameters with $R = 3$. We notice that in general, when the number of backup copies increases ($R > 2$), there exists a value $S = \frac{\epsilon F}{4-C^s}$



(A) $\mu = (2, 0.5, 0.5, 0.1, 0.1)$ (B) $\mu = (2, 0.5, 0.5, 0.1, 0.1)$



(C) $\mu = (2, 0.5, 0.1, 0.5, 0.1)$ (D) $\mu = (2, 0.5, 0.1, 0.1, 1)$

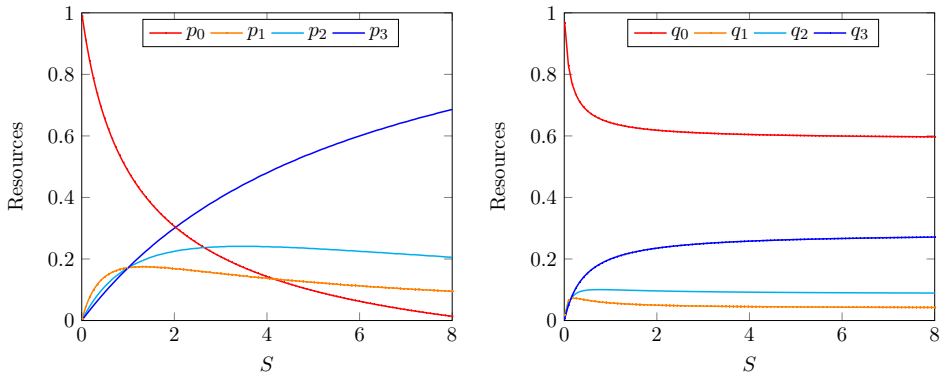


(E) $\mu = (2, 1, 0.1, 0.1, 0.1)$ (F) $\mu = (2, 1, 0.1, 0.1, 0.1)$

FIGURE B.2: IS game

under which we have the TPA's NE strategy $q_i^* > q_{i+1}^*$, $\forall i = \{1, ..., R-1\}$. Similarly for the CP, for $S > F/C^s$, we have $p_{i+1}^* > p_i^*$ $\forall i = \{1, ..., R-1\}$.

In our model, we do not take into account the impact of detecting a dishonest CP on his reputation, which can lead in practice in decreasing his future profits. Such assumption has an impact on the NE strategies of both players and in particular on the CP's frequency of not keeping all the backup copies of the data.



(A) (B)

FIGURE B.3: IS game: $\mu = (3, 0.5, 0.5, 0.1, 1)$

B.5.2 Correlated Strategies One-shot Game

For presentation reasons, we consider only two data items D_x and D_z and plot the strategies of the TPA and the CP when targeting at least one backup copy of the data. Let $R_x = 2$, $F_x = 1$, $R_z = 3$, $F_z = 2$, and $S^z = 1$. We will analyze the strategies of the TPA and the CP w.r.t. the importance S^x of the data D_x. Table B.2 exhibits the different values of parameters used in this section.

TABLE B.2: Values of parameters

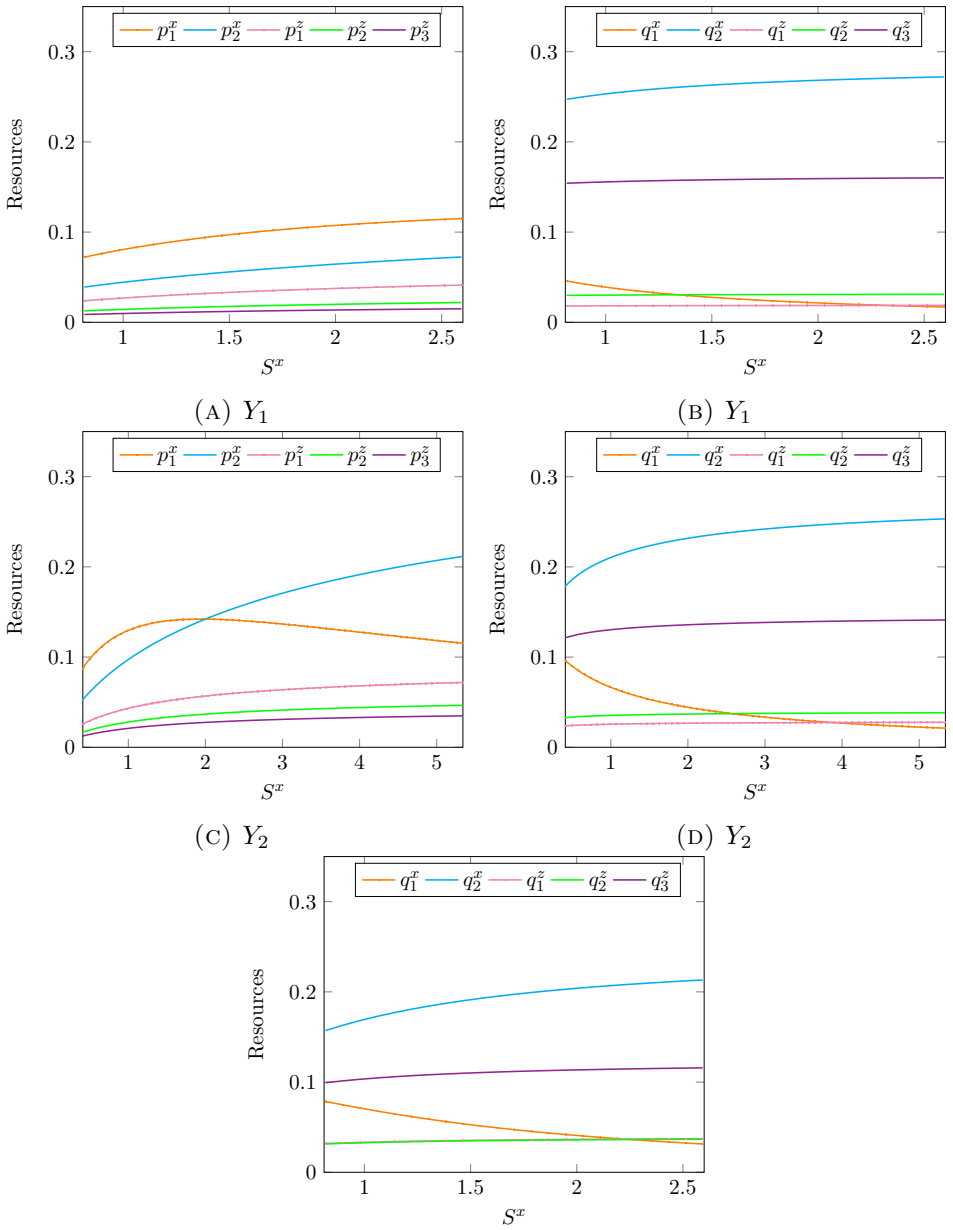
	C^s	C^t	ϵ
Y_1	0.1	0.1	0.1
Y_2	0.5	0.1	0.1
Y_3	0.1	0.1	1

B.5.2.1 Single Targets

In a CSST game, each player can target one type of data and execute one action related to that data at each instance of the game.

In Fig. B.4a, w.r.t. increasing values of S^x, we notice that p_1^{x*} and p_2^{x*} increase. As the importance of the data to the CP increases, he will be more tempted not to respect data backup requirements to free additional space on his servers. For data D_z (Fig. B.4a), p_i^{z*} increases, $\forall i \in \{1, ..., 3\}$. On the other hand, we notice that there exists a value $S' \approx 1.6$ s.t. $\sum_{i=0}^{2} p_i^{x*} < \sum_{i=0}^{3} p_i^{z*}$, $\forall S > S'$. The CP will therefore focus more on data D_z even though this does not necessarily translates in removing any backup copy of D_z at each instance of the game.

For the TPA (Fig. B.4b), we notice that q_1^{x*} decreases and q_2^{x*} increases. In this case, the TPA privileges checking the maximum number of backup copies given the high value of



(A) Y_1 (B) Y_1

(C) Y_2 (D) Y_2

(E) Y_3

FIGURE B.4: CSST game

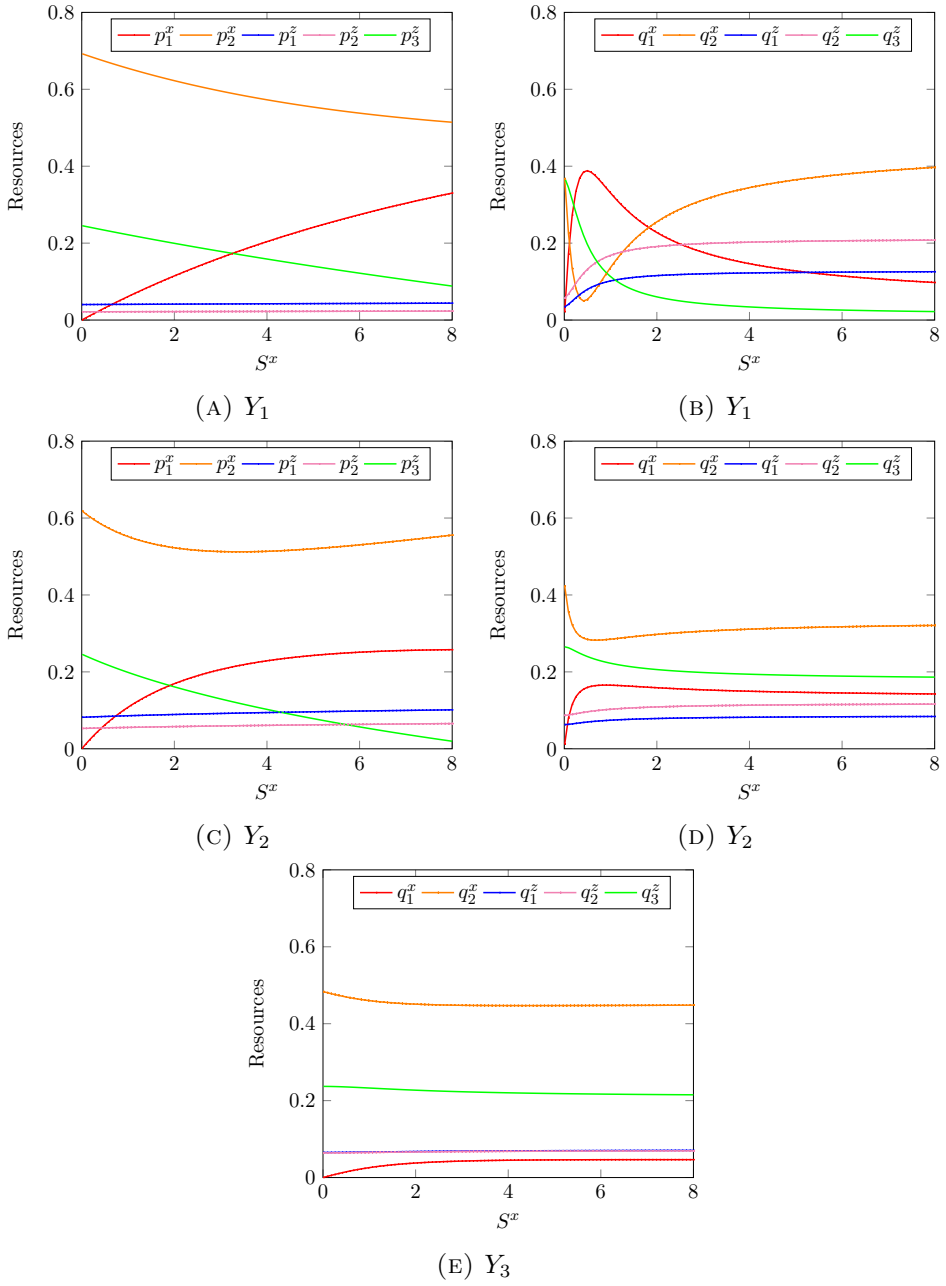
the data S^x to the CP, which is correlated with an increase in the likelihood that the CP dishonors the backup agreement for D_x.

Comparing Fig. B.4b and Fig. B.4e, we notice that higher values of ϵ do not affect the pattern of change of the TPA's NE strategy for data D_x.

Finally, for greater values of C^s, the interval of values of S^x in which a NE exists widens (Fig. B.4c). We notice that there exists a value S_0^x under which $p_1^{x*} > p_2^{x*}$. In addition, when S increases, the CP will allocate more resources to remove all backup copies of D_x.

B.5.2.2 Multiple Targets

In a CSMT game, each player can target multiple types of data at each instance of the game. We suppose the resource constraints $P = 1$ and $Q = 0.85$. In this case, we find that the attractive set $T_S = \{D_x, D_z\}$.



(A) Y_1 (B) Y_1

(C) Y_2 (D) Y_2

(E) Y_3

FIGURE B.5: CSMT game

In Fig. B.5a, interestingly, when S_x increases, p_1^{x*} increases while p_2^{x*} and p_3^{z*} decrease. For higher values of S_x, the CP strategically manages his resources in order to increase his

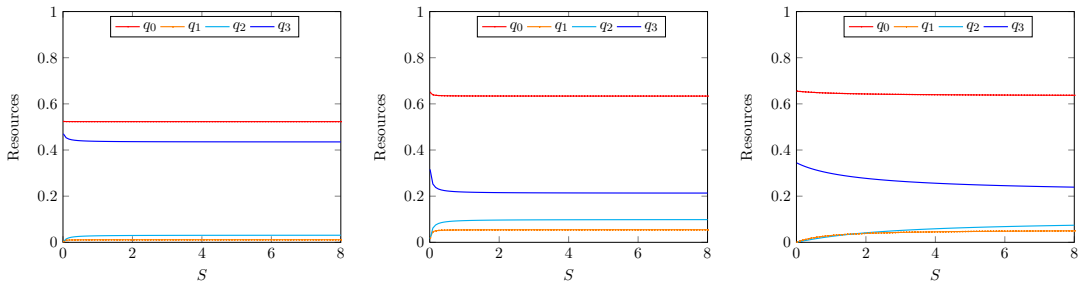
payoff by keeping only one copy of the data D_x, while at the same time reducing the risk of being caught by the TPA. On the other hand, the TPA responds by allocating more resources to verify the existence of two backup copies of D_x.

When C^s increases (Fig. B.5c and Fig. B.5d), the rate of change of both players' strategies increases. The TPA's NE strategy quickly stabilizes to certain values. Compared to Fig. B.5b, the TPA reduces the allocated resources to verify the existence of all the backup copies of data D_x, since the cost of verification if the CP was acting honestly is higher. However, the TPA increases the resources to verify the existence of one copy of D_x and at least one copy of D_z.

While increasing the value of ϵ does not impact the CP's strategy at the NE, it directly affects the TPA's NE strategy (Fig. B.5e). In particular, we notice that the TPA will focus on verifying the existence of all the backup copies of D_x and D_z.

B.5.3 Stackelberg Game

In this section, the TPA chooses his strategy first. Then, informed by the TPA's choice, the CP chooses his strategy. We consider the case where the strategy for a data D_i is independent of the strategies for other data D_j, $\forall j \neq i$. In Section B.4.3, we proved that in this case, a Stackelberg equilibrium (SE) of the game exists. The TPA's strategy at the SE discourages the CP from dishonoring the data backup agreement with the client.



(A) $\mu = (3, 1, 0.1, 0.1, 0.1)$ (B) $\mu = (3, 1, 0.9, 0.1, 0.1)$ (C) $\mu = (3, 1, 0.9, 0.1, 5)$

FIGURE B.6: Stackelberg game

From Fig. B.6a, w.r.t. increasing values of S, we find that q_0^* and q_R^* decrease while q_i^* increases, $\forall i \in \{1, ..., R\}$. However, the TPA's strategy quickly stabilizes afterwards. We note that we always have $q_0^* > 0.5$. When C^s increases (Fig. B.6b), for large values of S, the stable values of q_0^* increases, q_R^* decreases, and q_i^* increases, $\forall i \in \{1, ..., R\}$. Given the higher cost $C^s S$, the TPA focuses more on checking the existence of $k < R$ copies instead of R copies. When the incentive ϵ increases (Fig. B.6c), the rate of change of the TPA's strategy decreases and it stabilizes slower w.r.t. S. In this case, for smaller values of ϵ, the TPA's incentive is not sufficient to deter the CP from dishonoring the requirements of the backup process agreed on with the client, which forces the TPA to adopt an aggressive strategy even for small values of S.

B.6 Case Study

In this section, we present a practical application of our model through a concrete scenario based on storage Cloud.

B.6.1 Parameter Evaluation

Applying a theoretical model in a realistic scenario means being able to evaluate each of the parameters in the model. In this section, we provide a guideline of how to evaluate the model parameters by the different players in the game. We introduce additional intermediate parameters, which will be used to deduce the parameters used in the model. $\forall m \in \{1, ..N\}$, let T^m represent the size of data D_m, measured in bytes. Given a data integrity verification protocol, let b be the size ratio of the data being checked (e.g. $b = 0.1$ if 10% is the proportion of data D that is checked by the TPA), and t_{CP} and t_{TPA} the execution times of the protocol on the CP and the TPA sides respectively.

The input parameters of the model are the number of data items N, the data set $\mathcal{D}$, and for each data D_m, its size T^m and the number of backup copies R^m. The value of R^m is assumed to be known as it can be part of the SLA with the cloud provider. From these values, we will first evaluate S^m, which is the financial value corresponding to the storage of one copy of data D_m by the CP. Based on [CS11], the storage costs can be precisely deduced from T^m. If we denote α the storage cost per bit in a given fixed period of time, the value of which can be obtained from [CS11], then we get $S^m = \alpha T^m$. Note that S^m could also take into account the estimated financial loss due to the amount of money that the CP may have earned with the omission of one copy of D_m.

The next step consists of evaluating F^m, which is the financial value corresponding to the importance of one copy of data D_m from the client's perspective. In general, F^m is correlated to R^m. In fact, the client is more likely to ask for additional backups copies of his most important data, as it could happen in the context of Cloud archiving, for instance. The very nature of this parameter makes risk assessment methods, such as EBIOS [ANS10], one of the relevant methods to obtain the necessary information allowing the client to evaluate it. Financial cost due to data loss may be deduced from business knowledge, relying on criteria such as the loss of competitive advantage, or the difficulty to reproduce the data. Based on this assessment, the value F^m of one copy can be considered to be equal to the estimated value of the data divided by the number of copies R^m.

The verification cost and the cost of executing the verification query are given by $C^t S^i$ and $C^s S^i$ respectively. For a given data D_m of size T^m, t_{CP} and t_{TPA} can be measured from an implementation of the data integrity verification protocol, and the size ratio b to be checked in order to obtain a reliable proof, which can be deduced from the reference paper describing the protocol. From the values of t_{CP} and t_{TPA}, the number of CPU cycles can be estimated given the host characteristics. From [CS11], knowing the number of CPU

cycles, the execution costs per bit for the CP and the TPA C_{bit}^{CP} and C_{bit}^{TPA} respectively can be deduced. From the available information so far, we can write the following equation $C^s S^m = bT_m C_{bit}^{CP}$. Since $S^m = \alpha T^m$, we have $C^s = \frac{bC_{bit}^{CP}}{\alpha}$.

In the case study, it is more interesting to assess the impact of the parameter ϵ on the players' strategies rather than define a method to evaluate it. ϵF^m is used as a reward for the CP for acting honestly. For example, it may refer to gains in terms of the reputation of the CP that is being highlighted by the TPA for the good behavior. The objective for the TPA would therefore be to find the optimal value of ϵ that decreases the probability that the CP acts dishonestly at the NE.

B.6.2 Numerical Example

We consider the case where $N = 3$ data items are outsourced. The characteristics of data D_1, D_2, D_3 are defined as in Table B.3.

Table B.3: Data characteristics

	$\boldsymbol{T^m}$ (in GB)	$\boldsymbol{R^m}$	$\boldsymbol{F^m}$ (in \$)
$\boldsymbol{D_1}$	0.01	4	200
$\boldsymbol{D_2}$	5	5	300
$\boldsymbol{D_3}$	200	2	130

First, we compute the value of S^m for each data D_m. From [CS11], we know that the storage cost for a CP can be estimated between about 100 picocent/bit and 300 picocent/bit per year (1 picocent = 10^{-14} \$). In this case study, we consider a time period of one year, and an average storage value of 200 picocent/bit. Therefore, we find $S^1 = 0.000016$ \$, $S^2 = 0.08$ \$, and $S^3 = 3.2$ \$.

For the evaluation of C^s and C^t, the verification scheme we implemented is based on the open source proof of retrievability project by Zachary Peterson, and corresponds to the basic POR scheme from [JK07], where the number of checked sentinel blocks represent $b = 1\%$ of the data file size. We used a Linux Virtual Machine running on a laptop with an i7 Intel Core processor, with 2.3 GHz clock frequency, and 8 GB of RAM. For the biggest data D_3, we measured $t_{TPA} = 1.51s$ and $t_{CP} = 0.27s$, the difference being due to the fact that there is no specific processing on the CP side besides giving the correct sentinel blocks in this scheme. Based on the values in [CS11], the CP cycle cost can be estimated at 2 picocent/cycle, while the TPA cycle cost should rather be around 20 picocent/cycle. Therefore $C^s S^3 = 1.24.10^{-5}$ \$, and $C^t S^3 = 7.0.10^{-4}$ \$, which gives us $C^s = 3.88.10^{-6}$ and $C^t = 2.19.10^{-4}$.

We assess the impact of ϵ on the Nash Equilibrium strategies in the case of the CSMT game, where $P = 1$ and $Q = 0.75$. Since the primary objective consists of finding the optimal checking strategy, we focus in this section on the behavior of the TPA. Table B.4

depicts the probability of checking the existence of at least one backup copy of each data for different values of ϵ. We notice that when the value of ϵ increases, the TPA will spend less resources on checking the existence of at least one backup copy of data D_2. This can be explained by the fact that D_2 has the highest value F^2. For the CP, a higher ϵ means receiving a substantial reward when he acts honestly. In this case, the TPA will therefore not waste too many resources on checking the existence of backup copies of this data, as the incentive is assumed to be high enough for the CP to behave honestly. However, if we multiply the size of data D_2 by 10 (therefore $S^2 = 0.8$ \$), for $\epsilon = 0.01$, we notice that the TPA will spend twice as much resources to check the existence of at least one backup copy of D_2 w.r.t. the results for $\epsilon = 0.01$ in Table B.4. In this case, for data D_2, the TPA anticipates that the reward will be less effective in preventing the CP from acting dishonestly.

TABLE B.4: Probability of checking at least one backup copy of the data at the NE

	$\epsilon = 0.01$	$\epsilon = 0.1$
D_1	0.137	0.161
D_2	0.117	0.091
D_3	0.496	0.498

An example of finding an optimal value for ϵ would be to find the minimum incentive that guarantees that the TPA will not need to use more than 10% of his resources for checking the existence of at least one backup copy of data D_2 at the NE. Running the optimization in this case study, we find the value of 0.028 for ϵ.

It is worth mentioning that the difference between the values F of the data items and the other parameters in this case study is substantial, which could raise concerns about the influence of the values of F over the other parameters. However, if we multiply the size of data D_3 by 10, bringing S^3 to 32 \$, which is not negligible compared to F^3, we do not notice a big difference in the TPA's NE strategy w.r.t. the results in Table B.4.

B.7 Conclusion

In this chapter, we analyzed the problem of verifying data availability in the case of data outsourced to a cloud provider. We formulated the problem between the CP and the TPA as a non-cooperative game. The TPA's objective is to detect any deviation from the agreement signed between the CP and the client by checking the existence of the required number of backup copies of each type of data on the CP's servers. On the other hand, the CP's objective is to increase the storage capacity on his servers, which translates in practice in the existence of a number of copies less than the required number included in the contract with the client. We performed an in-depth analysis of multiple extensions of the simple model in [DKLC14] taking into account the existence of multiple backup copies of each data. In each proposed extension, we identified the optimal verification strategy for the TPA. Finally, we validated our analytical results on a case study.

One of the interesting results that we found relates to the stackelberg game in which we have a leader (the TPA) and a follower (the CP) in the game. This type of games reflects realistic scenarios that we can encounter in real life. Interestingly, our results show that a NE of the game exists and when it is achieved, the CP cannot improve his utility by acting dishonestly. At the NE, it is as if the trust of the TPA in the CP's actions outweigh any belief of a potential misconduct.

The results in this chapter rely on the basic assumption of the rationality of the CP and the TPA, which is a reasonable hypothesis in this case. However, one may argue about the relevance of the different types of parameters introduced in the model and the cost allocations (who will need to pay what). For instance, the signed agreement between the CP and the client can specify that the CP must always take charge of the cost of executing the verification query. While this is a realistic assumption, always taking the burden of this cost by the CP may result in an abusive verification behavior by the TPA. Therefore, in this chapter, we distinguished which player needs to pay that cost according to the detection of a malicious act by the CP. With ϵF, these parameters play the role of incentives and punishments for the CP and allows us to analyze their subsequent effects on his behavior. In addition, the analysis of the different types of games can be used not only to study the behavior of players, but can also be leveraged to help adjust these incentives and punishments to be aligned with the client's interests when negotiating an SLA with a CP.

The model presented in this chapter can be adapted to verify the existence of the required number of backup copies in specific geographical locations as is sometimes specified in an SLA. As future work, we plan to investigate the case where interactions between the CP and the TPA can occur on multiple occasions over time. This type of interactions is particularly interesting if we consider a repeated game setting where we have a number of TPAs, on behalf of multiple clients, verifying the CP's compliance with the signed agreements with the clients. In this case, the result of the interactions between the CP and a client is not limited to that particular client, but extends to impact the behavior of all the other players in the game. For example, we can study how the discovery of an improper act by the CP can affect his reputation and therefore his future payoffs, as clients will be more inclined to change provider. In this case, players' behaviors may change after it has been made public that a CP breached his agreement with a client. Therefore, each short-term gain of the CP must be weighted against the enduring long-term impact on his reputation, which automatically affects his future profits. The public exposure of the behavior of the CP is an important dimension that needs to be taken into account, which can play a decisive role of deterrence to force the CP to fully respect the backup agreements signed with the clients.

Appendix C

Evaluating Initial State Probability Distribution

The attack surface of a network refers to the set of entry points that an attacker can use to compromise the system. In this chapter, we focus on external attackers that have knowledge of the attack surface of a system. We are interested in computing the optimal distribution of attack resources on equipment contributing to the system attack surface. This optimization does not only depend on the attacker profile and his attack preferences, but also on the actions of the defender whose objective is to optimize the distribution of his detection resources on equipment in order to better detect attack attempts. Therefore, the attacker must choose his targets while taking into account the actions of the defender. In this chapter, under the assumption of the rationality of the attacker and the defender, we present a game theoretical approach for optimizing the allocation of attack and defense resources in a network, focusing on intrusion detection in which equipment interdependent vulnerabilities are taken into account. We model the interactions between the attacker and the defender as a two player non-cooperative static game. We pay a particular attention to the evaluation of the model parameters, as they are chosen in order to be naturally derived from information security risk assessment methods and correspond to what a chief information security officer would expect to find. We analyze the game, derive the Nash Equilibrium (NE), and discuss the engineering implications behind the analytical results. In the absence of any observation of the strategy of the defender before choosing an attack strategy, the attacker can only achieve an optimal distribution of his resources when operating at the NE given a best response strategy of the defender. In evaluating the initial state probability distribution β, we are only interested in the attacker's strategy at the NE. In particular, the defender needs not to operate at the NE in practice. However, in computing β, we are interested in the behavior of an attacker that anticipates the potential actions of a rational defender. We start by presenting a theoretical preliminary result for resource constrained network security games.

C.1 Resource Constrained Network Security Games

In this section, we present a new result for a set of security games. We refer to this type of games as *Resource Constrained Network Security* games and define these games as in Definition C.1.

Definition C.1 (RCNS game)**.** *A* Resource Constrained Network Security (RCNS) *game is a non-cooperative two player, static, complete information game with strategies* $\boldsymbol{p} = (p_1, ..., p_n) \in [0,1]^n$ *and* $\boldsymbol{q} = (q_1, ..., q_n) \in [0,1]^n$ *associated to the attacker and the defender respectively. The game features a set* $\mathcal{T}$ *of* n *targets to attack and defend, as well as resource constraints* $\sum_{i\in\mathcal{T}} p_i \leq P \leq 1$ *and* $\sum_{i\in\mathcal{T}} q_i \leq Q \leq 1$ *for the attacker and the defender respectively. The actions of the attacker and the defender on each target* i *of the network are limited to* {*Attack, Not attack*} *and* {*Defend, Not defend*} *respectively. The strategic form of a RCNS game for a target* i *is represented in Table C.1, where the attacker's payoffs* r_i, s_i, t_i, *and* u_i *are nonnegative real numbers and the defender's payoffs* r'_i, s'_i, t'_i, *and* u'_i *are nonpositive real numbers. We assume that* $u_i \leq t_i$, $s'_i \leq u'_i$, $r_i - s_i \leq t_i - u_i$, *and* $r'_i - t'_i \geq s'_i - u'_i$.

Table C.1: Strategic form of the RCNS game for target i

	Defend	Not defend
Attack	r_i, r'_i	t_i, t'_i
Not attack	s_i, s'_i	u_i, u'_i

We suppose that a realistic RCNS game satisfies $u_i \leq t_i$ and $s'_i \leq u'_i$. Moreover, the difference in payoff for the attacker between the Attack/Not attack actions is higher when the defender chooses not to defend, which translates to $r_i - s_i \leq t_i - u_i$. Similarly, on the defender's side, we have $r'_i - t'_i \geq s'_i - u'_i$. Table C.1 presents the strategic form of the game for target i.

Given the strategic form of the game shown in Table C.1, the utility function $U_A(p,q)$ of the attacker can be written as $U_A(p,q) = \sum_{i\in\mathcal{T}} \alpha_i p_i + \sigma_i q_i + \gamma_i p_i q_i + \delta_i$, where $\alpha_i = t_i - u_i$, $\sigma_i = s_i - u_i$, $\gamma_i = r_i - s_i - t_i + u_i$, and $\delta_i = u_i$. Similarly, the utility function $U_D(p,q)$ of the defender can be written as $U_D(p,q) = \sum_{i\in\mathcal{T}} \alpha'_i p_i + \sigma'_i q_i + \gamma'_i p_i q_i + \delta'_i$, where $\alpha'_i = t'_i - u'_i$, $\sigma'_i = s'_i - u'_i$, $\gamma'_i = r'_i - s'_i - t'_i + u'_i$, and $\delta'_i = u'_i$. According to the assumptions in Definition C.1, we have $\alpha_i \geq 0$, $\gamma_i \leq 0$, $\sigma'_i \leq 0$, and $\gamma'_i \geq 0$. We also consider that there exists at least one $j \in \mathcal{T}$ s.t. $\alpha_j + \gamma_j q_j > 0$. Otherwise, from the utility of the attacker, we can notice that he will not have any incentive to attack any target.

Many network security games, such as [CL09], [ZYB12], and [DKLC14], can be formulated as RCNS games. The resource constraints $\sum_{i\in\mathcal{T}} p_i \leq P$ and $\sum_{i\in\mathcal{T}} q_i \leq Q$ represent constraints on players' budgets. The result presented in this section specifies a necessary condition for the existence of a Nash Equilibrium (NE) in this type of games. In particular, we show that at least the attacker has to use all his resources for a NE to exist.

Theorem C.1. *A necessary condition for (p^*, q^*) to be a Nash Equilibrium in a Resource Constrained Network Security game is* $\sum_{i \in \mathcal{T}} p_i^* = P$.

Proof. We consider a generic RCNS game. First, we analyze the case where $\gamma_i = 0$. If $\gamma_i = 0$, then the hypothesis $t_i \geq u_i$ implies $r_i \geq s_i$. In this case, the attacker will always decide to attack node i since the payoff is higher independently from the behavior of the defender. This case being of no interest, we will suppose for the rest of this section that $\gamma_i < 0$. Similarly, we can show that when $\gamma_i' = 0$, the defender always gets a higher payoff by choosing not to defend. In the rest of this section, we suppose $\gamma_i' > 0$.

Let $\mathcal{T}_{S_d}$ be the set of targets on which the defender will allocate defense resources. For example, in a network, the defender monitors a subset of the network nodes to detect intrusions. Similarly, let $\mathcal{T}_{S_a}$ denote the target set that will be attacked by the attacker. In general, we note that $\mathcal{T}_{S_d} \cap \mathcal{T}_{S_a} \neq \varnothing$.

The conditions for the existence of a NE vary according to the hypothesis made on $\sum_{i \in \mathcal{T}} p_i$ and $\sum_{i \in \mathcal{T}} q_i$. In the general case where $\sum_{i \in \mathcal{T}} p_i \leq P$ and $\sum_{i \in \mathcal{T}} q_i \leq Q$, if a NE ($p^*$, q^*) exists, p^* is a best response strategy to the defender strategy and q^* is a best response strategy to the attacker strategy. Since the utility of the attacker is linear with respect to the attacker's strategy p, if a solution to the attacker's optimization problem exists, then an optimal solution at an extreme point of the feasible set defined by $\sum_{i \in \mathcal{T}} p_i \leq P$ exists (when $\sum_{i \in \mathcal{T}} p_i = P$). A similar analysis can be conducted for the case of the defender.

First case: $\sum_{i \in \mathcal{T}} p_i = P$ and $\sum_{i \in \mathcal{T}} q_i = Q$

From the definitions of $\mathcal{T}_{S_a}$ and $\mathcal{T}_{S_d}$, the constraints on the attack and defense resources become $\sum_{i \in \mathcal{T}_{S_a}} p_i = P$ and $\sum_{i \in \mathcal{T}_{S_d}} q_i = Q$. From the Karush-Kuhn-Tucker conditions, there exists $\lambda > 0$ s.t. $\frac{\partial U_A}{\partial p_i} = \lambda$ and $\lambda' > 0$ s.t. $\frac{\partial U_D}{\partial q_i} = \lambda'$. We have $\frac{\partial U_A}{\partial p_i} = \alpha_i + \gamma_i q_i$. Therefore, $\alpha_i + \gamma_i q_i > 0 \Rightarrow q_i < -\frac{\alpha_i}{\gamma_i} \Rightarrow Q < \sum_{i \in \mathcal{T}_{S_d}} \frac{-\alpha_i}{\gamma_i}$. Since $\alpha_i \geq 0$ and $\gamma_i < 0$, we have $\sum_{i \in \mathcal{T}_{S_d}} \frac{-\alpha_i}{\gamma_i} \geq 0$. Similarly, considering $\frac{\partial U_D}{\partial q_i} = \sigma_i' + \gamma_i' p_i$, we have $P > \sum_{i \in \mathcal{T}_{S_a}} \frac{-\sigma_i'}{\gamma_i'}$. Since $\sigma_i' \leq 0$ and $\gamma_i' > 0$, we have $\sum_{i \in \mathcal{T}_{S_a}} \frac{-\sigma_i'}{\gamma_i'} \geq 0$. We have already established that if a NE solution exists, it must exist at least when $\sum_{i \in \mathcal{T}} p_i = P$ and $\sum_{i \in \mathcal{T}} q_i = Q$. Therefore, from the results above, the necessary conditions for the existence of a NE are $Q < \sum_{i \in \mathcal{T}_{S_d}} \frac{-\alpha_i}{\gamma_i}$ and $P > \sum_{i \in \mathcal{T}_{S_a}} \frac{-\sigma_i'}{\gamma_i'}$.

Second case: $\sum_{i \in \mathcal{T}} p_i < P$ and $\sum_{i \in \mathcal{T}} q_i < Q$

TABLE C.2: Set of possible cases

	Conditions
$\sum_{i\in\mathcal{T}} p_i = P$, $\sum_{i\in\mathcal{T}} q_i = Q$	$Q < \sum_{i\in\mathcal{T}_{S_d}} \frac{-\alpha_i}{\gamma_i}$, $P > \sum_{i\in\mathcal{T}_{S_a}} \frac{-\sigma'_i}{\gamma'_i}$
$\sum_{i\in\mathcal{T}} p_i = P$, $\sum_{i\in\mathcal{T}} q_i < Q$	$Q < \sum_{i\in\mathcal{T}_{S_d}} \frac{-\alpha_i}{\gamma_i}$, $P = \sum_{i\in\mathcal{T},\mathcal{T}\neq\mathcal{T}_{S_a}} \frac{-\sigma'_i}{\gamma'_i}$
$\sum_{i\in\mathcal{T}} p_i < P$, $\sum_{i\in\mathcal{T}} q_i \leq Q$	Impossible

We have $\frac{\partial U_A}{\partial p_i} = 0$. Therefore, $q_i = -\frac{\alpha_i}{\gamma_i} \Rightarrow \sum_{i\in\mathcal{T}} q_i = -\sum_{i\in\mathcal{T}} \frac{\alpha_i}{\gamma_i}$. However, from the first case, we have $Q < \sum_{i\in\mathcal{T}_{S_d}} \frac{-\alpha_i}{\gamma_i} \leq \sum_{i\in\mathcal{T}} \frac{-\alpha_i}{\gamma_i} = \sum_{i\in\mathcal{T}} q_i$. Therefore, $Q < \sum_{i\in\mathcal{T}} q_i \Rightarrow$ *contradiction.* As a result, the scenario in which $\sum_{i\in\mathcal{T}} q_i < Q$ and $\sum_{i\in\mathcal{T}} p_i < P$ does not admit a NE.

Table C.2 exhibits the possible scenarios for the existence of a NE with respect to the assumptions about the resources of the attacker and the defender. In particular, a NE cannot be found when $\sum_{i\in\mathcal{T}} q_i < Q$ and $\sum_{i\in\mathcal{T}} p_i < P$. It is actually possible to extend these results for scenarios in which $\sum_{i\in\mathcal{T}} p_i > P$ or $\sum_{i\in\mathcal{T}} q_i > Q$. However, these conditions represent unrealistic cases where the attack (respectively defense) resources exceed the resource constraint. □

C.2 Intrusion Detection Game

As the amount of network communications keeps growing and the complexity of architectures keeps increasing, designing secure networks has become more challenging. One critical aspect of network security is optimizing the distribution of security resources given a limited defense budget. In addition to firewalls, reverse proxies, or application level countermeasures, Intrusion Detection Systems (IDSs) allow network administrators to substantially refine security management by analyzing data flows dynamically. However, analyzing all the traffic in the network can be complex and costly. Therefore, an optimal IDS deployment strategy to maximize the overall probability of detecting attacks is needed.

In general, networks are composed of a set of equipment such as routers, servers, and firewalls. Each of these equipment play a different role to guarantee the overall network functions. Compromising some of these equipment can provide the attacker with relevant information that he can leverage to compromise additional equipment in the network. Therefore, from the point of view of the attacker, some equipment will be more attractive to attack than others. In addition, the interdependencies of equipment vulnerabilities need to be taken into account. For example, accessing a user workstation is generally not very useful for an attacker unless if it allows him to get access to sensitive equipment more easily. Therefore, it is important to take into account such sequence of attacks in realistic approaches, as the actions of an attacker are not limited to independent atomic attacks.

C.2.1 Related Work

A set of related work covers various aspects of game-theoretical IDS optimization. For example, Nguyen et al. [NAB09] formulate the problem as a stochastic security game where node security assets and vulnerabilities are correlated. They establish a model for interdependencies using linear influence networks. In comparison, our model is based on a static game, which allows us to manipulate more complex utility functions in order to remain as realistic as possible. In [OMA+08], Otrok et al. present a mixed strategies model for maximizing the detection probability of an attack split over multiple packets. The model is analyzed and practical guidelines are provided for IDS optimal sampling strategy. In [AB06], Alpcan and Başar present a zero-sum stochastic game in which both players have limited information on the actions taken by the other player, and explore various learning schemes, upon which both players can optimize their strategies. Unlike our model, [OMA+08] and [AB06] do not take into account the interdependencies between various equipment vulnerabilities.

C.2.2 Game Model and Parameters

We consider a heterogeneous network comprised of n interdependent equipment referred to as nodes in the remaining of this section. The network can be represented as a weighted directed graph $G = (\mathcal{T}, \mathcal{R}, \Theta)$, where $\mathcal{T} = \{1, ..., n\}$ is the set of network nodes, and $\mathcal{R}$ is a particular subset of $\mathcal{T}^2$ and referred to as the edges of G. In particular, an edge (i, j) exists between node i and node j if compromising node i makes it easier for the attacker to compromise node j. Finally, a weight $\theta_i^j \in \Theta$, $\theta_i^j \in]0, 1]$, is associated to each edge $(i, j) \in \mathcal{R}$, quantifying the vulnerability dependency from node i to node j. An example of interdependencies between six nodes is shown in Fig. C.1.

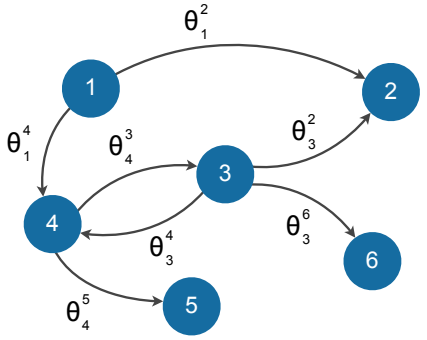


FIGURE C.1: An example of interdependencies θ_i^j between network nodes

We model the intrusion detection problem as a non-cooperative static game with two players, an attacker and a defender. We assume that both players are rational. This assumption holds in many realistic scenarios where the choice of a strategy depends on the payoff it provides to the player. The objective of the attacker is to compromise targets in the

network without being detected, whereas the defender's objective is to distribute monitoring resources on network nodes in order to detect attacks. For each node $i \in \mathcal{T}$, the attacker and the defender actions are limited to *Attack/Not attack* and *Monitor/Not monitor* respectively. The attacker's strategy is represented by a vector $\mathbf{p} = (p_1, ..., p_n) \in [0,1]^n$, where p_i is the probability of targeting node i. Similarly, the defender's strategy is represented by a vector $\mathbf{q} = (q_1, ..., q_n) \in [0,1]^n$, where q_i is the probability of monitoring node i. The resource constraints on the attacker and the defender budgets are P and Q respectively. Therefore, we have $\sum_{i=1}^{n} p_i \leq P$ and $\sum_{i=1}^{n} q_i \leq Q$, where $P \leq 1$ and $Q \leq 1$.

We associate to each node $i \in \mathcal{T}$ the following parameters: W_i, V_i^0, and μ_i. $W_i \geq 0$ represents the importance of services provided by node i to the network. Security assets are assumed to be independent, since the existing correlations between the node security assets may have already been taken into account through a formal risk analysis evaluation process. The intrinsic vulnerability $V_i^0 \in [0,1]$ quantifies local vulnerabilities of services on node i. Finally, $\mu_i \in [0,1]$ represents the probability of detecting an attack on node i considering the current configuration of the defense system.

We assume that the costs of attacking and monitoring a node $i \in \mathcal{T}$ are proportional to the security asset W_i. In addition, these costs are affected by the intrinsic vulnerability V_i^0 on node i. In particular, the cost of attacking node i is inversely proportional to V_i^0, while the cost of monitoring node i is directly proportional to V_i^0. Therefore, the costs to attack and monitor node i are given by $C_a(1 - V_i^0)W_i$ and $C_m V_i^0 W_i$ respectively, where C_a and $C_m \in [0,1]$. Let $C_a^i = C_a(1 - V_i^0)$ and $C_m^i = C_m V_i^0$.

Finally, we introduce a dependency parameter $\psi \in [0,1]$. ψ is used to assess the impact of interdependencies between network nodes in the utilities of the attacker and the defender. For example, $\psi = 0$ is equivalent to the case where interdependencies between network nodes are not taken into account in the model.

C.2.3 Utility Functions

Let $\Gamma^-(i)$ and $\Gamma^+(i)$ refer to the set of predecessors and the set of successors of node i in the network graph G respectively. The effect Δ_i of interdependencies on node i is defined as follows:

$$\Delta_i = \psi \sum_{j \in \Gamma^-(i)} \theta_j^i W_j p_j (1 - \mu_j q_j)$$

Δ_i is the sum of the effect of interdependencies on node i from all its predecessors j that have been attacked (hence the p_j factor) without being detected (hence the $(1 - \mu_j q_j)$ factor) while taking into account the vulnerability dependency $\theta_j^i \in]0,1]$ from node j to node i.

Table C.3 presents the payoff matrix for both players in strategic form for a node $i \in \mathcal{T}$. Its values remain generally close to the payoffs from [CL09] when $\psi = 0$. A successful (i.e. undetected) attack on node i, which happens with probability $1 - \mu_i$, gives the attacker and

TABLE C.3: Payoff matrix in strategic form for node i

	Monitor	Not monitor
Attack	$W_i(1-2\mu_i - C_a(1-V_i^0)) + \Delta_i$, $W_i(2\mu_i - 1 - C_m V_i^0) - \Delta_i$	$W_i(1-C_a(1-V_i^0)) + \Delta_i$, $-W_i - \Delta_i$
Not attack	Δ_i , $-C_m V_i^0 W_i - \Delta_i$	Δ_i , $-\Delta_i$

the defender the payoffs $W_i(1-\mu_i)$ and $-W_i(1-\mu_i)$ respectively. However, if the attack is detected, which happens with probability μ_i, the payoffs for the attacker and the defender are given by $-W_i\mu_i$ and $W_i\mu_i$ respectively. Contrary to [CL09], we take into account the impact of interdependencies between vulnerable network nodes. For example, even though the attacker can choose not to attack node i directly, he can benefit from the impact of attacks on the set of nodes whose compromise can affect his state on node i (e.g. in terms of information or privileges the attacker could decide to make use of).

The utilities U_A and U_D of the attacker and the defender respectively are as follows:

$$\begin{aligned} U_A(p,q) &= \sum_{i=1}^{n} p_i q_i (W_i(1-2\mu_i - C_a^i) + \Delta_i) + p_i(1-q_i)(W_i(1-C_a^i)+\Delta_i) \\ &\quad + (1-p_i)q_i\Delta_i + (1-p_i)(1-q_i)\Delta_i \\ &= \sum_{i=1}^{n} p_i W_i(1-2\mu_i q_i - C_a^i) + \Delta_i \end{aligned}$$

Similarly, we have $U_D(p,q) = \sum_{i=1}^{n} q_i W_i(2\mu_i p_i - C_m^i) - p_i W_i - \Delta_i$.

We note that the intrusion detection game on a network with interdependent nodes is a RCNS game, as defined in Section C.1.

C.2.4 Solving the Game

The intrusion detection game in a network with interdependent nodes defined in Section C.2.3 is a non-cooperative two player static game. An important solution concept for this type of games is the Nash Equilibrium (NE). At the NE, no player has any incentive to deviate from his strategy unilaterally.

C.2.4.1 Nodes Distribution

In our model, W_i refers to the security asset of a node i in the network. The values of the security assets and the impact of the interdependencies between nodes can affect the strategies of the attacker and the defender. In this section, we identify the set $\mathcal{T}_S$ of sensible targets that are attractive to the attacker and needs therefore to be monitored by the

defender. Let $\mathcal{T}_{\mathcal{U}}$ refer to the set of unattractive nodes that will not be the target of attacks. Therefore, we have $\mathcal{T} = \mathcal{T}_{\mathcal{S}} \cup \mathcal{T}_{\mathcal{U}}$. Let $\lambda_i = (1 - C_a^i + \psi \sum_{j \in \Gamma^+(i)} \theta_i^j)$ and $\nu_i = \mu_i(2 + \psi \sum_{j \in \Gamma^+(i)} \theta_i^j)$, $\forall i \in \mathcal{T}$.

Definition C.2 (Sensible target set)**.** *The* sensible target set $\mathcal{T}_S$ *and the* unattractive target set $\mathcal{T}_U$ *are defined as follows:*

$$\begin{cases} W_i\lambda_i > \xi & \forall i \in \mathcal{T}_S \\ W_i\lambda_i < \xi & \forall i \in \mathcal{T}_U \end{cases} \quad \textit{where } \xi = \frac{\sum_{k \in \mathcal{T}_S} \frac{\lambda_k}{\nu_k} - Q}{\sum_{k \in \mathcal{T}_S} \left(\frac{1}{W_k \nu_k} \right)}.$$

The case where $W_i\lambda_i = \xi$ does not need to be taken into account. In fact, this case happens with very low probability. Therefore, should this case happen, and since these values rely on estimations, replacing for instance W_i with a slightly different estimation $W_i + \epsilon$ or $W_i - \epsilon$ would be enough to solve the problem, where $\epsilon > 0$.

For the rest of this section, we suppose that network nodes are numbered according to the following rule: $i < j \Leftrightarrow W_i\lambda_i \geq W_j\lambda_j$.

Lemma C.1. *Given a network comprised of n nodes, $\mathcal{T}_S$ is uniquely determined and consists of n_S nodes with the highest $W_i\lambda_i$ values.*

Proof. We need to prove that $\mathcal{T}_S$ consists of the d highest $W_i\lambda_i$ values, where $d = n_S$ and the cases where $d < n_S$ and $d > n_S$ cannot be achieved.

First, it is easy to prove that if $i \in \mathcal{T}_S$, then $\forall j < i$, $j \in \mathcal{T}_S$. We prove that $d = n_S$ with a proof by contradiction. Let us suppose that $d < n_S$, we have:

$$W_{n_S}\lambda_{n_S} > \frac{\sum_{k=1}^{n_S} \frac{\lambda_k}{\nu_k} - Q}{\sum_{k=1}^{n_S} \left(\frac{1}{W_k \nu_k} \right)} \Rightarrow W_{n_S}\lambda_{n_S} \sum_{k=1}^{n_S} \left(\frac{1}{W_k \nu_k} \right) - \sum_{k=d+1}^{n_S} \frac{\lambda_k}{\nu_k} > \sum_{k=1}^{d} \frac{\lambda_k}{\nu_k} - Q$$

Noticing that $W_{n_S}\lambda_{n_S} \leq W_i\lambda_i, \forall i \leq n_S$ and $d < n_S$ (i.e. $W_{d+1}\lambda_{d+1} \geq W_{n_S}\lambda_{n_S}$), we have:

$$\begin{aligned} W_{d+1}\lambda_{d+1} \sum_{k=1}^{d} \left(\tfrac{1}{W_k \nu_k} \right) &\geq W_{n_S}\lambda_{n_S} \sum_{k=1}^{d} \left(\tfrac{1}{W_k \nu_k} \right) \\ &= W_{n_S}\lambda_{n_S} \sum_{k=1}^{n_S} \left(\tfrac{1}{W_k \nu_k} \right) - W_{n_S}\lambda_{n_S} \sum_{k=d+1}^{n_S} \left(\tfrac{1}{W_k \lambda_k} \tfrac{\lambda_k}{\nu_k} \right) \\ &\geq W_{n_S}\lambda_{n_S} \sum_{k=1}^{n_S} \left(\tfrac{1}{W_k \nu_k} \right) - \sum_{k=d+1}^{n_S} \left(\tfrac{\lambda_k}{\nu_k} \right) > \sum_{k=1}^{d} \tfrac{\lambda_k}{\nu_k} - Q \end{aligned}$$

However, from Definition C.2, we have $W_{d+1}\lambda_{d+1} \leq \dfrac{\sum_{k=1}^{d} \frac{\lambda_k}{\nu_k} - Q}{\sum_{k=1}^{d} \left(\frac{1}{W_k \nu_k}\right)}$. This contradiction shows that it is impossible to have $d < n_S$. Similarly, we can show that it is impossible to have $d > n_S$. Therefore, $d = n_S$ is uniquely determined, and so are $\mathcal{T}_S$ and $\mathcal{T}_U$. □

After identifying the sensible target set $\mathcal{T}_S$, we study the behavior of both players in this scenario.

Theorem C.2. *A rational attacker has no incentive to attack any node $i \in \mathcal{T}_U$.*

Proof. The proof consists of showing that regardless of the defender's strategy q, for any $p \in R_A$ s.t. $\exists i \in \mathcal{T}_U$, $p_i > 0$, we can construct another strategy p' s.t. $p'_i = 0\ \forall i \in \mathcal{T}_U$ and $U_A(p, q) < U_A(p', q)$. If $\mathcal{T}_U = \varnothing$, Theorem C.2 holds. We focus in our proof on the case where $\mathcal{T}_U \neq \varnothing$.

We consider a vector $q^0 = (q_1^0, q_2^0, ..., q_N^0)$ s.t. :

$$q_i^0 = \begin{cases} \dfrac{Q - \sum_{k \in \mathcal{T}_S} \left(\frac{\lambda_k}{\nu_k}\right)}{W_i \nu_i \sum_{k \in \mathcal{T}_S} \left(\frac{1}{\nu_k W_k}\right)} + \dfrac{\lambda_i}{\nu_i} & \forall i \in \mathcal{T}_S \\ 0 & \forall i \in \mathcal{T} - \mathcal{T}_S \end{cases}$$

It holds that $\sum_{i \in \mathcal{T}_S} q_i^0 = Q$, and $q_i^0 \geq 0,\ \forall i$. Let $q = (q_1, ..., q_n)$ denote a defender strategy s.t. $\sum_{i \in \mathcal{T}_S} q_i \leq Q$. By the pigeonhole principle, it holds that $\exists m \in \mathcal{T}_S$ s.t. $q_m \leq q_m^0$.

We consider an attacker strategy $p = (p_1, ..., p_n)$ satisfying $\sum_{i \in \mathcal{T}_U} p_i > 0$, i.e. the attacker attacks at least one target outside the sensible target set $\mathcal{T}_S$ with nonzero probability. We construct another attacker strategy profile p' based on p s.t. :

$$p'_i = \begin{cases} p_i & i \in \mathcal{T}_S \text{ and } i \neq m \\ p_m + \sum_{j \in \mathcal{T}_U} p_j & i = m \\ 0 & i \in \mathcal{T}_U \end{cases}$$

We have $W_i \lambda_i < \xi,\ \forall i \in \mathcal{T}_U$ where ξ is given in Definition C.2. By noticing that $\sum_{i \in \mathcal{T}} \sum_{j \in \Gamma^-(i)} \theta_j^i W_j p_j (1 - \mu_j q_j) = \sum_{i \in \mathcal{T}} W_i p_i (1 - \mu_i q_i) \sum_{j \in \Gamma^+(i)} \theta_i^j$, we get $U_A(p) = \sum_{i \in \mathcal{T}} p_i W_i (\lambda_i - \nu_i q_i)$.

Therefore:

$$
\begin{aligned}
U_A(p) - U_A(p') &= \sum_{i\in\mathcal{T}} p_i W_i(\lambda_i - \nu_i q_i) - \sum_{i\in\mathcal{T}} p'_i W_i(\lambda_i - \nu_i q_i) \\
&= \sum_{i\in\mathcal{T}} p_i W_i(\lambda_i - \nu_i q_i) - \sum_{i\in\mathcal{T}_S, i\neq m} p_i W_i(\lambda_i - \nu_i q_i) - \Big(p_m + \sum_{i\in\mathcal{T}-\mathcal{T}_S} p_i\Big) W_m(\lambda_m - \nu_m q_m) \\
&= \sum_{i\in\mathcal{T}-\mathcal{T}_S} p_i W_i(\lambda_i - \nu_i q_i) - \sum_{i\in\mathcal{T}-\mathcal{T}_S} p_i W_m(\lambda_m - \nu_m q_m) \\
&\leq \sum_{i\in\mathcal{T}-\mathcal{T}_S} p_i W_i(\lambda_i - \nu_i q_i) - \sum_{i\in\mathcal{T}-\mathcal{T}_S} p_i W_m(\lambda_m - \nu_m q^0_m) \\
&= \sum_{i\in\mathcal{T}-\mathcal{T}_S} p_i W_i(\lambda_i - \nu_i q_i) - \sum_{i\in\mathcal{T}-\mathcal{T}_S} p_i \left(\frac{\sum\limits_{k\in\mathcal{T}_S} \left(\frac{\lambda_k}{\nu_k}\right) - Q}{\sum\limits_{k\in\mathcal{T}_S} \left(\frac{1}{\nu_k W_k}\right)} \right) \\
&\leq \sum_{i\in\mathcal{T}-\mathcal{T}_S} p_i W_i \lambda_i - \sum_{i\in\mathcal{T}-\mathcal{T}_S} p_i \left(\frac{\sum\limits_{k\in\mathcal{T}_S} \left(\frac{\lambda_k}{\nu_k}\right) - Q}{\sum\limits_{k\in\mathcal{T}_S} \left(\frac{1}{\nu_k W_k}\right)} \right) \\
&= \sum_{i\in\mathcal{T}_U} p_i \left[W_i \lambda_i - \left(\frac{\sum\limits_{k\in\mathcal{T}_S} \left(\frac{\lambda_k}{\nu_k}\right) - Q}{\sum\limits_{k\in\mathcal{T}_S} \left(\frac{1}{\nu_k W_k}\right)} \right) \right] < 0
\end{aligned}
$$

Therefore, $U_A(p,q) < U_A(p',q)$ and the attacker is always better off attacking only the nodes in the sensible target set $\mathcal{T}_S$. □

Theorem C.2 shows that the attacker only needs to attack nodes that belong to $\mathcal{T}_S$ in order to maximize his utility. Therefore, the defender has no incentive to monitor nodes that do not belong to $\mathcal{T}_S$. As a consequence, valuable defense resources would be wasted by monitoring nodes in $\mathcal{T}_U$. Therefore, a rational defender only needs to monitor nodes in $\mathcal{T}_S$.

C.2.4.2 NE Analysis

Let p^* and q^* be the strategies of the attacker and the defender at the NE respectively. We will study the NE depending whether both players use all their available resources.

Theorem C.3. *Under the assumption that* $\sum_{i\in\mathcal{T}} p^*_i = P$ *and* $\sum_{i\in\mathcal{T}} q^*_i = Q$*, a NE exists and is given by:*

$$p_i^* = \begin{cases} \dfrac{P - \sum\limits_{k\in\mathcal{T}_S} \dfrac{C_m^k}{\nu_k}}{W_i\nu_i \sum\limits_{k\in\mathcal{T}_S}\left(\dfrac{1}{W_k\nu_k}\right)} + \dfrac{C_m^i}{\nu_i} & \forall i \in \mathcal{T}_S \\ 0 & \forall i \in \mathcal{T}_U \end{cases}$$

$$q_i^* = \begin{cases} \dfrac{Q - \sum\limits_{k\in\mathcal{T}_S}\left(\dfrac{\lambda_k}{\nu_k}\right)}{W_i\nu_i \sum\limits_{k\in\mathcal{T}_S}\left(\dfrac{1}{W_k\nu_k}\right)} + \dfrac{\lambda_i}{\nu_i} & \forall i \in \mathcal{T}_S \\ 0 & \forall i \in \mathcal{T}_U \end{cases}$$

In this case, the attacker/defender uses all his resources to attack/defend the network. The game can be seen as a resource allocation problem in which each player's objective is to maximize his/her utility given the action of the other player.

The necessary conditions for the obtained result to be a NE are:

$$\begin{cases} W_i(2\mu_i p_i^* - C_m^i) + \psi W_i \mu_i p_i^* \sum\limits_{j\in\Gamma^+(i)} \theta_i^j \geq 0 \\ W_i(1 - 2\mu_i q_i^* - C_a^i) + \psi W_i(1 - \mu_i q_i^*) \sum\limits_{j\in\Gamma^+(i)} \theta_i^j \geq 0 \end{cases} \Rightarrow \begin{cases} P \geq \sum\limits_{i\in\mathcal{T}_\mathcal{S}} \left(\dfrac{C_m^i}{\nu_i}\right) \\ Q \leq \sum\limits_{i\in\mathcal{T}_\mathcal{S}} \left(\dfrac{\lambda_i}{\nu_i}\right) \end{cases}$$

In this case, the attacker and the defender focus on attacking and monitoring a subset $\mathcal{T}_\mathcal{S}$ of nodes in the network. These nodes yield the maximum payoff for the attacker and therefore need to be monitored.

Theorem C.4. *A NE does not exists under the assumption that* $\sum\limits_{i\in\mathcal{T}} p_i^* < P$ *and* $\sum\limits_{i\in\mathcal{T}} q_i^* < Q$.

In this case, both the attacker and the defender do not use all the available resources to attack and defend the network respectively. The result follows directly from Theorem C.1.

In Section C.4, we provide a numerical analysis for a general network architecture in which we validate the model on a real industrial case study.

C.3 Computing β

We consider the case of external attackers. Therefore, the graph G that we considered in the model represents the interconnections between equipment that contribute to the attack surface of the system at $t = 0$. Solving the game in Section C.2 enables us to identify the sensible target set $\mathcal{T}_S$ and to compute the NE strategy of the attacker p^*. To simplify the presentation, we assume that $P = 1$. In this case, we have $\sum\limits_{i\in\mathcal{T}_S} p_i^* = 1$. The NE strategy of the attacker represents the optimal allocation of attack resources on equipment given a

best response strategy of the defender. As we have mentioned earlier, since the game is a one-shot game, we are interested in the behavior of a rational attacker that lakes any observation of the defender's strategy before committing to an attack strategy. Therefore, the best payoff for the attacker is achieved by operating at the NE assuming a best response strategy of a rational defender.

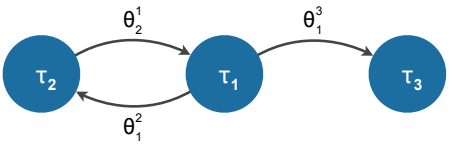


FIGURE C.2: Example of equipment in a network $\mathcal{N}$ accessible by an external attacker

We will show how to compute β for the *CMDP Type II* on an example. Let us take the case of a network $\mathcal{N}$ in which a set of equipment τ_1, τ_2, and τ_3 can be accessed by an external attacker at $t = 0$. Fig. C.2 depicts the interdependencies between this set of equipment. We assume that after solving the game corresponding to this scenario, we found the NE strategy of the attacker p^* and that $\mathcal{T}_S = \{\tau_1, \tau_2\}$. Let γ_1 be a vulnerability that exists on equipment τ_1. Let $\mathcal{G}'$ refer to the attack graph corresponding to the network $\mathcal{N}$. The first set of actions in this attack graph are depicted in Fig. C.3.

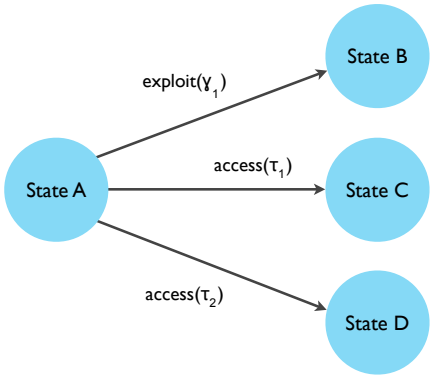


FIGURE C.3: Part of the attack graph of a network $\mathcal{N}$

In the *CMDP Type II* corresponding to $\mathcal{G}'$, the initial set of states that the attacker can be in refers to states x_{AB}, x_{AC}, and x_{AD} in which he is trying to exploit γ_1, access τ_1, and access τ_2 respectively. The NE strategy of the attacker p^* gives us the probability of attacking equipment τ_1 and τ_2 assuming an optimal distribution of intrusion detection resources on equipment by a rational defender. To find β, we combine p^* with the attack preferences of the attacker. Therefore, the probability of being in state x_{AB} is given by $\beta(x_{AB}) = p_1^* \times P_c(\textit{exploit } \gamma_1)$, where p_1^* and $P_c(\textit{exploit } \gamma_1)$ refer to the probability of attacking equipment τ_1 and the probability of choosing to exploit γ_1 when τ_1 is targeted respectively. Similarly, we find $\beta(x_{AC})$ and $\beta(x_{AD})$.

We note that if $P < 1$, we introduce an absorbing state to the CMDP in which the probability of being in that state at $t = 0$ equals $1 - P$. Finally, we note that we can use the

result of the game introduced in the previous section as a guideline in order to focus only on the sensible target set when generating the attack graph of the network.

In the next section, we validate the model in Section C.2 on a case study.

C.4 Case Study

We validate the model in Section C.2 on a network of a real industrial case study in which the values of the model parameters were the result of the application of a risk assessment method. However, even though in general most parameters such as W_i, μ_i and V_i^0 could be evaluated through a risk assessment method such as EBIOS [ANS10], evaluating the values of interdependencies θ_i^j is more challenging. In this section, we perform a sensitivity analysis to evaluate the impact of estimation errors on the values of θ_i^j on the strategies of the attacker and the defender.

We consider a network comprised of $n = 10$ nodes. The type of the nodes and the values of some of the model parameters are depicted in Table C.4 and Table C.5. The nodes in both tables are already sorted and numbered according to decreasing $W_i\lambda_i$ values as described in Section C.2.

TABLE C.4: Node types and individual parameters

Number	**Node Type**	$\boldsymbol{W_i}$	$\boldsymbol{V_i^0}$	$\boldsymbol{\mu_i}$
1	Business App. A	0.75	0.6	0.7
2	Intranet Portal	0.75	0.6	0.6
3	Mailing Server	0.75	0.3	0.6
4	Webmail Server	0.4	0.3	0.1
5	Business App. B	0.5	0.6	0.7
6	Intranet Common Services	1	0.6	0.1
7	Storage Area Network	1	0	0.1
8	Office Server	0.4	0.3	0.7
9	Authority Station	0.1	1	0.8
10	User Station	0.1	1	0.8

We study the NE strategies of both players in two different scenarios. In the first scenario, we consider a typical network in which the attack and defense costs are relatively high compared with the security assets of the nodes (i.e. $C_a = C_m = 0.1$). In addition, the use of the interdependencies between nodes in the attack process is not considered of high criticality (i.e. $\psi = 0.5$). In this scenario, the attacker may not be tempted to fully exploit the node interdependencies in his attack. The resource constraints for the attacker and the defender are set to $P = 0.8$ and $Q = 0.9$ respectively, which means that the budget of the defender is slightly superior to the budget of the attacker. In the second scenario, the values of nodes security assets outweigh attack and defense costs (i.e. $C_a = C_m = 0.001$), and in which exploiting the interdependencies between nodes can play a significant role in

TABLE C.5: Nodes interdependencies θ_i^j

i \ j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0	1	0	0	0	0.5	1	0	0	0
2	0	0	0	0	0	0.9	0.9	0	0	0
3	0	0	0	0	0	0.8	1	0	0	0
4	0	1	1	0	0	0.9	1	0	0	0
5	0	1	0	0	0	0.5	1	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0.5	0.9	0	0	0
9	0.8	0.9	0.3	0.1	0.8	0.9	0	0.3	0	0
10	0.5	0.5	0.2	0.1	0.5	0.9	0	0.2	0	0

the attack process (i.e. $\psi = 1$). In addition, due to the security requirements of such critical networks, the detection rate μ_i on all nodes is supposed to be $\mu_i \geq 0.5$ in the second scenario. Finally, we consider that the attack and defense resource constraints are set to $P = 1$ and $Q = 1$ respectively.

The NE strategies of the attacker and the defender are depicted in Table C.6. In both scenarios, the attacker/defender uses all his available resources to attack/defend. We note that both players focus on an attractive target set comprised of nodes 1, 2, 3, and 4 in the first scenario, and nodes 1, 2, 3, 4, and 5 in the second scenario. It is interesting to note that nodes 9 and 10 are not sensitive nodes despite having many dependencies stemming from them, as they have low security assets values to be worth attacking or defending. On the contrary, nodes 6 and 7 are not part of the attractive set despite their relatively high security assets and the absence of dependencies stemming from them. In the second scenario, the attractive target set increased by one node (node 5). This is due most probably to the fact that the attacker has additional available resources and that node 4 had its detection probability μ_i raised to 0.5 from 0.1, hence discouraging the attacker from spending too many resources to attack this node.

For both scenarios, we studied the case where the defender chooses a strategy different from the NE by performing a simulation of 1000 random strategies q^r for the defender, to which the attacker replies with the best response p' obtained using simple linear programming. The results of this experiment are displayed in Table C.7, where $U_D(p', q^r)_B$ represents the best utility for the defender out of the simulated strategies, and $U_D(p', q^r)_A$ represents the average utility for the defender. These results clearly show that deviating from the NE represents a loss in the utility of the defender, and that the guidelines provided in Section C.2 are indeed accurate.

The Security Information and Event Management (SIEM) software used in this industrial case study defines a metric to quantify the overall security of the network. This metric, which cannot be described in detail due to confidentiality reasons, consists in assessing, for each

TABLE C.6: Nash equilibrium in scenarios 1 and 2

Scenario 1	**Scenario 2**
p_1^*= 0.0712, q_1^*= 0.3135	p_1^*=0.1377, q_1^*=0.3762
p_2^*= 0.0931, q_2^*= 0.2088	p_2^*=0.1903, q_2^*=0.2127
p_3^*= 0.0758, q_3^*= 0.1915	p_3^*=0.1901, q_3^*=0.2126
p_4^*= 0.5599, q_4^*= 0.1862	p_4^*=0.2754, q_4^*=0.1897
p_5^*= 0, q_5^*= 0	p_5^*=0.2065, q_5^*=0.0088
p_6^*= 0, q_6^*= 0	p_6^*= 0, q_6^*= 0
p_7^*= 0, q_7^*= 0	p_7^*= 0, q_7^*= 0
p_8^*= 0, q_8^*= 0	p_8^*= 0, q_8^*= 0
p_9^*= 0, q_9^*= 0	p_9^*= 0, q_9^*= 0
p_{10}^*= 0, q_{10}^*= 0	p_{10}^*= 0, q_{10}^*= 0
$U_A = 0.898$, $U_D = -0.953$	$U_A = 1.736$, $U_D = -1.737$

TABLE C.7: Defender's payoff when deviating from NE in scenarios 1 and 2

	$U_D(p', q^r)_B$	$U_D(p', q^r)_A$
Scenario 1	-1.064	-1.248
Scenario 2	-1.967	-2.389

node, the types of attacks that can be mitigated given the current IDS configuration while taking into account the interdependencies between nodes in the evaluation process. After applying the optimal allocation of defense resources obtained at the NE, which translates in practice in configuring more efficient IDSs on critical nodes, we were able to notice a significant improvement of the overall security of the network, hence confirming the validity of our approach.

Sensitivity to $\boldsymbol{\theta_i^j}$. We analyze the impact of θ_i^j estimation errors on the identity of nodes that belong to the sensible target set $\mathcal{T}_S$. In both scenarios, nodes 8 to 10, due to their low security assets, remain in the unattractive set $\mathcal{T}_U$ even with a 20% estimation error on the values of each θ_i^j. In our model, the importance of a node is quantified by the value $W_i\lambda_i$, where λ_i mainly depends on ψ and the interdependencies θ_i^j. Therefore, inaccurate assessment of the interdependencies can have a significant impact on the results when the values of ψ and W_i are high. In our case study, when nodes 1, 2 and 3 have slightly erroneous interdependencies evaluations, we do not note any change in the sets $\mathcal{T}_S$ and $\mathcal{T}_U$. However, at the NE, we observe a small increase and decrease in the attacker and defender utilities respectively. For example, if on node 2, which has a relatively high security asset ($W_2 = 0.75$), $\sum_{j\in\Gamma^+(2)} \theta_2^j$ was overestimated by 0.4 (i.e. a 16% estimation error), U_A increases by 10% and U_D decreases by 5%. On the other hand, overestimating $\sum_{j\in\Gamma^+(5)} \theta_5^j$ by 0.1 (i.e. a 4% error) in scenario 1 is enough to include node 5 in $\mathcal{T}_S$. However, the impact of the error on U_A and U_D remains very low ($< 1\%$). Similarly, underestimating $\sum_{j\in\Gamma^+(5)} \theta_5^j$ by 0.1 in scenario 2 leads to the exclusion of node 5 from $\mathcal{T}_S$. At the NE, the attacker leverages

this situation and targets node 5. However, it is interesting to note that the impact on the players' utilities remains inferior to 1% in this case as well. This shows that in some cases, an approximate construction of the sensible target set $\mathcal{T}_S$ does not necessarily entail a sudden substantial utility gain (*resp.* loss) for the attacker (*resp.* defender).

These observations demonstrate that the model is robust enough to deal with slight inaccuracies in the evaluation of interdependencies parameters. However, given the number of parameters θ_i^j to evaluate in large networks, important estimation errors on these parameters could have a significant impact on the strategies of the attacker and the defender, hence justifying the need for a more formal and rigorous evaluation method of these parameters.

Appendix D

Symbols I

$$\theta_i^m = 2iF^m + (R^m - i)C^s S^m$$

$$\phi_i^m = 2(R^m - i)S^m + i(C^s S^m + \epsilon F^m)$$

$$\psi_i^m = \prod_{j=1}^{i-1} \Big(1 + \frac{C^s S^m}{\theta_j^m}\Big)$$

$$\omega^m = \sum_{i=1}^{R^m - 1} \frac{iF^m}{\theta_i^m} \psi_i^m$$

$$\tau^m = \sum_{i=1}^{R^m - 1} \frac{\psi_i^m}{\theta_i^m}$$

$$\alpha^m = 1 + \frac{NC^t}{C^s} - \frac{C^t S^m R^m}{C^s} \sum_{i \in \mathcal{D}} \Big(\frac{1}{S^i R^i} + \frac{C^s}{R^i}\big(\tau^i + \frac{R^i - 2\omega^i}{2R^i F^i}\big)\Big)$$

$$\beta^m = S^m R^m \sum_{i \in \mathcal{D}} \Big(\frac{1}{S^i R^i} + \frac{C^s}{R^i}\big(\tau^i + \frac{R^i - 2\omega^i}{2R^i F^i}\big)\Big)$$

$$\gamma^m = \frac{1}{\Big(1 + \frac{S^m}{\phi_{R^m}^m}\Big)\Big(\phi_1^m\big(1 - \frac{S^m}{\phi_{R^m-1}^m} + \frac{(S^m)^2}{\phi_{R^m-1}^m} + ((S^m)^2 - 1) \sum_{j=2}^{R^m - 2} \frac{1}{\phi_j^m}\big) + (S^m)^2\Big)}$$

$$\delta^m = \frac{S^m}{S^m + R^m(C^s S^m + \epsilon F^m)} + \gamma_m S^m \Big(1 + \phi_1^m \sum_{j=2}^{R^m - 1} \frac{1}{\phi_j^m}\Big)$$

$$\eta^m = \frac{F^m}{\delta^m R^m F^m + \gamma^m \phi_1^m \omega^m}$$

$$E^m = 2R^m F^m + C^s S^m (R^m - 2\omega^m) + 2R^m F^m C^s S^m \sum_{i=1}^{R^m - 1} \frac{\psi_i^m}{\theta_i^m}$$

$$G^m = 2R^m F^m - C^t S^m (R^m - 2\omega^m) - 2R^m F^m C^t S^m \sum_{i=1}^{R^m-1} \frac{\psi_i^m}{\theta_i^m}$$

$$\nu = \frac{\sum_{m \in \mathcal{T}_S} \frac{G^m}{E^m} - (|\mathcal{T}_S| - P)}{\sum_{m \in \mathcal{T}_S} \frac{1}{E^m}}$$

$$H^m = 1 + \frac{2S^m \sum_{i=1}^{R^m-1} \frac{\phi_1^m}{\phi_i^m}}{\phi_1^m - 2S^m}$$

$$W^m = \frac{-S^m \sum_{i=1}^{R^m-1} \frac{\phi_1^m}{\phi_i^m}}{\phi_1^m - 2S^m}$$

$$\kappa = \frac{|\mathcal{T}_S| - Q - \sum_{m \in \mathcal{T}_S} \frac{(1 - W^m)(2S^m + \phi_{R^m}^m) - S^m}{H^m(2S^m + \phi_{R^m}^m)}}{\sum_{m \in \mathcal{T}_S} \frac{1}{H^m(2S^m + \phi_{R^m}^m)}}$$

Appendix E

Symbols II

$$\alpha = \sum_{i\in\mathcal{V}_S}\frac{1}{W_i} + \sum_{r=1}^{N-1}\sum_{i\in L_r\cap(\mathcal{V}_S\setminus\mathcal{L}(\mathcal{T}_S))}\frac{\sum_{j=r+1}^{N_S(i)}(-1)^{j-r}\Delta_i^j}{W_i}$$

$$\beta = -\sum_{r=1}^{N-1}\sum_{i\in L_r\cap(\mathcal{V}_S\setminus\mathcal{L}(\mathcal{T}_S))}\sum_{j=r+1}^{N_S(i)}\frac{(-1)^{j-r}}{W_i}\Big(\frac{C_a}{1-a}\sum_{m\in Ch_S(i,j)}W_m + \sum_{\substack{m\in\overline{Ch_S(i,j)}\\ f(m)\in\mathcal{V}_S\setminus\mathcal{L}(\mathcal{T}_S)}}W_m\Big)$$
$$+\sum_{r=1}^{N-1}\sum_{i\in L_r\cap\mathcal{L}(\mathcal{T}_S)}\sum_{j\in Ch(i)}W_j\sum_{l=1}^{r}\frac{(-1)^{r-l}}{W_i^l}$$

$$A_i = \frac{1-a}{\alpha}\Big(1+\sum_{j=k+1}^{N_S(i)}(-1)^{j-k}\Delta_i^j\Big)\Big(Y_A\big(1-\frac{C_a}{1-a}\big)+\beta-S\Big)$$
$$+C_a\sum_{j=k+1}^{N_S(i)}(-1)^{j-k}\sum_{m\in Ch_S(i,j)}W_m + (1-a)\sum_{j=k+1}^{N_S(i)}(-1)^{j-k}\sum_{\substack{m\in\overline{Ch_S(i,j)}\\ f(m)\in\mathcal{V}_S\setminus\mathcal{L}(\mathcal{T}_S)}}W_m$$
$$-\sum_{j=k+1}^{N_S(i)}\sum_{m\in L_j\cap\mathcal{L}(\mathcal{T}_S)\cap Ch_S(i,j)}\mathbb{1}_{(j\neq N)}(-1)^{j-k}\sum_{t\in Ch(m)}(1-a)W_t$$

$$\gamma = 1 + W_1\sum_{r=2}^{N}\sum_{i\in L_r\cap\mathcal{V}_S}\Big(\frac{1}{W_i}+\sum_{j=1}^{r-1}\frac{(-1)^{r-j}}{W_i^j}\Big)$$

Appendix F

Résumé en français

Un réseau électrique intelligent, ou *smart grid*, est un réseau de nouvelle génération dont l'objectif est de fournir de nouveaux services, à la fois aux compagnies d'électricité et aux usagers. Le *smart grid*, reposant de plus en plus sur les technologies de l'information et de la communication, voit sa surface d'attaque augmenter. La rapidité de l'évolution des systèmes, mais aussi des menaces, contraste avec les longs cycles de vie de ce type d'infrastructure. La très forte hétérogénéité des composants, et la présence de contraintes techniques, topologiques, et organisationnelles, posent de nouveaux obstacles à l'application directe des approches traditionnelles de sécurité. Ainsi, la protection des biens sensibles, l'optimisation du déploiement des ressources de défense, et la capacité d'assurer à tout moment un certain niveau de confiance, sont des défis de sécurité pour ce type de système. L'estimation des risques de sécurité sur ces systèmes, requiert deux tâches complémentaires d'évaluation. La première vise à identifier et à estimer la vraisemblance des scénarios de menace pesant sur le système; le défenseur essaie, en particulier, d'identifier les différentes actions et méthodes qui permettent à un attaquant d'atteindre ses objectifs. Dans la seconde tâche, le défenseur évalue l'impact des menaces sur les biens critiques du système.

Dans la première partie de cette thèse, la théorie des jeux est utilisée pour optimiser le déploiement des ressources de défense dans le *smart grid*, en mettant l'accent sur l'impact des attaques sur les équipements. En analysant les interactions entre l'attaquant et le défenseur, nous identifions le choix optimal des modes de sécurité sur les équipements d'une infrastructure relative aux compteurs intelligents, ou *Advanced Metering Infrastructure (AMI)*, permettant de protéger la confidentialité des données des clients. En outre, nous caractérisons les ressources de défense minimales requises pour contrer toute menace qui compromettrait les données des clients dans l'AMI. Dans le *smart grid*, l'interdépendance entre l'infrastructure de communication et le réseau électrique rend également la gestion de la sécurité plus difficile. Nous avons abordé cette problématique en proposant un modèle analytique pour identifier et renforcer les équipements de communication utilisés dans le réseau électrique qui sont les plus sensibles. En se basant sur la théorie des jeux non-coopératifs, nous avons modélisé les interactions entre un attaquant et un défenseur, et inféré les ressources de

défense minimales requises ainsi que la stratégie optimale de défense minimisant le risque dans le réseau électrique. Notre modèle est ensuite validé via une étude de cas basée sur le réseau de transport d'électricité du réseau électrique polonais.

Le *smart grid* repose sur des systèmes de contrôle industriel pour une distribution de l'électricité qui soit efficace, fiable, et sûre. Afin d'améliorer la sécurité de ces systèmes, la stratégie de défense a besoin d'être à la fois proactive, en anticipant les cibles potentielles des attaquants, et réactive en ajustant le type et la puissance de la réponse en fonction du niveau de la menace. La deuxième partie de cette thèse aborde cette problématique et présente une solution qui calcule la politique de sécurité optimale, garantissant la protection des biens critiques, en utilisant une approche basée sur les graphes d'attaques afin de représenter l'évolution de l'état de l'attaquant dans le système. D'abord, nous proposons un modèle de graphes d'attaques prenant en compte les caractéristiques spécifiques aux systèmes de contrôle industriel, et une méthode permettant sa construction. Dans cette étude, nous mettons l'accent sur l'évaluation du risque des cyber attaques sur les systèmes de contrôle industriel entre deux périodes de maintenance successives. Nous identifions en particulier, pour un attaquant donné, la séquence des actions qui peut être exécutée pour compromettre un équipement sensible dans le système. Pour calculer de façon automatique la politique de sécurité optimale garantissant que les objectifs du défenseur seront satisfaits, nous proposons une approche qui utilise les processus de décision markoviens sous contraintes (PDMC) en se basant sur les informations contenues dans le graphe d'attaque. La solution du PDMC peut être utilisée comme système d'aide à la décision pour répondre aux intrusions d'une manière efficace, ou pour prioriser le déploiement des mesures de sécurité avant qu'une attaque ait lieu. En outre, la solution du PDMC peut être combinée avec l'information présente dans le graphe d'attaque pour comparer la sécurité relative de deux architectures ou de deux configurations du système. Nous validons notre approche sur une étude de cas d'un système AMI.

F.1 Analyse des attaques sur la confidentialité des données dans l'AMI basée sur la théorie des jeux

L'infrastructure relative aux compteurs intelligents (*Advanced Metering Infrastructure* ou *AMI*), est un système composé de compteurs intelligents, de réseaux de communication, et de systèmes de gestion de données qui permet une communication bidirectionnelle entre la compagnie d'électricité et les clients. Le déploiement des compteurs intelligents dans l'AMI doit assurer la sécurité des données privées. En effet, si les données des compteurs intelligents sont compromises par un attaquant, elles peuvent potentiellement lui permettre de connaître les habitudes des utilisateurs et même prédire leurs comportements. Dans ce chapitre, nous nous intéressons à la protection de la confidentialité des données dans l'infrastructure AMI constituée d'un ensemble de nœuds ayant des actifs de sécurité corrélés. Sur chacun de ces nœuds, le défenseur peut choisir un mode de sécurité parmi un ensemble

de modes disponibles sur le nœud. Nous essayons de répondre aux questions suivantes : Quel est le comportement d'un attaquant rationnel ? Quelle est la stratégie optimale du défenseur ? Est-ce que nous pouvons configurer les modes de sécurité sur chaque nœud afin de décourager l'attaquant d'attaquer ?

Pour répondre à ces questions, nous formulons le problème en un jeu non-coopératif entre un attaquant et un défenseur. En fait, en plus de la valeur des données des clients, stockées sur les nœuds, la stratégie du défenseur doit prendre en compte les cibles potentielles de l'attaquant. Dans notre jeu, l'objectif de l'attaquant est de choisir ces cibles afin d'intercepter la quantité maximale de données privées des clients. En revanche, l'objectif du défenseur est de choisir le mode de sécurité qui doit être configuré sur chaque nœud de l'AMI pour protéger au maximum la confidentialité des données des clients envoyées par ce nœud.

F.1.1 Modèle du système

Nous considérons une architecture de communication arborescente $\mathcal{T}$ pour l'AMI avec un seul nœud racine comme indique la Figure F.1. Dans cette architecture, les nœuds représentent des équipements dans l'AMI. Pour chaque nœud i, la valeur des données W_i représente la perte en termes de confidentialité des données si une attaque sur i réussit. Nous supposons que ces valeurs ont été déjà évaluées suite à l'application d'une méthode d'analyse des risques (par exemple [ANS10]). Chaque nœud i collecte des données de ces nœuds fils $Ch(i)$, les agrège, puis les envoie à son nœud parent. Par conséquent, nous supposons que $W_i \geq \sum_{j \in Ch(i)} W_j$. La valeur des données du nœud i est la somme de la valeur des données générées par ce nœud et celle des données générées par ces fils. Nous supposons qu'il existe N niveaux d'agrégation. Nous considérons que chaque nœud ne peut appartenir qu'à un seul niveau d'agrégation. Le nœud racine de $\mathcal{T}$ correspond au niveau 1. Les compteurs intelligents sont représentés par des nœuds appartenant au niveau d'agrégation N.

La table F.1 liste les principaux symboles utilisés dans cette section.

F.1.2 Formulation du jeu

Nous considérons un jeu avec deux joueurs, un attaquant et un défenseur. Nous supposons que les deux joueurs ont une connaissance complète de l'architecture du système. Sur chaque nœud, le défenseur choisit un mode de sécurité parmi un ensemble de modes disponibles sur le nœud. Dans notre cas, le défenseur choisit le taux de chiffrement des données envoyées par le nœud. Par exemple, si 100 paquets sont envoyés par le nœud, le défenseur choisira l'ensemble de paquets qui doivent être chiffrés. Nous considérons que les données envoyées sur chaque lien de communication sont chiffrées aves des clefs de chiffrement différentes ou en utilisant des algorithmes de chiffrement différents. Sur le nœud racine, les données sont chiffrées pour être stockées après analyse.

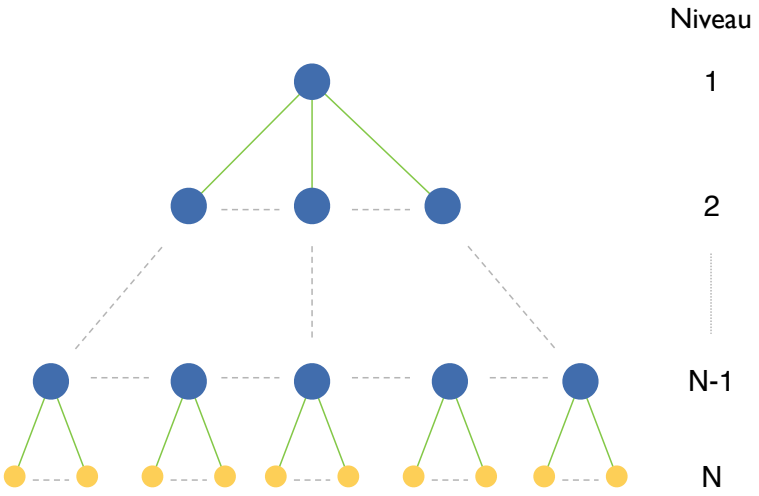


FIGURE F.1: Architecture de communication de l'AMI

L'objectif de l'attaquant est d'intercepter les données en attaquant les nœuds sans qu'il soit détecté. Si l'attaquant souhaite intercepter les données envoyées par le nœud i, il a le choix d'attaquer le nœud i ou son nœud parent. Nous faisons l'hypothèse que les clefs de chiffrement sont stockées dans un cryptoprocesseur qui ne soit pas accessible à l'attaquant. Les données arrivant dans un nœud sont déchiffrées en utilisant la clef correspondante, traitées, puis chiffrées en utilisant une autre clef de chiffrement. L'attaquant n'a pas de contrôle sur les processus de chiffrement et de déchiffrement. Nous supposons qu'un système de détection d'intrusion est installé sur chaque nœud avec un taux de détection a.

Soit p_i la probabilité d'attaquer le nœud i. La stratégie de l'attaquant est soumise à la contrainte budgétaire $\sum_i p_i \leq P \leq 1$ $(0 \leq p_i \leq 1 \ \forall i)$. Nous considérons qu'à un certain moment, l'attaquant ne peut attaquer qu'un seule nœud. Soit s_i le taux de chiffrement des paquets envoyés par le nœud i. Dans notre modèle, la stratégie du défenseur est soumise à la contrainte suivante $\sum_i s_i \leq S \leq Y$ $(0 \leq s_i \leq 1 \ \forall i)$. Nous considérons que le coût pour compromettre et chiffrer les données d'un nœud i est proportionnel à la valeur des données W_i du nœud et sont données par $C_a W_i$ et $C_e W_i$ respectivement, où $0 \leq C_a, C_e \leq 1$.

La probabilité de compromettre des données non chiffrées envoyées par un nœud i utilisant un taux de chiffrement s_i, sans être détecté, est donnée par $W_i(p_i + p_{f(i)})(1-a)(1-s_i)$. Nous supposons que $1-a > C_a$. Sinon, l'attaquant n'aura pas intérêt à attaquer, puisque le coût d'attaque sera supérieur au gain dans le cas où l'attaque réussit sans qu'il soit détecté.

Les fonctions d'utilité U_A et U_D de l'attaquant et du défenseur respectivement sont les suivantes :

TABLE F.1: Liste des principaux symboles dans la Section F.1

$\mathcal{T}$	un arbre (un graphe connecté sans cycles)
$\mathcal{L}(\mathcal{T})$	ensemble des feuilles de l'arbre $\mathcal{T}$
$\mathcal{V}$	ensemble des nœuds dans $\mathcal{T}$
$\mathcal{V}_S$	ensemble des nœuds sensibles dans $\mathcal{T}$
$\mathcal{T}_S$	un sous-arbre de $\mathcal{T}$ qui consiste des nœuds $i \in \mathcal{V}_S$
N	nombre des niveaux d'agrégation dans $\mathcal{T}$
Y	nombre de nœuds dans $\mathcal{T}$
$N_S(i)$	niveau d'agrégation maximale des feuilles du sous-arbre $\mathcal{T}_i$ de $\mathcal{T}_S$ qui ont i comme nœud racine
L_i	ensemble de nœuds appartenant au niveau d'agrégation i
W_i	valeur des données du nœud i
W_i^k	valeur des données du parent du nœud $i \in L_m$ au niveau $k < m$
$f(i)$	parent du nœud i
$Ch(i,k)$	ensemble des fils du nœud $i \in L_m$ au niveau $k > m$
$Ch(i)$	ensemble des fils du nœud $i \in L_m$ au niveau $m+1$
$Ch_S(i,k)$	ensemble des fils du nœud $i \in L_m$ au niveau $k > m$ appartenant à $\mathcal{V}_S$
$Ch_S(i)$	ensemble des fils du nœud $i \in L_m$ au niveau $m+1$ appartenant à $\mathcal{V}_S$
$\overline{Ch_S(i,k)}$	$Ch(i,k) \backslash Ch_S(i,k)$
p_i	probabilité d'attaquer le nœud i
s_i	taux de chiffrement des données envoyées par le nœud i
P	budget d'attaque
S	ressources de chiffrement

$$
\begin{aligned}
U_A(p,s) &= \sum_{i\in\mathcal{V}} \big(W_i(p_i + p_{f(i)})(1-a)(1-s_i) - p_i C_a W_i\big) \\
&= \sum_{i\in\mathcal{V}} \big(W_i p_i (1-a)(1-s_i) - p_i C_a W_i\big) + \sum_{\substack{i\in\mathcal{V} \\ i\notin L_N}} \sum_{j\in Ch(i)} p_i W_j (1-a)(1-s_j) \\
U_D(p,s) &= -\sum_{i\in\mathcal{V}} \big(W_i p_i (1-a)(1-s_i) + s_i C_e W_i\big) - \sum_{\substack{i\in\mathcal{V} \\ i\notin L_N}} \sum_{j\in Ch(i)} p_i W_j (1-a)(1-s_j)
\end{aligned}
$$

F.1.3 Résolution du jeu

Dans le contexte des jeux non-coopératifs, nous nous intéressons au concept de l'équilibre de Nash où aucun joueur n'a intérêt à changer de stratégie d'une manière unilatérale [OR94]. Cette notion d'équilibre permet à chaque joueur de maximiser sa fonction d'utilité en prenant en compte les stratégies des autres joueurs.

F.1.3.1 Ensemble de cibles sensibles

Dans la section précédente, nous avons considéré que chaque joueur possède des ressources limitées. Par conséquent, il est raisonnable de supposer que les joueurs vont distribuer

leurs ressources d'une manière intelligente afin de maximiser leurs fonctions d'utilité. Nous pouvons alors prédire que l'attaquant va identifier les cibles qui vont lui permettre de maximiser son gain et ainsi essayer de les compromettre. Dans ce cas, l'objectif du défenseur sera d'identifier ces cibles afin de protéger la confidentialité de leurs données.

Soit $\mathcal{V}_S$ l'ensemble des nœuds sensibles qui vont être attractifs pour l'attaquant. Nous prouvons que $\mathcal{V}_S$ peut être déterminé par l'algorithme 12. Nous notons que dans le cas où un nœud appartient à $\mathcal{V}_S$, son nœud père appartiendra aussi à $\mathcal{V}_S$.

Algorithm 12

Input: Tree $\mathcal{T}$ and the set of nodes $\mathcal{V}$
Result: The sensible target set $\mathcal{V}_S$

function FindSensibleTargetSet($\mathcal{T}$, $\mathcal{V}$)
  for $x \in \mathcal{V}$ **do**
    if $x \in \mathcal{V} \backslash \mathcal{L}(\mathcal{T})$ **then**
      $W_{t_i} \leftarrow W_i + \frac{1}{(1-\frac{C_a}{1-a})} \sum_{j \in Ch(i)} W_j$
    else
      $W_{t_i} \leftarrow W_i$
    end if
  end for
  $W'_i \leftarrow$ SortInDescendingOrder($W_{t_{\sigma(i)}}$)
  Initialization: $Y_A = Y$, α, β
  while $Y_A \geq 1$ & $W'_{Y_A} \leq \frac{1}{\alpha(1-\frac{C_a}{1-a})}\big(Y_A(1-\frac{C_a}{1-a}) + \beta - S\big)$ **do**
    $Y_A \leftarrow Y_A - 1$
    update(α)
    update(β)
  end while
  $\mathcal{V}_S = \{\sigma(i) \in \mathcal{V},\ s.t.\ i \in [\![1; Y_A]\!]\}$
end function

Nous prouvons le lemme suivant :

Lemme F.1. *Dans le cas où le défenseur possède S_{min} ressources pour le chiffrement, les données stockées sur chaque nœud vont être chiffrées avec un taux de chiffrement strictement positif, où S_{min} est donnée par la formule suivante :*

$$S_{min} = Y\Big(1 - \frac{C_a}{1-a}\Big) + \beta$$

β est donné dans l'Annexe E. Pour la suite, nous supposerons que $S \leq S_{min}$.

Nous avons ainsi le résultat suivant :

Théorème F.1. *Un attaquant rationnel va attaquer seulement l'ensemble des nœuds $\mathcal{V}_S$.*

Nous analysons deux types d'interactions qui pourront avoir lieu entre l'attaquant et le défenseur.

Jeu simultané. Lorsque l'attaquant et le défenseur prennent leurs décisions en même temps (type d'interaction connu sous le nom de jeu simultané [OR94]), nous prouvons que sous l'hypothèse de ressources limitées $\sum_i p_i = P$ et $\sum_i s_i = S$, un équilibre de Nash existe et peut être caractérisé analytiquement. Cet équilibre représente les stratégies optimales acceptables pour les deux joueurs. Ainsi, la stratégie du défenseur à l'équilibre de Nash représente sa meilleure réponse à la stratégie de l'attaquant. Notons que la stratégie du défenseur de chiffrer les données du nœud i ne dépend pas seulement de W_i et de la stratégie de l'attaquant, mais aussi du nombre de nœuds et de la valeur de leurs données le long du trajet du nœud i au nœud racine.

Jeu de Stackelberg. En général, l'attaquant choisit sa stratégie en prenant en compte les mesures de défense déjà déployées dans le système. Nous analysons les interactions entre l'attaquant et le défenseur dans le contexte d'un jeu de Stackelberg [OR94]. Dans ce type de jeu, le leader choisit sa stratégie en premier. Le suiveur, observant la stratégie du leader, choisit sa stratégie. Le problème du leader est d'anticiper quelle sera la stratégie du suiveur et ainsi choisir une stratégie qui va lui permettre de maximiser son gain connaissant la réaction du suiveur. Dans notre jeu, le leader est le défenseur et le suiveur est l'attaquant. Nous prouvons qu'un équilibre de Stackelberg existe et peut être caractérisé analytiquement. Nous prouvons aussi le théorème suivant :

Théorème F.2. *Le défenseur a besoin d'au moins* $Y(1 - \frac{C_a}{1-a}) - \frac{C_a}{1-a} \sum_{i \in L_k} W_i \sum_{j=1}^{k-1} \frac{(-1)^{k-j}}{W_i^j}$ *ressources de chiffrement pour décourager l'attaquant d'attaquer.*

Le théorème F.2 montre qu'avec des ressources de chiffrement suffisantes, le défenseur est capable de décourager l'attaquant d'attaquer en augmentant le coût d'attaque par rapport au gain potentiel pour l'attaquant.

F.2 Gestion des interdépendances des risques de sécurité entre le réseau électrique et le réseau de communication basée sur la théorie des jeux

Au cours des dix dernières années, les opérations de gestion et de contrôle du réseau électrique reposent de plus en plus sur l'infrastructure de communication. Cette infrastructure a donc le potentiel d'augmenter la surface d'attaque du réseau électrique. Une attaque sur un équipement de communication utilisé pour contrôler un processus industriel peut avoir un impact considérable sur les infrastructures critiques. Réciproquement, un équipement électrique responsable d'alimenter un ensemble d'équipements de communication est critique pour l'infrastructure de communication : si la source d'alimentation de ces équipements est compromise, les équipements de communication ne seront plus en mesure d'atteindre leurs objectifs. Dans ce chapitre, le système de communication désignera l'infrastructure responsable du contrôle et de la gestion du réseau électrique.

Traditionnellement, la fiabilité du réseau électrique et la sécurité du système de communication ont été évalués d'une manière indépendante en utilisant différentes méthodologies (par exemple [Wen05] pour les réseaux électriques et [ANS10] pour les réseaux de communications). Récemment, des chercheurs se sont focalisés sur la modélisation des interdépendances entre les infrastructures critiques, et en particulier entre le réseau électrique et l'infrastructure de communication [LKK07, CDN11, BCG+12, LVS+12]. L'objectif est d'évaluer l'impact des attaques informatiques qui ciblent les équipements de communication sur le réseau électrique. Les modèles qui se basent sur des analyses comportementales et la simulation sont parmi les principales catégories de modèles qui ont été proposées pour analyser les défaillances en cascade et étudier la dynamique des pannes électriques. Dans cette catégorie de modèle, nous trouvons principalement les modèles basés sur des agents [CGT07], les réseaux de petri [CSAB11], et la co-simulation [LVS+12].

L'interdépendance entre le réseau électrique et l'infrastructure de communication rend donc la gestion du risque de sécurité plus difficile et doit ainsi être prise en compte dans la sécurisation du système. Dans ce chapitre, nous adressons cette problématique en présentant un modèle analytique pour identifier et sécuriser les équipements de communications les plus critiques, utilisés dans le réseau électrique. En utilisant la théorie des jeux non-coopératifs, nous modélisons les interactions entre l'attaquant et le défenseur. Nous calculons les ressources de défense minimales et la stratégie optimale du défenseur qui permettent de minimiser le risque de sécurité sur le réseau électrique. En plus, nous proposons une méthode qui permet d'évaluer les valeurs des paramètres du modèle analytique utilisé pour l'évaluation de l'impact des défaillances des équipements dans le réseau électrique. La structure des fonctions d'utilité, qui prennent en compte l'existence d'équipements de redondance dans l'infrastructure de communication, nous permettent de caractériser analytiquement les stratégies des joueurs à l'équilibre de Nash. Par conséquent, nous pouvons évaluer les changements potentiels dans les comportements des joueurs suite à des erreurs d'estimation des valeurs d'un ensemble de paramètres du modèle. Nous validons notre modèle sur une étude de cas basé sur le réseau de transport d'électricité polonais.

Dans ce chapitre, un certain nombre d'hypothèses sur les connaissances et les compétences de l'attaquant ont été prises. Par exemple, nous supposons que l'attaquant connaît la topologie du réseau électrique. Toutefois, bien que cette hypothèse soit forte, Li et al. [LPS13] montrent qu'un attaquant ayant accès à un ensemble limité de données peut déduire la topologie du réseau électrique.

F.2.1 Modèle d'interdépendance

Nous appelons risque initial, le risque de sécurité sur un nœud avant que l'impact d'un accident ou une attaque se propage entre les nœuds du système. Soient $r_i^e(0)$ et $r_j^c(0)$ le risque initial sur le nœud électrique i et l'équipement de communication j respectivement. Nous supposons que le risque initial sur un nœud du système est un nombre réel positif et a été évalué suite à l'application d'une méthode d'analyse des risques. Nous nous intéressons

au processus de diffusion du risque entre les nœuds d'une même infrastructure et les nœuds de deux infrastructures différentes.

Nous nous basons sur le modèle proposé par Alpcan et Bambos dans [AB09] pour représenter les dépendances des risques de sécurité à l'aide de graphes. Nous modélisons les interdépendances entre l'infrastructure de communication et le réseau électrique par un graphe orienté pondéré $\mathcal{D} = (V,\ E,\ f)$ où $V = \{v_1, v_2, ..., v_N\}$ est un ensemble fini de sommets représentant les nœuds électriques et les nœuds de communication, E l'ensemble des arêtes, et $f : E \to \mathbb{R}^+$ une fonction où $f(e_{ij})$ représente le poids associé à l'arête e_{ij}.

Soit $V = \{\mathcal{T}^e, \mathcal{T}^c\}$ où $\mathcal{T}^e = \{v_1, v_2, ..., v_{N_e}\}$ et $\mathcal{T}^c = \{v_{N_e+1}, v_{N_e+2}, ..., v_{N_e+N_c}\}$ représentent l'ensemble des nœuds électriques et les nœuds de communication respectivement. Nous représentons $\mathcal{D}$ par la matrice d'adjacence pondérée $M = [m_{ij}]_{N\times N}$ suivante :

$$M = \begin{pmatrix} B & D \\ F & S \end{pmatrix}$$

où $B = [b_{ij}]_{N_e\times N_e}$, $D = [d_{ij}]_{N_e\times N_c}$, $F = [f_{ij}]_{N_c\times N_e}$, et $S = [s_{ij}]_{N_c\times N_c}$. Les éléments des matrices B, D, F, et S sont des nombres réels positifs. Nous supposons que ces matrices sont des matrices stochastiques gauches (la somme sur chaque colonne vaut 1). Pour chaque nœud k, nous évaluons le poids des autres nœuds à impacter le nœud k. Par exemple, les matrices B et S représentent les dépendances entre les nœuds électriques et les nœuds de communication respectivement.

F.2.2 Diffusion du risque

Nous considérons que les premiers effets d'une attaque sur un équipement de communication aura lieu dans l'infrastructure de communication. Nous introduisons la métrique t_c dans le système de communication représentant le temps moyen pour que l'impact d'une attaque sur un équipement de communication se propage dans l'infrastructure de communication.

Soit $R^e(t) = [r_i^e(t)]_{N_e\times 1}$ et $R^c(t) = [r_i^c(t)]_{N_c\times 1}$ les vecteurs représentant respectivement les risques sur les nœuds électriques et les nœuds de communication au temps t. Nous observons l'évolution du système en temps discret. Soit $S^l = [s_{ij}^l]_{N_c\times N_c}$ la puissance l-ième de la matrice S. À l'étape d'attaque r, le gain est multiplié par γ_c^r où $\gamma_c \in [0, 1]$. En fait, nous considérons que chaque action de l'attaquant dans le système augmentera la probabilité qu'il soit détecté. Soit $S^{max} = [s_{ij}^{max}]_{N_c\times N_c}$ la matrice tel que $s_{ij}^{max} = \max_{l=1,...,\lfloor t_c \rfloor} \gamma_c^l s_{ij}^l$ où t_c représente le temps moyen pour qu'une attaque sur un nœud de communication atteigne les autres nœuds de communication et S_n^{max} la matrice normalisée de S^{max} par rapport à ces lignes tel que $\forall j,\ \sum_i {s_n^{max}}_{ij} = 1$.

Nous adoptons une approche similaire à [AB09] pour pondérer le risque immédiat par rapport au risque future. Soient β le poids du risque sur les nœuds de communication et τ le poids du risque sur les nœuds électriques se propageant vers les nœuds de communication

au temps $t = 0$, et δ le poids du risque futur par rapport au risque total sur les nœuds de communication. Nous prouvons que le système itératif de diffusion du risque converge et qu'un équilibre existe lorsque $\delta < 1$ et il est donné dans ce cas par $R^{c*} = (I-\delta H)^{-1}(\beta R^c(0)+\tau D^T R^e(0))$ où $H = S_n^{max} FBD$, et β, τ, et δ sont des nombres réels positifs tel que $\beta+\tau+\delta = 1$. Par conséquent, nous pouvons prédire la propagation du risque d'une attaque sur un équipement de communication dans le réseau électrique et le réseau de communication. Dans le cas où l'attaquant a accès à H, il pourra choisir ces cibles d'une manière intelligente pour maximiser l'impact de ces attaques sur le réseau électrique.

F.2.3 Jeu de sécurité

Nous considérons un jeu avec deux joueurs, un attaquant et un défenseur. L'objectif de l'attaquant/défenseur est de distribuer ses ressources d'attaque/de défense sur les nœuds de communication pour maximiser/minimiser l'impact des attaques sur le réseau électrique. Nous formulons le problème en un jeu non-coopératif et nous analysons le comportement de l'attaquant et du défenseur à l'équilibre de Nash. Nous considérons le pire des cas dans lequel l'attaquant et le défenseur connaissent l'architecture du système. Nous associons à chaque nœud de communication i une charge l_i qui représente la charge de travail dont ce nœud est responsable. Soient $L = diag(l_i)_{N_c \times N_c}$ la matrice de charge et $W = [w_{ij}]_{N_c \times N_c}$ la matrice de redondance tel que $\forall i$, $w_{ii} = -1$ et $\sum_{j,j\neq i} w_{ij} \leq 1$. Lorsque $i \neq j$, w_{ij} représente la fraction de la charge de travail du nœud i, le nœud j sera responsable dans le case où i est compromis.

Les fonctions d'utilité U_a et U_d de l'attaquant et du défenseur respectivement sont les suivantes :

$$U_a(p,q) = pR_D^{c*}(e^T - q^T) - pR_D^c(0)C^a p^T - \psi pL(Wq^T - I(e^T - 2q^T))$$

$$U_d(p,q) = -pR_D^{c*}(e^T - q^T) - qR_D^c(0)C^d q^T + \psi pL(Wq^T - I(e^T - 2q^T))$$

où $p = [p_i]_{1\times N_c}$ représente la stratégie de l'attaquant où $p_i \in [0,1]$ représente les ressources d'attaque allouées à la cible $i \in \mathcal{T}^c$, $q = [q_j]_{1\times N_c}$ représente la stratégie du défenseur où $q_j \in [0,1]$ représente les ressources de défense allouées à la cible $j \in \mathcal{T}^c$, $R_D^c(0)$, R_D^{c*}, C^a et C^d sont des matrices diagonales et C^a et C^d représentent respectivement les coûts pour attaquer et défendre les nœuds de communication, I est la matrice identité, et $e = (1,...,1)_{1\times N_c}$.

Les fonctions d'utilité des joueurs sont composées de trois parties : le gain des attaques, le coût des actions *attaquer/défendre*, et l'effet de l'existence des équipements de redondance permettant d'assurer le contrôle du réseau électrique lorsqu'un ensemble de nœuds de communication est compromis. Le paramètre $\psi \in [0,1]$ représente la probabilité que les équipements de redondance soient capable de prendre en charge la charge de travail des nœuds de communication compromis.

Nous analysons les cas où l'attaquant et le défenseur prennent leurs décisions en même temps (jeu simultané [OR94]) et le cas où l'attaquant choisit sa stratégie après avoir observé la stratégie du défenseur (Jeu de Stackelberg). Nous prouvons qu'un équilibre de Nash existe pour le jeu simultané et qu'un équilibre de Stackelberg existe pour le jeu de Stackelberg et que dans les deux cas, l'équilibre peut être caractérisé analytiquement.

F.3 Un modèle d'exécution d'attaque pour évaluer la sécurité des systèmes de contrôle industriel

La gestion des risques de sécurité, dans un système de contrôle industriel, représente un défi. En particulier, afin d'évaluer l'impact d'une attaque sur un système de contrôle industriel, les interdépendances entre les différents composants doivent être prises en compte. En plus, la probabilité du succès d'une attaque est fortement corrélée au profil de l'attaquant et ses connaissances de l'architecture du système.

Dans ce chapitre, nous présentons le modèle d'exécution d'attaque (*Attack Execution Model* ou AEM) qui est un graphe d'attaque représentant l'évolution de l'état de l'attaquant dans le système après chaque action de l'attaquant. Nous nous intéressons à l'évaluation des risques de sécurité sur un système de contrôle industriel avant la prochaine date de maintenance. Pour un profil d'attaquant, nous générons toutes les actions potentielles que l'attaquant pourra exécuter dans le système. En général, ayant des contraintes opérationnelles (par exemple, des services qui doivent être protégés pour des raisons de sûreté de fonctionnement, difficulté d'arrêter une partie du système sans impacter le processus industriel) et des contraintes sur les ressources de défense disponibles, l'opérateur doit prendre un certain nombre de décisions pour protéger le système vulnérable. Pour prendre la meilleure décision, l'opérateur doit pouvoir quantifier le risque que les vulnérabilités non patchées posent sur le système. En plus, l'évaluation de la probabilité d'exploiter ces vulnérabilités avec succès doit prendre en compte le profil de l'attaquant qui inclut ses compétences, son niveau d'accès sur les équipements, et sa connaissance de la topologie du système de contrôle. Dans certain cas, selon le profil de l'attaquant, certaines vulnérabilités ne sont pas exploitables.

Vu la complexité croissante des interconnexions entre les équipements industriels, une évaluation manuelle de l'impact de compromettre une vulnérabilité dans le système industriel représente un défi. Par conséquent, un outil permettant d'évaluer les impacts des attaques sur les équipements d'un système industriel et les services du processus industriel associé est nécessaire. Pour atteindre cet objectif, l'outil doit satisfaire les exigences suivantes : i) pouvoir modéliser les interdépendances pouvant exister entre les équipements physiques et les services du processus industriel, ii) pouvoir modéliser les interdépendances entre les services, iii) pouvoir modéliser le temps nécessaire pour exécuter chaque action d'attaque, et finalement iv) pouvoir prendre en compte le profil de l'attaquant (connaissance de l'architecture du système, niveau de compétence, etc.) et l'ensemble des connaissances acquises par l'attaquant après avoir compromis un ensemble d'équipements dans le système.

F.3.1 Architecture du système de contrôle

Dans notre modèle, nous représentons le système de contrôle en deux couches : la couche réseau et la couche service. Dans la couche réseau, nous modélisons les équipements physiques et leurs interconnexions. Le réseau est représenté par un graphe dirigé $\mathcal{H} = \langle \mathcal{T}, \mathcal{Y}, l_{\mathcal{Y}} \rangle$ où $\mathcal{T}$ représente l'ensemble des sommets et $\mathcal{Y}$ l'ensemble des arêtes de $\mathcal{H}$. Chaque sommet représente une machine physique dans le réseau. La communication entre l'équipement τ_i et l'équipement τ_j est représentée par l'arête y_{ij}. La fonction $l_{\mathcal{Y}} : \mathcal{Y} \rightarrow \{n, m\}$ associe une étiquette à une arête où n représente une communication réseau et m représente une communication qui dépend d'une intervention humaine. Une communication réseau peut être établie entre deux équipements s'ils peuvent communiquer via le réseau. Par contre, dans certain cas, un opérateur humain intervient manuellement pour transférer des données ou des fichiers de configuration entre deux machines du système. Ce scénario a lieu en général lorsqu'une machine a besoin de données provenant d'une autre machine avec laquelle elle ne peut pas communiquer via le réseau.

La couche service représente les services utilisés pour exécuter des processus industriels et leurs interdépendances. Chaque nœud du système fournit un service ou un ensemble de services. Plusieurs nœuds peuvent interagir pour fournir un service. Le graphe de dépendance entre les services est représenté par le tuple $\mathcal{D} = \langle \Delta, \rightarrow \rangle$ où Δ représente l'ensemble des services dans le système et $\rightarrow$ est une relation binaire représentant une dépendance entre deux services. En général, nous pouvons avoir des ensembles disjoints de services interdépendants et des dépendances cycliques entre les services.

F.3.2 Modèle d'exécution d'attaque

Le profil de l'attaquant joue un rôle important dans l'évaluation de la probabilité du succès des attaques et leurs impact potentiel sur le système. Un *attaquant* est représenté par un type, un niveau de compétence, un ensemble d'actions qu'il peut exécuter (scanner le réseau, accéder aux équipements, exploiter des vulnérabilités), et un ensemble de préférences (profondeur du scan, coût de l'attaque, le gain suite à une attaque, etc.).

L'exécution d'une action nécessite que l'attaquant ait acquis un ensemble de connaissances. Par exemple, l'attaquant devrait connaître l'existence d'une vulnérabilité sur une machine et ait acquis les niveaux d'accès et développé les outils nécessaires pour pouvoir compromettre cette vulnérabilité. Nous considérons les types de connaissance suivants : les informations sur les machines (services en cours d'exécution, etc.) et leurs interconnexions, l'ensemble des *credentials* sur les machines, et l'ensemble des outils nécessaires pour exploiter certaines vulnérabilités dans le réseau.

Nous définissons l'état de l'attaquant comme suit :

Définition F.1 (État de l'attaquant)**.** *À un certain instant, l'*état de l'attaquant *représente l'ensemble des niveaux d'accès acquis sur les équipements et l'ensemble des connaissances à la disposition de l'attaquant.*

L'état de l'attaquant dans le système évolue en fonction de ces actions. Cette évolution est décrite selon un ensemble de règles Ξ. Il existe quatre types de règles : scan, accès réseau, accès humain, et exploit. Chaque règle a besoin d'un ensemble de préconditons pour qu'elle soit exécutable et un ensemble de postconditions qui représentent le résultat de l'exécution de la règle. Un coût et un gain sont associés à l'exécution de chaque règle. L'attaquant essaye d'accéder à un équipement à distance en utilisant son niveau d'accès sur une machine qu'il a déjà compromis et un ensemble de connaissances. Dans le cas où ce type d'accès est permis dans la politique de contrôle d'accès, le résultat de l'exécution de la règle *accès réseau* est l'ensemble des niveaux d'accès accordés à l'attaquant sur l'équipement à distance et un accès aux fichiers de configuration système et aux *credentials* qui se trouvent sur cet équipement accessibles avec les nouveaux niveaux d'accès. Lorsqu'un équipement a besoin de données d'un autre équipement avec lequel il ne peut pas se connecter via le réseau, un opérateur intervient et transmet ces fichiers manuellement (par exemple, en utilisant des clés USB). Nous décrivons ce type de scénario par la règle *accès humain.*

Pour représenter l'évolution de l'attaquant dans le système, nous adoptons la notion des attaques atomiques. Lorsqu'un ensemble de préconditions d'une action est satisfait, l'exécution de cette action changera l'état de l'attaquant dans le système. Plus formellement :

Définition F.2 (Attaque atomique)**.** *Une* attaque atomique *est un couple* (ξ_i, τ_i) *où* ξ_i *est une règle exécutée sur le nœud* $\tau_i \in \mathcal{T}$*.*

Une exécution d'attaque est une séquence d'exécution d'attaques atomiques correspondant aux règles exécutées par l'attaquant. Soit P l'ensemble de toutes les exécutions d'attaques dans l'intervalle $[0, \kappa_t^M]$ où κ_t^M représente le temps jusqu'à la deuxième date de maintenance. Nous définissons une exécution d'attaque comme suit :

Définition F.3 (Exécution d'attaque)**.** *Une* exécution d'attaque $p_i \in P$ *est un tuple* $\langle(\Xi_i, \tau_i, q_i), >\rangle$ *où :*

- Ξ_i *représente une règle exécutée sur un nœud* τ_i *;*
- q_i *représente la probabilité d'exécuter* Ξ_i *avec succès ;*
- $>$ *est un ordre strict sur les exécutions d'attaques atomiques (par exemple,* $z_1 > z_2$ *si l'attaque atomique* z_1 *est exécutée avant l'attaque atomique* z_2*).*

Soit $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ l'ensemble des super-sommets où chaque super-sommet v_i représente un état du système. L'état du système est défini comme étant l'état des connexions et les règles d'interactions entre les différents composants du système de contrôle. Par

exemple, un changement de la politique de contrôle d'accès pourra modifier l'état du système en refusant ou en accordant des accès sur les équipements. En plus, dans les systèmes de contrôle critiques, compromettre un ensemble de services pourra activer des mécanismes de sûreté de fonctionnement qui pourront modifier les connexions existantes et changer les règles régissant les interactions entre les équipements. Notons que nous nous appuyons sur l'opérateur pour la définition des paramètres ou des actions qui pourront déclencher des mécanismes qui changerons l'état du système.

Dans chaque super-sommet v_i, soit X_i l'ensemble des sommets $\{x_1^i, x_2^i, ..., x_{N_i}^i\}$ où x_j^i représente un état de l'attaquant quand l'état du système est v_i. $\mathcal{E}$ est un sous-ensemble de $\{\bigcup_i X_i\}^2$. Soit $\Sigma_\mathcal{V}$ et $\Sigma_\mathcal{E}$ deux alphabets finis des étiquettes des super-sommets et des arêtes respectivement. $\Sigma_\mathcal{V}$ représente une description de l'ensemble des états auxquels le système peut passer suite à une action de l'attaquant. $\Sigma_\mathcal{E} = \{$*scan, accès réseau, accès humain, exploit*$\}$. $l_\mathcal{V} : \mathcal{V} \to \Sigma_\mathcal{V}$ et $l_\mathcal{E} : \mathcal{E} \to \Sigma_\mathcal{E}$ sont deux fonctions associant des étiquettes aux super-sommets et aux arêtes respectivement.

Ainsi, nous définissons notre modèle d'exécution d'attaque, qui est un graphe d'attaque, comme suit :

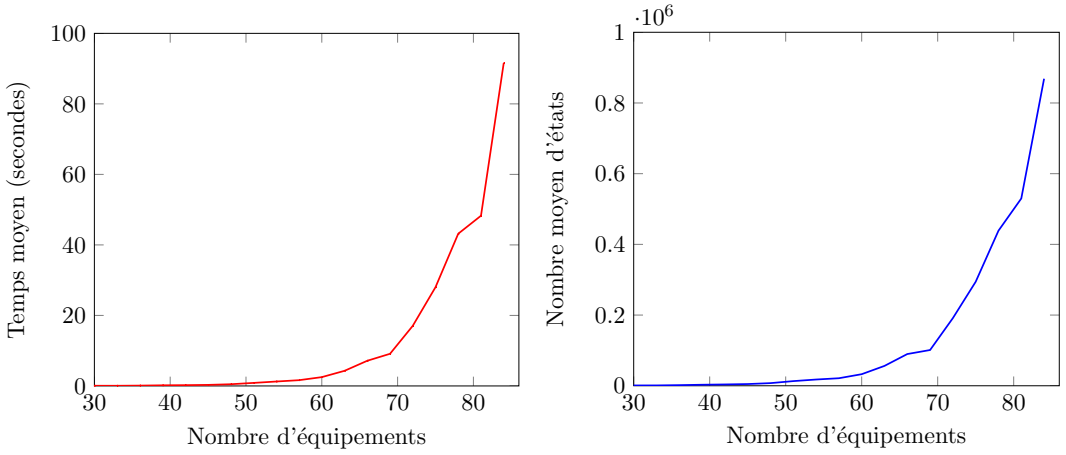
Définition F.4 (Modèle d'exécution d'attaque)**.** *Un* modèle d'exécution d'attaque *est un supergraphe étiqueté représenté par le tuple* $\langle \mathcal{V}, \mathcal{E}, \Sigma_\mathcal{V}, \Sigma_\mathcal{E}, l_\mathcal{V}, l_\mathcal{E} \rangle$.

Notons qu'en pratique, comme la plupart des techniques de génération de graphes d'attaque, nous pouvons simplifier le processus de génération des chemins d'attaque en supposant que l'état du système ne change pas. Par exemple, les actions de l'attaquant ne changent pas la politique de contrôle d'accès et l'état des connexions entre les équipements du réseau. Néanmoins, dans ce cas, il est important d'évaluer l'impact de cette hypothèse sur l'exactitude du graphe d'attaque généré.

F.3.3 Étude de performance

Dans notre évaluation, afin de réduire la complexité de la construction du graphe d'attaque, nous supposons que l'attaquant ne visite plus un équipement qu'il a déjà compromis même partiellement. Nous générons le graphe d'attaque en utilisant une stratégie d'exploration en profondeur. Nous avons implémenté notre algorithme de génération de graphe d'attaque en C++. Pour l'évaluation de la performance, nous avons utilisé un Mac OS X 10.8.5 avec un processeur 2.6 GHz Intel Core i5 et 8 GB de RAM. Nous évaluons la performance de notre outil sur une architecture d'un système de contrôle industriel composé de quatre niveaux hiérarchiques. Les équipements de chaque niveau sont séparés par des passerelles. Dans un même niveau hiérarchique, nous considérons que chaque équipement interagit avec quatre autres équipements et que chaque équipement est vulnérable. Nous supposons le pire des cas dans lequel l'attaquant est capable d'exploiter toutes les vulnérabilités. Puisque l'évaluation des interdépendances entre les services est effectuée avant la phase de génération

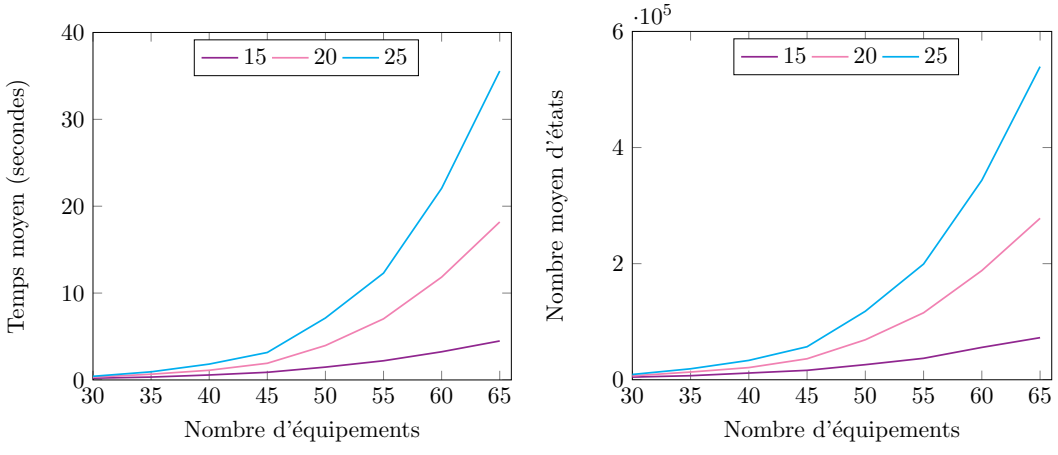
du graphe d'attaque, nous omettons l'existence de la couche service dans notre évaluation pour simplifier l'analyse.



(A) (B)

FIGURE F.2: Évaluation de performances de l'outil de génération du graphe d'attaque sans contraintes sur la longueur des chemins d'attaques

Pour un nombre total fixé d'équipements, nous générons 40 architectures aléatoires et nous faisons la moyenne des résultats en termes du temps nécessaire pour construire le graphe d'attaque et le nombre d'états générés. Nous présentons les résultats dans la Figure F.2 en fonction du nombre total d'équipements dans le système. De manière prévisible, la complexité de la construction de notre graphe d'attaque est exponentielle par rapport au nombre d'équipements dans le système. Par contre, dans la limite du nombre total d'équipements, notre modèle est applicable pour évaluer la sécurité des systèmes de contrôle industriel, qui ont des restrictions similaires.



(A) (B)

FIGURE F.3: Évaluation de performances de l'outil de génération du graphe d'attaque en fonction de la longueur des chemins d'attaque

En pratique, le nombre d'actions qu'un attaquant exécute dans le système aura un impact sur la probabilité de le détecter. Figure F.3 présente le temps moyen de génération du graphe d'attaque et le nombre moyen d'états générés pour 200 configurations aléatoires pour différentes contraintes sur la longueur des chemins d'attaques. Nous pouvons constater que la complexité et le temps nécessaire pour construire le graphe d'attaque diminuent lorsque le nombre d'actions permis dans chaque chemin d'attaque diminue. Notons que dans notre implémentation, nous définissons une équivalence entre deux états du graphe d'attaque en se focalisant sur ce que l'attaquant est capable de réaliser avec l'ensemble des connaissances à sa disposition au lieu de se focaliser sur ce qu'il a déjà réalisé. Cette approche est optimisée pour le cas où nous n'avons pas de contraintes sur le nombre maximale d'actions dans un chemin d'attaque. Par conséquent, nous pouvons constater que le temps nécessaire pour générer le graphe d'attaque sous des longueurs de 20 et 25 pour les chemins d'attaque est supérieur comparé au cas dans lequel nous n'avons pas de telles contraintes (Figure F.2).

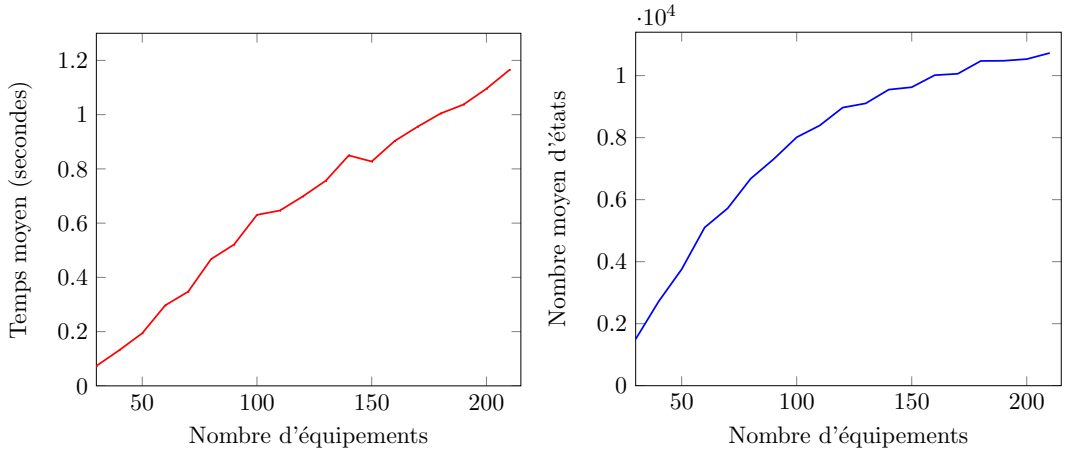


FIGURE F.4: Évaluation de performances de l'outil de génération du graphe d'attaque pour une longueur maximale des chemins d'attaque de 10

Finalement, nous fixons la longueur maximale d'un chemin d'attaque à 10 et nous faisons la moyenne des résultats de 200 configurations aléatoires. Nous présentons le résultat dans la Figure F.4 en fonction du nombre total d'équipements dans le système.

F.4 Optimisation des politiques de sécurité pour les systèmes de contrôle industriel

Des contraintes budgétaires pourraient imposer au défenseur de faire des compromis par rapport au choix de déploiement des équipements de sécurité dans le réseau. Par conséquent, l'optimisation de la distribution des ressources de sécurité dans le système est nécessaire. Un des défis pour sécuriser les systèmes de contrôle industriel est la gestion du processus de patch des vulnérabilités. En général, dans ce type de système, les dates de maintenance

sont planifiées et l'opérateur profite de l'arrêt du système pour effectuer les mises à jour correspondant aux vulnérabilités critiques. Par contre, durant la période de temps entre deux dates de maintenance, le système sera exposé aux menaces des vulnérabilités nouvellement découvertes. Dans ce cas, l'opérateur doit décider s'il arrête le système pour patcher les vulnérabilités critiques, vu leurs impact sur le système, ou attendre jusqu'à la deuxième date de maintenance pour patcher ces vulnérabilités.

La configuration et le déploiement des mesures de défense d'une manière efficace, pour optimiser la détection et minimiser l'impact des attaques, représentent toujours un défi. En particulier, dans les réseaux fournissant des services critiques, le déploiement dépend des types des interdépendances qui peuvent exister entre les équipements vulnérables du système. En plus, en l'absence d'une évaluation des compétences et des préférences de l'attaquant, le déploiement des mesures de sécurité peut ne pas fournir une protection optimale. L'évaluation du profil de l'attaquant, ainsi que les contraintes budgétaires et techniques (par exemple, des équipements et des services critiques qui doivent être protégés ou des mesures de sécurité qui ne peuvent être déployées que sur une partie du système), vont influencer les décisions du défenseur.

En s'appuyant sur les informations présentes dans le graphe d'attaque généré dans la section F.3, le défenseur peut améliorer le processus de prise de décision en choisissant un déploiement de mesures de défense qui limitent l'espace des actions de l'attaquant suivant son état courant dans le système. La politique de sécurité, dans laquelle la mise à jour des vulnérabilités peut être un choix parmi d'autres pour protéger le système, dépendra de l'état de l'attaquant et devra satisfaire un ensemble de contraintes. Par conséquent, le défenseur essaye de répondre à la question suivante : "*Face à une menace d'un attaquant qui a pu avoir un certain accès au système, quelles sont les mesures de sécurité que je dois déployer pour l'empêcher de compromettre des biens critiques et de progresser d'avantage dans le système ?*" Dans ce cas, l'optimisation de la politique de sécurité dépendra de l'incertitude liée aux types d'actions que l'attaquant est en train d'exécuter ou voudra exécuter dans le système. Dans ce chapitre, nous adressons cette problématique et nous présentons une approche basée sur les processus de décision markoviens avec contraintes (PDMC) pour calculer la politique de sécurité qui permet d'offrir la protection optimale.

F.4.1 Approche basée sur les processus de décision markoviens avec contraintes

Le choix de la stratégie de défense du système de contrôle d'une infrastructure critique dépend de l'optimisation des ressources de défense. Selon le contexte et le type du système de contrôle, différents types de contraintes doivent être pris en compte. La stratégie de défense optimale doit parvenir à faire un compromis entre les différentes contraintes (techniques, financières, et environnementales) imposées par l'opérateur du système. Dans cette section, nous nous intéressons au calcul d'une politique de sécurité permettant la protection du système tout en satisfaisant un ensemble de contraintes.

F.4.1.1 Processus de décision markovien avec contraintes

Nous associons un ensemble de mesures de sécurité pour chaque action qui peut être exécuté par l'attaquant. L'objectif du déploiement d'une mesure de sécurité est d'empêcher l'exécution d'une attaque ou de diminuer sa probabilité de succès. Par conséquent, les probabilités de transition entre les différents états dans le graphe d'attaque vont être directement affectées par l'ensemble des mesures de sécurité déployées. Contrairement aux résolutions des problèmes classiques de recherche des politiques optimales pour les processus de décision markoviens dans lesquels nous essayons de minimiser une fonction objective, nous nous intéressons à la recherche d'une politique de sécurité qui satisfait en plus un ensemble de contraintes. Ce type de problème est connu sous le nom de processus de décision markovien avec contraintes (PDMC) [Alt99]. Nous définissons un PDMC comme suit :

Définition F.5 (PDMC)**.** *Un* processus de décision markovien (PDMC) *est un tuple* $\mathcal{J} = \langle \mathcal{X}, \mathcal{A}, \mathcal{P}, \omega, c, \beta \rangle$ *où :*

- $\mathcal{X}$ *est un ensemble fini d'états ;*
- $\mathcal{A}$ *est un ensemble fini d'actions. Soit* $\mathcal{A}(x)$ *l'ensemble d'actions disponible dans l'état* x *et soit* $\mathcal{Q} = \{(x, a) : x \in \mathcal{X}, a \in \mathcal{A}(x)\}$ *l'ensemble des paires état-action ;*
- $\mathcal{P} : \mathcal{Q} \times \mathcal{X} \to [0, 1]$ *est la fonction de transition.* $\mathcal{P}(x, a, y)$ *représente la probabilité de se trouver dans l'état* y *en effectuant l'action* a *sachant que nous étions dans l'état* x *; si* $a \in \mathcal{A}(x)$*, nous avons* $\sum_{y \in \mathcal{X}} \mathcal{P}(x, a, y) = 1$ *;*
- $\omega : \mathcal{Q} \to \mathbb{R}^+$ *est le coût immédiat ;*
- $c : \mathcal{Q} \to \mathbb{R}^{\mathrm{M}}$ *est un vecteur à* M *dimensions de coûts immédiats, relatif aux* M *contraintes ;*
- $\beta \in \mathbb{P}(\mathcal{X})$ *est la distribution initiale de probabilité sur l'état initial où* $\mathbb{P}(\mathcal{X})$ *représente l'ensemble des distributions de probabilités sur* $\mathcal{X}$*. Par exemple, initialement à* $t = 0$*, la probabilité d'être dans l'état* x *est* $\beta(x)$*.*

Définition F.6 (PDMC étiqueté)**.** *Un* processus de décision markovien étiqueté *est un tuple* $\mathcal{J}' = \langle \mathcal{X}, \mathcal{A}, \mathcal{P}, L, \omega, c, \beta \rangle$ *où* $\langle \mathcal{X}, \mathcal{A}, \mathcal{P}, \omega, c, \beta \rangle$ *est un PDMC et* $L : \mathcal{X} \to \{Critique, Non\ critique\}$ *est une fonction d'étiquetage.*

Un état x représente un état de l'attaquant dans le système. Cet état est étiqueté *critique* si l'attaquant est capable de compromettre des équipements ou des services critiques. Dans le PDMC étiqueté, $\mathcal{A}(x)$ représente l'ensemble des mesures de sécurité disponible au défenseur dans l'état x. Dans la suite, nous faisons référence au PDMC étiqueté par PDMCE.

Nous observons le système aux intervalles de temps $t = 1, 2, ..., n$ où n représente l'horizon de temps (il peut être fini ou infini). Les actions du défenseur dans chaque état sont choisies

selon une règle de décision que nous appelons une politique. Les actions peuvent être choisies selon une distribution de probabilité.

Définition F.7 (Politique)**.** *Soit* $h_t = (x_1, a_1, ..., x_{t-1}, a_{t-1}, x_t)$ *l'historique au temps* t *représentant la séquence des états et des actions précédentes. Une* politique $u = (u_1, u_2, ..., u_n)$ *est une séquence où* $u_t(a'|h_t)$ *représente la probabilité de choisir l'action* a' *au temps* t *si l'historique observé est* h_t*. Soit* $\mathcal{U}$ *la classe de toutes les politiques de ce type.*

Nous identifions en général trois classes de politiques. Une *politique markovienne* est une politique dans laquelle la décision de choisir une action au temps t dans l'état x_t dépend seulement de x_t. Une politique est *stationnaire* si la décision de choisir une action dépend seulement de l'état x et est indépendante du temps t. Finalement, une *politique stationnaire déterministe* est une politique dans laquelle nous choisissons, pour chaque état x, une action $a \in \mathcal{A}(x)$ avec une probabilité 1.

F.4.1.2 Fonction de coût et mesure d'occupation

Étant données une distribution initiale de probabilité sur l'état initial β et une politique u, nous définissons la fonction coût $\Omega_\alpha(\beta, u)$ pour un horizon de temps n comme suit :

$$\Omega_\alpha(\beta, u) = (1-\alpha)\sum_{t=1}^{n} \alpha^{t-1} E^u_\beta \omega(\mathcal{X}_t, \mathcal{A}_t) \tag{F.1}$$

E^u_β représente l'espérance et $\mathcal{X}_t$ et $\mathcal{A}_t$ représentent les processus stochastiques des états et des actions respectivement. Soit $\alpha \in (0, 1)$ le facteur d'actualisation. Le paramètre α représente le fait que nous donnons moins d'importance au futur par rapport au présent (du fait des incertitudes liées au futur).

D'une manière similaire, nous définissons la fonction de coût liée aux M contraintes $C^k_\alpha(\beta, u)\ \forall k = 1, ..., M$ pour un horizon de temps fini comme suit :

$$C^k_\alpha(\beta, u) = (1-\alpha)\sum_{t=1}^{n} \alpha^{t-1} E^u_\beta c^k(\mathcal{X}_t, \mathcal{A}_t) \tag{F.2}$$

La politique de sécurité optimale est la solution du problème d'optimisation suivant :

$$\min_{u \in U} \Omega_\alpha(\beta, u) \text{ avec } C_\alpha(\beta, u) \leq S \tag{F.3}$$

où $S = (s^1, ..., s^M)$ représente le vecteur à M dimensions correspondant aux contraintes sur la fonction coût $C_\alpha(\beta, u)$.

Pour une distribution β et une politique u, soit $\rho(x,a)$ la mesure d'occupation donnée par :

$$\rho(x,a) = (1-\alpha)\sum_{t=1}^{\infty}\alpha^{t-1}P_{\beta}^{u}(\mathcal{X}_t = x, \mathcal{A}_t = a),\ \forall x \in \mathcal{X},\ a \in \mathcal{A}(x) \tag{F.4}$$

Par conséquent, $\Omega_\alpha(\beta,u)$ et $C_\alpha^k(\beta,u)$ peuvent être écrits sous les formes suivantes :

$$\Omega_\alpha(\beta,u) = \sum_{x\in\mathcal{X}}\sum_{a\in\mathcal{A}(x)}\rho(x,a)\omega(x,a) \tag{F.5}$$

$$C_\alpha^k(\beta,u) = \sum_{x\in\mathcal{X}}\sum_{a\in\mathcal{A}(x)}\rho(x,a)c^k(x,a) \tag{F.6}$$

F.4.1.3 Problème d'optimisation

Pour résoudre le problème d'optimisation défini dans l'équation F.3, nous le formulons sous la forme d'un problème de programmation linéaire primal [Alt99] comme suit :

$$\begin{gathered}
\min_{\rho}\sum_{x\in\mathcal{X}}\sum_{a\in\mathcal{A}(x)}\rho(x,a)\omega(x,a) \qquad (F.7)\\
\text{avec}\sum_{x\in\mathcal{X}}\sum_{a\in\mathcal{A}(x)}\rho(x,a)c^i(x,a) \le s^i \quad \forall i = 1,...,M\\
\text{et } \forall x \in \mathcal{X},\ a \in \mathcal{A}(x),\\
\sum_{a\in\mathcal{A}(x)}\rho(x,a) - \alpha\sum_{y\in\mathcal{X}}\sum_{a\in\mathcal{A}(y)}\rho(y,a)\mathcal{P}(y,a,x) = (1-\alpha)\beta(x)\\
\rho(x,a) \ge 0
\end{gathered}$$

Pour chaque état $x \in \mathcal{X}$ tel que $\sum_{a\in\mathcal{A}(x)}\rho(x,a) > 0$, soit $b_x(a)$ la politique stationnaire suivante :

$$b_x(a) = \frac{\rho(x,a)}{\sum_{a'\in\mathcal{A}(x)}\rho(x,a')} \quad \forall a \in \mathcal{A}(x) \tag{F.8}$$

Théorème F.3. *Le résolution du problème d'optimisation défini dans l'équation F.3 est possible si et seulement si la résolution du problème d'optimisation défini dans l'équation F.7 est possible.*

Le théorème F.3 se déduit directement du Théorème 3.3 dans [Alt99]. En plus, à partir du théorème 3.3 [Alt99], il existe une solution optimale ρ^* pour le problème défini dans l'équation F.7 si la résolution du problème d'optimisation défini dans l'équation F.3 est possible, et dans ce cas, la politique stationnaire définie dans l'équation F.8 est optimale pour le problème d'optimisation défini dans l'équation F.3.

F.4.2 Construction du PDMCE

Nous proposons deux algorithmes pour la construction du PDMCE en s'appuyant sur les informations présentes dans le graphe d'attaque généré dans la section F.3. Il existe deux interprétations du PDMCE selon la sémantique associée aux états. Un état dans le PDMCE peut représenter un état de l'attaquant dans le système. Dans ce cas, les actions associées à cet état représentent l'ensemble des mesures de sécurité qui permet de protéger le système et d'empêcher l'attaquant de compromettre de nouveaux équipements. Dans ce type de représentation, nous pouvons constater que lorsque le nombre d'actions que l'attaquant peut exécuter avec le même ensemble de connaissances augmente, le nombre de combinaisons des mesures de sécurité qui doivent être prises en compte augmente et ainsi la complexité de construire ce type de PDMCE. Nous pouvons aussi interpréter la sémantique des états dans le PDMCE différemment. Dans un état, l'attaquant est en train ou voudrait exécuter une action. Dans ce cas, nous associons à cet état un ensemble de mesures de sécurité qui permet d'empêcher l'attaquant d'exécuter cette action. Si la mesure de sécurité choisie par le défenseur est efficace, l'attaque échoue. Sinon, si l'attaque réussit, nous passons avec une certaine probabilité vers un autre état du système dans lequel l'attaquant essaye d'exécuter une autre action. Une des deux interprétations précédentes s'avérera plus avantageuse suivant l'ensemble des mesures de sécurité disponibles et le nombre de défenseurs qui vont les déployer.

F.4.3 Recommandations optimales de défense

Dans cette section, nous présentons les différentes interprétations et scénarios d'utilisation de la solution du problème d'optimisation défini dans l'équation F.7 pour un PDMCE donné.

F.4.3.1 Résponse optimale aux intrusions

La solution du problème d'optimisation défini dans l'équation F.7, si elle existe, est une politique stochastique qui garantit que les objectifs de sécurité définis par le défenseur sont atteints. Cette politique peut être transformée en une politique stationnaire (équation F.8) afin de trouver une solution pour le problème d'optimisation initial défini dans l'équation F.3. Connaissant l'état courant de l'attaquant, le modèle peut ainsi être utilisé comme un système d'aide à la décision qui permet au défenseur de répondre aux intrusions d'une manière efficace en minimisant les risques de sécurité sur le système.

F.4.3.2 Classement des mesures de sécurité

La solution du problème d'optimisation défini dans l'équation F.7, si elle existe, donnera une distribution de probabilité sur la mesure d'occupation ρ. Soit a^i la mesure de sécurité déployée dans un endroit i dans le système. Par exemple, si nous avons le même type de mesure de sécurité déployé sur les équipements τ_1 et τ_2, nous faisons référence à la mesure de sécurité déployée sur chacun de ces équipements par a^1 et a^2 respectivement. Pour chaque mesure de sécurité a^i, soit $\mathcal{X}(a^i)$ l'ensemble des états dans $\mathcal{X}$ dans lesquels a^i peut être déployé et soit $\rho(a^i) = \frac{\sum_{x \in \mathcal{X}(a^i)} \rho(x, a^i)}{1 - \sum_{x \in \mathcal{X}} \rho(x, a_0)}$. En conséquence, nous avons $\sum_{a^i \in \mathcal{A}, a \neq a_0} \rho(a^i) = 1$. La distribution de probabilité sur l'ensemble des mesures de sécurité disponibles peut être interprétée comme étant un classement pour le déploiement de ces mesures de sécurité dans le système. Par exemple, la mesure de sécurité a^i ayant le classement le plus élevé (meilleur $\rho(a^i)$) peut être considérée comme la plus urgente devant être déployée. Ce classement est important lorsque nous sommes en train d'anticiper une menace quand il existe une contrainte sur le nombre de défenseur disponible. Dans ce cas, prioriser le déploiement des mesures de sécurité devient une tâche importante pour protéger le système.

F.4.3.3 Comparaison de la sécurité relative de deux architectures

Lorsque nous voulons comparer deux architectures ou configurations de sécurité d'un système de contrôle industriel, les graphes d'attaques générés de ces deux architectures ou configurations et les politiques de sécurité optimales des PDMCE associées nous offrent un aperçu important pour évaluer leurs niveaux de sécurité relatifs. Par exemple, le budget d'attaque minimal, le temps pour effectuer l'attaque, et le nombre d'actions nécessaires pour compromettre un équipement ou un service critique dans chaque architecture peuvent être considérés comme des critères de base pour la comparaison. De plus, la sécurité d'une architecture peut être évaluée selon le coût minimal nécessaire pour le déploiement des mesures de sécurité. En revanche, étant donnée une contrainte sur le budget de défense, il est possible de calculer le risque résiduel des attaques sur le système après le déploiement des mesures de sécurité. Dans ce cas, selon les ressources de défense disponibles, la meilleure architecture ou configuration est celle qui minimise le risque résiduel des attaques.

Bibliography

[AAG09] E. Altman, K. Avrachenkov, and A. Garnaev. Jamming in wireless networks: The case of several jammers. In *International Conference on Game Theory for Networks (GameNets)*, pages 585–592, 2009.

[AB06] T. Alpcan and T. Başar. An intrusion detection game with limited observations. In *International Symposium on Dynamic Games and Applications*, pages 697–703, 2006.

[AB09] T. Alpcan and N. Bambos. Modeling dependencies in security risk management. In *Proceedings of the 4th International Conference on Risks and Security of Internet and Systems (CRISIS)*, 2009.

[AB10] T. Alpcan and T. Başar. *Network Security: A Decision and Game-Theoretic Approach*. Cambridge University Press, 2010.

[ADMT08] G. Ateniese, R. Di Pietro, L.V. Mancini, and G. Tsudik. Scalable and efficient provable data possession. In *Proceedings of the 4th International Conference on Security and Privacy in Communication Networks*, page 9, 2008.

[AFG+10] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[AJN12] M. Albanese, S. Jajodia, and S. Noel. Time-efficient and cost-effective network hardening using attack graphs. In *42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 1–12, 2012.

[AJPS11] M. Albanese, S. Jajodia, A. Pugliese, and V.S. Subrahmanian. Scalable analysis of attack scenarios. In *Proceedings of the 16th European Conference on Research in Computer Security (ESORICS)*, pages 416–433. Springer-Verlag, 2011.

[Alt99] E. Altman. *Constrained Markov Decision Processes*. Chapman & Hall/CRC, 1999.

[AMGC09] Z. Anwar, M. Montanari, A. Gutierrez, and R. H. Campbell. Budget constrained optimal security hardening of control networks for critical cyber-infrastructures. *International Journal of Critical Infrastructure Protection*, 2(1):13–25, 2009.

[Ami01] M. Amin. Toward self-healing energy infrastructure systems. *IEEE Computer Applications in Power*, 14(1):20–28, 2001.

[ANS10] ANSSI. EBIOS (Expression of Needs and Identification of Security Objectives) Risk Management Method, 2010. URL: http://www.ssi.gouv.fr/IMG/pdf/EBIOS-1-GuideMethodologique-2010-01-25.pdf.

[ASH13] S. Amin, G.A. Schwartz, and A. Hussain. In quest of benchmarking security risks to cyber-physical systems. *IEEE Network*, 27(1):19–24, 2013.

[AWK02] P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, graph-based network vulnerability analysis. In *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS)*, pages 217–224, 2002.

[AYWO15] H. Almohri, D. Yao, L. Watson, and X. Ou. Security optimization of dynamic networks with probabilistic graph modeling and linear programming. *IEEE Transactions on Dependable and Secure Computing*, PP(99), 2015.

[Bal94] Robert W. Baldwin. Su-kuang: Rule-based security checking. Technical Report Programming Systems Research Group, Lab. for Computer Science, MIT, 1994.

[Bas93] R. Baskerville. Information systems security design methods: Implications for information systems development. *ACM Computing Surveys*, 25(4):375–414, 1993.

[BB03] M. Bouissou and J.-L. Bon. A new formalism that combines advantages of fault-trees and markov models: Boolean logic driven Markov processes. *Reliability Engineering & System Safety*, 82(2):149–163, 2003.

[BCE+08] M. Belenkiy, M. Chase, C. C. Erway, J. Jannotti, A. Küpçü, and A. Lysyanskaya. Incentivizing outsourced computation. In *Proceedings of the Third International Workshop on Economics of Networked Systems*, pages 85–90, 2008.

[BCG+12] M. Beccuti, S. Chiaradonna, F.D. Giandomenico, S. Donatelli, G. Dondossola, and G. Franceschinis. Quantification of dependencies between electrical and information infrastructures. *International Journal of Critical Infrastructure Protection*, 5(1):14–27, 2012.

[BCP+10] R. Bloomfield, N. Chozos, P.T. Popov, V. Stankovic, D. Wright, and R. Howell-Morris. Preliminary interdependency analysis (PIA): Method and tool support. Technical Report D/501/12102/2, 2010.

[BFM04] E.J. Byres, M. Franz, and D. Miller. The use of attack trees in assessing vulnerabilities in SCADA systems. In *International Infrastructure Survivability Workshop (IISW)*, 2004.

[BFP06] S. Bistarelli, F. Fioravanti, and P. Peretti. Defense trees for economic evaluation of security investments. In *First International Conference on Availability, Reliability and Security (ARES)*, pages 416–423, 2006.

[BKH10] A. Bensoussan, M. Kantarcioglu, and S.C. Hoe. A game-theoretical approach for finding optimal strategies in a botnet defense model. In *Proceedings of the First International Conference on Decision and Game Theory for Security (GameSec)*, pages 135–148. 2010.

[BMRL03] I. Balepin, S. Maltsev, J. Rowe, and K. Levitt. Using specification-based intrusion detection for automated response. In *Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection (RAID)*, pages 136–154, 2003.

[BNT07] K. Burbeck and S. Nadjm-Tehrani. Adaptive real-time anomaly detection with incremental clustering. *Information Security Technical Report*, 12(1):56–67, 2007.

[BPP+10] S.V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464:1025–1028, 2010.

[BS11] R. Berthier and W.H. Sanders. Specification-based intrusion detection for advanced metering infrastructures. In *Proceedings of the 17th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 184–193, 2011.

[BSP12] R.R.R Barbosa, R. Sadre, and A. Pras. Difficulties in modeling SCADA traffic: A comparative analysis. In *Proceedings of the 13th International Conference on Passive and Active Measurement*, pages 126–135, 2012.

[BSP13] R.R.R. Barbosa, R. Sadre, and A. Pras. Flow whitelisting in SCADA networks. *International Journal of Critical Infrastructure Protection*, 6(3–4):150–158, 2013.

[Bur08] Elie Bursztein. *Anticipation Games.* PhD thesis, Ecole Normale Supérieure de Cachan, 2008.

[Byr13] E. Byres. The air gap: SCADA's enduring security myth. *Communications of the ACM*, 56(8):29–31, 2013.

[BZ11] S. Bi and Y.J. Zhang. Defending mechanisms against false-data injection attacks in the power system state estimation. In *IEEE International Workshop*

on Smart Grid Communications and Networks, GLOEBECOM, pages 1162–1167, 2011.

[Car01] Curtis A. Carver. *Adaptive Agent-Based Intrusion Response*. PhD thesis, Texas A&M University, 2001.

[CCZ08] H. Cavusoglu, H. Cavusoglu, and J. Zhang. Security patch management: Share the burden or share the damage? *Management Science*, 54(4):657–670, 2008.

[CDF+07] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes. Using model-based intrusion detection for SCADA networks. In *Proceedings of the SCADA Security Scientific Symposium*, 2007.

[CDN11] S. Chiaradonna, F. Di Giandomenico, and N. Nostro. Modeling and analysis of the impact of failures in electric power systems organized in interconnected regions. In *IEEE/IFIP 41st International Conference on Dependable Systems Networks (DSN)*, pages 442–453, 2011.

[Cen] Centre for the Protection of National Infrastructure. URL: http://www.cpni.gov.uk.

[CGT07] E. Casalicchio, E. Galli, and S. Tucci. Federated agent-based modeling and simulation approach to study interdependencies in it critical infrastructures. In *Proceedings of the 11th IEEE International Symposium on Distributed Simulation and Real-Time Applications*, pages 182–189, 2007.

[CKBA08] R. Curtmola, O. Khan, R. Burns, and G. Ateniese. Mr-pdp: Multiple-replica provable data possession. In *Proceedings of the 28th International Conference on Distributed Computing Systems (ICDCS)*, pages 411–420, 2008.

[CL09] L. Chen and J. Leneutre. A game theoretical framework on intrusion detection in heterogeneous networks. *IEEE Transactions on Information Forensics and Security*, 4(2):165–178, 2009.

[Cle08] F. M. Cleveland. Cyber security issues for advanced metering infrastructure (AMI). In *IEEE Power and Energy Society General Meeting*, 2008.

[CM02] F. Cuppens and A. Miege. Alert correlation in a cooperative intrusion detection framework. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 202–215, 2002.

[CN06] M. Cremonini and D. Nizovtsev. Understanding and influencing attackers' decisions: Implications for security investment strategies. In *Proceedings of the 5th Annual Workshop on Economics and Information Security (WEIS)*, 2006.

[CO00] F. Cuppens and R. Ortalo. LAMBDA: A language to model a database for detection of attacks. In *Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection (RAID)*, pages 197–216, 2000.

[CRY08] H. Cavusoglu, S. Raghunathan, and W. Yue. Decision-theoretic and game-theoretic approaches to IT security investment. *Journal of Management Information Systems*, 25(2):281–304, 2008.

[CS11] Y. Chen and R. Sion. To cloud or not to cloud? Musings on costs and viability. In *Proceedings of the Second ACM Symposium on Cloud Computing (SOCC)*, 2011.

[CSAB11] T.M. Chen, J.C. Sanchez-Aarnoutse, and J. Buford. Petri net modeling of cyber-physical attacks on smart grid. *IEEE Transactions on Smart Grid*, 2(4):741–749, 2011.

[CSW12] L. Changwei, A. Singhal, and D. Wijesekera. Using attack graphs in forensic examinations. In *Proceedings of the 7th International Conference on Availability, Reliability and Security (ARES)*, pages 596–603, 2012.

[Dac94] Marc Dacier. *Towards a quantitative evaluation of computer security.* PhD thesis, Report LAAS n° 94488, Institut National Polytechnique de Toulouse, 1994.

[DCH02] J. Dawkins, C. Campbell, and J. Hale. Modeling network attacks: Extending the attack tree paradigm. In *Proceedings of the Workshop on Statistical and Machine Learning Techniques in Computer Intrusion Detection*, 2002.

[DD94] M. Dacier and Y. Deswarte. Privilege graph: An extension to the typed access matrix model. In *Proceedings of the Third European Symposium on Research in Computer Security (ESORICS)*, pages 319–334, 1994.

[DDK96] M. Dacier, Y. Deswarte, and M. Kaaniche. Quantitative assessment of operational security: Models and tools. Technical Report 96493, LAAS, 1996.

[Dep] U.S. Department of Homeland Security. Smart grid initiatives. URL: http://www.smartgrid.gov/federal_initiatives/federal_smart_grid_task_force/department_of_homeland_security, [accessed on 27.10.2015].

[Dep09a] Department of Homeland Security. Primer control systems cyber security framework and technical metrics. Technical report, 2009.

[Dep09b] U.S. Department of Homeland Security. National infrastructure protection plan, 2009. URL: http://www.dhs.gov/xlibrary/assets/NIPP_Plan.pdf.

[Din70] Y. Dinitz. An algorithm for the solution of the max-flow problem with the polynomial estimation. *Soviet Mathematics Doklady*, 11:1277–1280, 1970.

[DJOR12] M. van Dijk, A. Juels, A. Oprea, and R. L. Rivest. FlipIt: The game of "stealthy takeover". Cryptology ePrint Archive, Report 2012/103, 2012.

[DKLC14] B. Djebaili, C. Kiennert, J. Leneutre, and L. Chen. Data integrity and availability verification game in untrusted cloud storage. In *Proceedings of the 5th International Conference on Decision and Game Theory for Security (GameSec)*, pages 287–306, 2014.

[DLBK15a] K. Durkota, V. Lisy, B. Bosansky, and C. Kiekintveld. Approximate solutions for attack graph games with imperfect information. In *Proceedings of the 6th Conference on Decision and Game Theory for Security (GameSec)*, pages 228–249, 2015.

[DLBK15b] K. Durkota, V. Lisy, B. Bosansky, and C. Kiekintveld. Optimal network security hardening using attack graph games. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 7–14, 2015.

[DNP] DNP3 Users Group. URL: http://www.dnp.org.

[DPRW07] R. Dewri, N. Poolsappasit, I. Ray, and D. Whitley. Optimal security hardening using multi-objective optimization on attack tree models of networks. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS)*, pages 204–213, 2007.

[DRPW12] R. Dewri, I. Ray, N. Poolsappasit, and D. Whitley. Optimal security hardening on attack tree models of networks: A cost-benefit analysis. *International Journal of Information Security*, 11(3):167–188, 2012.

[EK10] C. Efthymiou and G. Kalogridis. Smart grid privacy via anonymization of smart metering data. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 238–243, 2010.

[EKPT09] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia. Dynamic provable data possession. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS)*, pages 213–222, 2009.

[Ele] Electricité Réseau Distribution France (ERDF). URL: http://www.erdf.fr/linky-le-compteur-communicant-derdf.

[Ene13] U.S. Energy Information Administration. International energy outlook, July 2013. URL: http://www.eia.gov/forecasts/ieo/pdf/0484(2013).pdf.

[ENZH11] M. Esmalifalak, H. Nguyen, R. Zheng, and Z. Han. Stealth false data injection using independent component analysis in smart grid. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 244–248, 2011.

[ES09] M. Ekstedt and T. Sommestad. Enterprise architecture models for cyber security analysis. In *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*, pages 1–6, 2009.

[Eur12a] European Network and Information Security Agency. Appropriate security measures for smart grids, Guidelines to assess the sophistication of security measures implementation, December 2012.

[Eur12b] European Network and Information Security Agency. Smart grid security: Annex I. General concepts and dependencies with ICT, 2012.

[Eur12c] European Network and Information Security Agency. Smart grid security: Annex II. Security aspects of the smart grid, 2012.

[Far91] B. Farquhar. One approach to risk assessment. *Computers & Security*, 10(1):21–23, 1991.

[FF05] J.D. Fernandez and A.E. Fernandez. SCADA systems: Vulnerabilities and remediation. *Journal of Computing Sciences in Colleges*, 20(4):160–168, 2005.

[FMC11] N. Falliere, L.O. Murchu, and E. Chien. W32.stuxnet Dossier, Version 1.4, February 2011. URL: http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf.

[FWM+05] B. Foo, Y.-S. Wu, Y.-C. Mao, S. Bagchi, and E. Spafford. ADEPTS: Adaptive intrusion response using attack graphs in an e-commerce environment. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN)*, pages 508–517, 2005.

[FWSJ08] M. Frigault, L. Wang, A. Singhal, and S. Jajodia. Measuring network security using dynamic bayesian network. In *Proceedings of the 4th ACM Workshop on Quality of Protection*, pages 23–30, 2008.

[Gar89] P. E. Gardner. Evaluation of five risk assessment programs. *Computers & Security*, 8(6):479–485, 1989.

[GCC08] J. Grossklags, N. Christin, and J. Chuang. Secure or insure? A game-theoretic analysis of information security games. In *Proceedings of the 17th International World Wide Web Conference (WWW)*, pages 209–218, 2008.

[GGP10] R. Gennaro, C. Gentry, and B. Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *Advances in Cryptology (CRYPTO)*, pages 465–482. 2010.

[Gil08] G. Gilchrist. Secure authentication for DNP3. In *IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pages 1–3, 2008.

[GJ09] J. Grossklags and B. Johnson. Uncertainty in the weakest-link security game. In *International Conference on Game Theory for Networks (GameNets)*, pages 673–682, 2009.

[GJC10] J. Grossklags, B. Johnson, and N. Christin. The price of uncertainty in security games. In *Economics of Information Security and Privacy*, pages 9–32. Springer US, 2010.

[GJDC12] G.G. Granadillo, G. Jacob, H. Debar, and L. Coppolino. Combination approach to select optimal countermeasures based on the RORI index. In *Proceedings of the Second International Conference on Innovative Computing Technology (INTECH)*, pages 38–45, 2012.

[GK04] A. Gehani and G. Kedem. Rheostat: Real-time risk management. In *Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection (RAID)*, pages 296–314. Springer Berlin Heidelberg, 2004.

[GM12] A. Gueye and V. Marbukh. A game-theoretic framework for network security vulnerability assessment and mitigation. In *Proceedings of the Third Conference on Decision and Game Theory for Security (GameSec)*, pages 186–200. 2012.

[Gor09] S. Gorman. Electricity grid in U.S. penetrated by spies, April 2009. URL: http://online.wsj.com/article/SB123914805204099085.html.

[GRCC06] M. Gupta, J. Rees, A. Chaturvedi, and J. Chi. Matching information security vulnerabilities to organizational security profiles: A genetic algorithm approach. *Decision Support Systems*, 41(3):592–603, 2006.

[GT11] L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning*. MIT Press, 2011.

[Gua87] S. B. Guarro. Principles and procedures of the LRAM approach to information systems risk analysis and management. *Computers & Security*, 6(6):493–504, 1987.

[GW13] N. Goldenberg and A. Wool. Accurate modeling of modbus/tcp for intrusion detection in SCADA systems. *International Journal of Critical Infrastructure Protection*, 6(2):63–75, 2013.

[HGB+11] X. Huang, J. Gao, S. Buldyrev, S. Havlin, and H. Stanley. Robustness of interdependent networks under targeted attack. *Physical Review E*, 83:065101, 2011.

[HLCH11] Y. Huang, H. Li, K.A. Campbell, and Z. Han. Defending false data injection attack on smart grid network using adaptive CUSUM test. In *Proceedings of 45th IEEE Annual Conference on Information Sciences and Systems (CISS)*, 2011.

[Hol14] H. Holm. A large-scale study of the time required to compromise a computer system. *IEEE Transactions on Dependable and Secure Computing*, 11(1):2–15, 2014.

[Hop12] N. Hopkins. Hostile states using cyberwarfare to attack UK infrastructure, December 2012. URL: http://www.guardian.co.uk/technology/2012/dec/03/hostile-states-cyberwarfare-uk-infrastructure.

[HSH11] M.M. Hassan, B. Song, and E.-N. Huh. Distributed resource allocation games in horizontal dynamic cloud federation platform. In *Proceedings of the 13th International Conference on High Performance Computing and Communications (HPCC)*, pages 822–827, 2011.

[Hub09] D. Hubbard. *The Failure of Risk Management: Why Its Broken and How to Fix It*. Wiley, 2009.

[ICL+09] K. Ingols, M. Chu, R. Lippmann, S. Webster, and S. Boyer. Modeling modern network attacks and countermeasures using attack graphs. In *Proceedings of the 25th Annual Computer Security Applications Conference (ACSAC)*, pages 117–126, 2009.

[ICS12] ICS-CERT. Key management erros in RuggedCom's rugged operating system, August 2012. URL: http://www.us-cert.gov/control_systems/pdf/ICS-ALERT-12-234-01.pdf.

[ICS14] ICS-CERT. Year in review, 2014. URL: https://ics-cert.us-cert.gov/sites/default/files/Annual_Reports/Year_in_Review_FY2014_Final.pdf.

[ICS15] ICS-CERT. NCCIC/ICS-CERT Monitor, September 2014–February 2015. URL: https://ics-cert.us-cert.gov/sites/default/files/Monitors/ICS-CERT_Monitor_Sep2014-Feb2015.pdf.

[IECa] IEC 61850. Standard: Communication networks and systems in substations. URL: http://www.iec.ch.

[IECb] IEC 62351. Standard: Data and communication security. URL: http://www.iec.ch.

[ILBC14] Z. Ismail, J. Leneutre, D. Bateman, and L. Chen. A game theoretical analysis of data confidentiality attacks on smart grid AMI. *IEEE Journal on Selected Areas in Communications*, 32(7):1486–1499, 2014.

[ILBC15] Z. Ismail, J. Leneutre, D. Bateman, and Lin Chen. A game-theoretical model for security risk management of interdependent ict and electrical infrastructures. In *Proceedings of the IEEE 16th International Symposium on High Assurance Systems Engineering (HASE)*, pages 101–109, 2015.

[ILF15] Z. Ismail, J. Leneutre, and A. Fourati. An attack execution model for industrial control systems security assessment. In *Proceedings of the First Conference on Cybersecurity of Industrial Control Systems (CyberICS)*, 2015.

[ILP06] K. Ingols, R. Lippmann, and K. Piwowarski. Practical attack graph generation for network defense. In *Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC)*, pages 121–130, 2006.

[JB14] S. Jegelka and J. Bilmes. Graph cuts with interacting edge costs - examples, approximations, and algorithms, 2014. *CoRR*, abs/1402.0240. URL http://arxiv.org/abs/1402.0240.

[JGCC10] B. Johnson, J. Grossklags, N. Christin, and J. Chuang. Are security experts useful? Bayesian nash equilibria for network security games with limited information. In *Proceedings of the 15th European Symposium on Research in Computer Security (ESORICS)*, pages 588–606, 2010.

[JK07] A. Juels and B. Kaliski. PORs: Proofs of retrievability for large files. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS)*, pages 584–597, 2007.

[JN10] S. Jajodia and S. Noel. Topological vulnerability analysis. In S. Jajodia, P.Liu, V. Swarup, and C. Wang, editors, *Cyber Situational Awareness*, volume 46 of *Advances in Information Security*, pages 139–154. Springer US, 2010.

[JNO03] S. Jajodia, S. Noel, and B. O'Berry. Topological analysis of network attack vulnerability. In *Managing Cyber Threats: Issues, Approaches and Challenges*. Kluwer Academic Publisher, 2003.

[JNY11] D. Jin, D.M. Nicol, and G. Yan. An event buffer flooding attack in DNP3 controlled SCADA systems. In *Proceedings of the 2011 Winter Simulation Conference (WSC)*, pages 2614–2626, 2011.

[JSW02] S. Jha, O. Sheyner, and J. Wing. Two formal analyses of attack graphs. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop*, pages 49–63, 2002.

[KCBCD10] N. Kheir, N. Cuppens-Boulahia, F. Cuppens, and H. Debar. A service dependency model for cost-sensitive intrusion response. In *Proceedings of the 15th European Symposium on Research in Computer Security (ESORICS)*, pages 626–642, 2010.

[KED+10] G. Kalogridis, C. Efthymiou, S. Denic, T. Lewis, and R. Capeda. Privacy for smart meters: Towards undetectable appliance load signatures. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 232–237, 2010.

[KH03] H. Kunreuther and G. Heal. Interdependent security. *Journal of Risk and Uncertainty*, 26(2):231–249, 2003.

[KJTT10] O. Kosut, L. Jia, R.J. Thomas, and L. Tong. Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 220–225, 2010.

[KMRS11] B. Kordy, S. Mauw, S. Radomirović, and P. Schweitzer. Foundations of attack-defense trees. In *Proceedings of the 7th International Conference on Formal Aspects of Security and Trust (FAST)*, pages 80–95, 2011.

[KPC14] M. Khouzani, V. Pham, and C. Cid. Incentive engineering for outsourced computation in the face of collusion. In *Proceedings of the 13th Annual Workshop on the Economics of Information Security*. 2014.

[KT79] D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979.

[KT84] D. Kahneman and A. Tversky. Choices, values, and frames. *American Psychologist*, 39(4):341–350, 1984.

[KW14] A. Kochumol and M.J. Win. Proving possession and retrievability within a cloud environment: A comparative survey. *International Journal of Computer Science and Information Technologies*, 5(1):478–485, 2014.

[LAP12] Y. W. Law, T. Alpcan, and M. Palaniswami. Security games for voltage control in smart grid. In *Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing*, pages 212–219, 2012.

[LB08] D.J. Leversage and E.J. Byres. Estimating a system's mean time-to-compromise. *IEEE Security & Privacy*, 6(1):52–60, 2008.

[LFB15] A. Laszka, M. Felegyhazi, and L. Buttyan. A survey of interdependent information security games. *ACM Computing Surveys*, 47(2), 2015.

[LFK+11] E. LeMay, M.D. Ford, K. Keefe, W.H. Sanders, and C. Muehrcke. Model-based security metrics using adversary view security evaluation (advise). In *Proceedings of the 8th International Conference on Quantitative Evaluation of Systems (QEST)*, pages 191–200. IEEE Computer Society, 2011.

[LFM+02] W. Lee, W. Fan, M. Miller, S. J. Stolfo, and E. Zadok. Toward cost-sensitive modeling for intrusion detection and response. *Journal of Computer Security*, 10(1–2):5–22, 2002.

[LH11] H. Li and Z. Han. Manipulating the electricity power market via jamming the price signaling in smart grid. In *IEEE Globecom Workshop on Smart Grid Communications*, pages 1168–1172, 2011.

[LHFB13] A. Laszka, G. Horvath, M. Felegyhazi, and L. Buttyán. Flipthem: Modeling targeted attacks with FlipIt for multiple resources. Technical report, budapest university of technology and economics, 2013.

[LIS+06] R. Lippmann, K. Ingols, C. Scott, K. Piwowarski, K. Kratkiewicz, M. Artz, and R. Cunningham. Validating and restoring defense in depth using attack graphs. In *IEEE Military Communications Conference (MILCOM)*, 2006.

[LKK07] J.C. Laprie, K. Kanoun, and M. Kaaniche. Modeling interdependencies between the electricity and information infrastructures. In *Proceedings of the 26th International Conference on Computer Safety, Reliability and Security (SAFECOMP)*, pages 54–67, 2007.

[LLC+03] C. Laughman, K. Lee, R. Cox, S. Shaw, S. Leeb, L. Norford, and P. Armstrong. Power signature analysis. *IEEE Power and Energy Magazine*, 1(2):56–63, 2003.

[LLL10] F. Li, B. Luo, and P. Liu. Secure information aggregation for smart grids using homomorphic encryption. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 327–332, 2010.

[LM05] Y. Liu and H. Man. Network vulnerability assessment using bayesian networks. In *Proceedings of the SPIE Conference on Data Mining, Intrusion Detection, Information Assurance and Data Networks Security*, pages 61–71, 2005.

[LNR11] Y. Liu, P. Ning, and M.K. Reiter. False data injection attacks against state estimation in electric power grids. *ACM Transactions on Information and System Security*, 14(1), 2011.

[LPS13] X. Li, H.V. Poor, and A. Scaglione. Blind topology identification for power systems. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2013.

[LVS+12] H. Lin, S. S. Veda, S. K. Shukla, L. Mili, and J. S. Thorp. GECO: Global Event-Driven Co-Simulation framework for interconnected power system and communication network. *IEEE Transactions on Smart Grid*, 3(3):1444–1456, 2012.

[LYY+12] Jie Lin, Wei Yu, Xinyu Yang, Guobin Xu, and Wei Zhao. On false data injection attacks against distributed energy routing in smart grid. In *IEEE/ACM Third International Conference on Cyber-Physical Systems (ICCPS)*, pages 183–192, 2012.

[Mat] Security Matters. URL: http://www.secmatters.com/.

[MBFB06] M. McQueen, W. Boyer, M. Flynn, and G. Beitel. Quantitative cyber risk reduction estimation methodology for a small SCADA control system. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, 2006.

[MBH07] A. McIntyre, B. Becker, and R. Halbgewachs. Security metrics for process control systems. Sandia Report SAND2007-2070P, Sandia National Laboratories, 2007.

[MBZ+06] V. Mehta, C. Bartzis, H. Zhu, E. Clarke, and J. Wing. Ranking attack graphs. In *Proceedings of the 9th International Symposium on Recent Advances in Intrusion Detection (RAID)*, pages 127–144, 2006.

[MC13] R. Mitchell and I. Chen. Effect of intrusion detection and response on reliability of cyber physical systems. *IEEE Transactions on Reliability*, 62(1):199–210, 2013.

[Mes07] J. Meserve. Sources: Staged cyber attack reveals vulnerability in power grid, September 2007. URL: http://articles.cnn.com/2007-09-26/us/power.at.risk_1_generator-cyber-attack-electric-infrastructure?_s=PM:US.

[MG11] P. Mell and T. Grance. The NIST definition of cloud computing (draft). *NIST special publication*, 800(145):7, 2011.

[MK11] F. A. Mohamed and H. N. Koivo. Multiobjective optimization using modified game theory for online management of microgrid. *European Transactions on Electrical Power*, 21(1):839–854, 2011.

[MKYBM08] R. A. Miura-Ko, B. Yolken, N. Bambos, and J. Mitchell. Security investment games of interdependent organizations. In *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing*, pages 252–260, 2008.

[MKYMB08] R. A. Miura-Ko, B. Yolken, J. Mitchell, and N. Bambos. Security decision-making among interdependent organizations. In *Proceedings of the IEEE 21st Computer Security Foundations Symposium (CSF)*, pages 66–80, 2008.

[ML10] C. Mu and Y. Li. An intrusion response decision-making model based on hierarchical task network planning. *Expert Systems with Applications*, 37(3):2465–2472, 2010.

[MM01] C. Michel and L. Mé. Adele: An attack description language for knowledge-based intrusion detection. In *Proceedings of the 16th International Conference on Information Security (IFIP/SEC)*, pages 353–368, 2001.

[MMA11] S. McLaughlin, P. McDaniel, and W. Aiello. Protecting consumer privacy from electric load monitoring. In *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS)*, pages 87–98, 2011.

[Mob] Mobius. URL: https://www.mobius.illinois.edu/.

[MPPW06] M. Majdalawieh, F. Parisi-Presicce, and D. Wijesekera. DNPSec: Distributed network protocol version 3 (DNP3) security framework. In *Advances in Computer, Information, and Systems Sciences, and Engineering*, pages 227–234. 2006.

[MRT15] E. Miehling, M. Rasouli, and D. Teneketzis. Optimal defense policies for partially observable spreading processes on bayesian attack graphs. In *Proceedings of the Second ACM Workshop on Moving Target Defense*, pages 67–76, 2015.

[MRWJ+10] A.-H. Mohsenian-Rad, V.W.S. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia. Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. *IEEE Transactions on Smart Grid*, 1(3):320–331, 2010.

[MW04] P. Manadhata and J.M. Wing. Measuring a system's attack surface. Technical Report CMU-CS-04-102, 2004.

[MWB11] D.K. Mulligan, L. Wang, and A.J. Burstein. Privacy in the smart grid: an information flow analysis. Report for the privacy issues in the smart grid project, 2011.

[MZA+10] M. Manshaei, Q. Zhu, T. Alpcan, T. Başar, , and J.P. Hubaux. Game theory meets network security and privacy. Technical Report EPFL-REPORT-151965, École Polytechnique Fédérale de Lausanne (EPFL), 2010.

[NAB09] K.C. Nguyen, T. Alpcan, and T. Başar. Stochastic games for security in networks with interdependent nodes. In *International Conference on Game Theory for Networks (GameNets)*, pages 697–703, 2009.

[NAC14] E. Nekouei, T. Alpcan, and D. Chattopadhyay. A game-theoretic analysis of demand response in electricity markets. In *IEEE PES General Meeting*, pages 1–5, 2014.

[NAC15] E. Nekouei, T. Alpcan, and D. Chattopadhyay. Game-theoretic frameworks for demand response in electricity markets. *IEEE Transactions on Smart Grid*, 6(2):748–758, 2015.

[Nat10] National Institute of Standards and Technology. NIST framework and roadmap for smart grid interoperability standards, Release 1.0, January 2010.

[Nat14a] National Institute of Standards and Technology. Guide to Industrial Control Systems (ICS) security, May 2014.

[Nat14b] National Institute of Standards and Technology. Guidelines for smart grid cybersecurity, Volume 1 - Smart grid cybersecurity strategy, architecture, and high-level requirements. NISTIR 7628 Revision 1, September 2014.

[Nat14c] National Institute of Standards and Technology. NIST framework for improving critical infrastructure cybersecurity, February 2014.

[NJ08] S. Noel and S. Jajodia. Optimal ids sensor placement and alert prioritization using attack graphs. *Journal of Network and Systems Management*, 16(3):259–275, 2008.

[NJWS10] S. Noel, S. Jajodia, L. Wang, and A. Singhal. Measuring security risk of networks using attack. *International Journal of Next-Generation Computing*, 1(1):135–147, 2010.

[NK12a] R. Nix and M. Kantarcioglu. Contractual agreement design for enforcing honesty in cloud outsourcing. In *Proceedings of the Third International Conference on Decision and Game Theory for Security (GameSec)*, pages 296–308. 2012.

[NK12b] R. Nix and M. Kantarcioglu. Efficient query verification on outsourced data: A game-theoretic approach. *arXiv preprint arXiv:1202.1567*, 2012.

[NWD+12] A. Nicholson, S. Webber, S. Dyer, T. Patel, and H. Janicke. SCADA security in the light of cyber-warfare. *Computers & Security*, 31(4):418–436, 2012.

[OBM06] X. Ou, W. F. Boyer, and M. A. McQueen. A scalable approach to attack graph generation. In *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS)*, pages 336–345, 2006.

[ODK99] R. Ortalo, Y. Deswarte, and M. Kaaniche. Experimenting with quantitative evaluation tools for monitoring operational security. *IEEE Transactions on Software Engineering*, 25(5):633–650, 1999.

[OGA05] X. Ou, S. Govindavajhala, and A. W. Appel. MulVAL: A logic-based network security analyzer. In *Proceedings of the 14th Conference on USENIX Security Symposium*, 2005.

[OMA+08] H. Otrok, M. Mehrandish, C. Assi, M. Debbabi, and P. Bhattacharya. Game theoretic models for detecting network intrusions. *Computer Communications*, 31(10):1934–1944, 2008.

[OR94] M. J. Osborne and A. Rubinstein. *A course in game theory*. MIT Press, 1994.

[Ort98] Rodolphe Ortalo. *Quantitative Evaluation of Information Systems Security*. PhD thesis, Report LAAS n° 98164, Institut National Polytechnique de Toulouse, 1998.

[PC12] V. Pham and C. Cid. Are we compromised? Modelling security assessment games. In *Proceedings of the Third Conference on Decision and Game Theory for Security (GameSec)*, pages 234–247. 2012.

[PCB10a] L. Pièvre-Cambacédès and M. Bouissou. Attack and defense modeling with BDMP. In *Proceedings of the 5th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security*, pages 86–101, 2010.

[PCB10b] L. Pièvre-Cambacédès and M. Bouissou. Beyond attack trees: Dynamic security modeling with Boolean Logic Driven Markov Processes (BDMP). In *Proceedings of the 8th European Dependable Computing Conference (EDCC)*, pages 199–208, 2010.

[PDR12] N. Poolsappasit, R. Dewri, and I. Ray. Dynamic security risk management using bayesian attack graphs. *IEEE Transactions on Dependable and Secure Computing*, 9(1):61–74, 2012.

[PFP] PFP Cybersecurity. URL: http://www.pfpcyber.com.

[PLM+11] R.A. Popa, J.R. Lorch, D. Molnar, H.J. Wang, and L. Zhuang. Enabling security in cloud storage SLAs with CloudProof. In *Proceedings of the 2011 USENIX Conference on USENIX Annual Technical Conference (USENIXATC)*, 2011.

[PM13] M. Parandehgheibi and E. Modiano. Robustness of interdependent networks: The case of communication networks and the power grid. In *IEEE Global Communications Conference (GLOBECOM)*, 2013.

[Pou03] K. Poulsen. Slammer worm crashed Ohio nuke plant network. *SecurityFocus*, August 2003. URL: http://www.securityfocus.com/news/6767.

[PP11] B. Paramasivan and K. M. Pitchai. Comprehensive survey on game theory based intrusion detection system for mobile adhoc networks. *IJCA Special Issue on Network Security and Cryptography*, NSC(5):23–29, 2011.

[PRO] PROMAPS. http://www.goodtech.se/en/services/power/promaps.

[PS98] C. Phillips and L. Swiler. A graph-based system for network-vulnerability analysis. In *Proceedings of the 1998 Workshop on New Security Paradigms*, pages 71–79, 1998.

[PSS+10] U.K. Premaratne, J. Samarabandu, T.S. Sidhu, R. Beresh, and J.-C. Tan. An intrusion detection system for IEC61850 automated substations. *IEEE Transactions on Power Delivery*, 25(4):2376–2383, 2010.

[PTC11] R. Pfitzner, K. Turitsyn, and M. Chertkov. Statistical classification of cascading failures in power grids. In *IEEE Power and Energy Society General Meeting*, pages 1–8, 2011.

[QWT+11] H. Qi, X. Wang, L. M. Tolbert, F. Li, F.Z. Peng, P. Ning, and M. Amin. A resilient real-time system design for a secure and reconfigurable power grid. *IEEE Transactions on Smart Grid*, 2(4):770–781, 2011.

[RA00] R. W. Ritchey and P. Ammann. Using model checking to analyze network vulnerabilities. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 156–165, 2000.

[RB06] H. A. Rahman and K. Besnosov. Identification of sources of failures and their propagation in critical infrastructures from 12 years of public failure reports. In *Third International Conference on Critical Infrastructures (CRIS)*, 2006.

[RD11] A. Rial and G. Danezis. Privacy-preserving smart metering. In *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, 2011.

[Reu12] Reuters. Aramco says cyberattack was aimed at production, December 2012. URL: http://www.nytimes.com/2012/12/10/business/global/saudi-aramco-says-hackers-took-aim-at-its-production.html?ref=technology&_r=0.

[RGS14] D.A. Rusu, B. Genge, and C. Siaterlis. SPEAR: A systematic approach for connection pattern-based anomaly detection in SCADA systems. *Procedia Technology*, 12:168–173, 2014.

[RIT+08] V. Rosato, L. Issacharoff, F. Tiriticco, S. Meloni, S. De Porcellinis, and R. Setola. Modelling interdependent infrastructures using interacting dynamical models. *International Journal of Critical Infrastructures*, 4(1/2):63–79, 2008.

[RKT10a] A. Roy, D. S. Kim, and K. S. Trivedi. ACT: Attack Countermeasure Trees for information assurance analysis. In *Proceedings of INFOCOM IEEE Conference on Computer Communications Workshops , 2010*, pages 1–2, 2010.

[RKT10b] A. Roy, D. S. Kim, and K. S. Trivedi. Cyber security analysis using attack countermeasure trees. In *Proceedings of the 6th Annual Workshop on Cyber Security and Information Intelligence Research*, pages 1–4, 2010.

[RKT12] A. Roy, D. S. Kim, and K. S. Trivedi. Attack Countermeasure Trees (ACT): Towards unifying the constructs of attack and defense trees. *Security and Communication Networks*, 5(8):929–943, 2012.

[RMR12] Md. A. Rahman and H. Mohsenian-Rad. False data injection attacks with incomplete information against smart power grids. In *IEEE Global Communications Conference (GLOBECOM)*, 2012.

[Ros65] J. B. Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica*, 33(3):520–534, 1965.

[RS98] C. R. Ramakrishnan and R. Sekar. Model-based vulnerability analysis of computer systems. In *Proceedings of the Second International Workshop on Verification, Model Checking, and Abstract Interpretation*, 1998.

[Rub98] A. Rubinstein. *Modeling bounded rationality.* MIT Press, 1998.

[RVC13] C. Rottondi, G. Verticale, and A. Capone. Privacy-preserving smart metering with multiple data consumers. *Computer Networks*, 57(7):1699–1713, 2013.

[RVK13] C. Rottondi, G. Verticale, and C. Krauss. Distributed privacy-preserving aggregation of metering data in smart grids. *IEEE Journal on Selected Areas in Communications*, 31(7):1342–1354, 2013.

[Sch99] B. Schneier. Attack trees: Modeling security threats. Dr. Dobb's journal, 1999.

[Sch00] B. Schwartz. Self-determination: The tyranny of freedom. *American Psychologist*, 55(1):79–88, 2000.

[Sch04] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer-Verlag, 2004.

[SCH13] J. L. Sánchez, R. Caire, and N. Hadjsaid. ICT and power distribution modeling using complex networks. In *IEEE Powertech Conference*, 2013.

[SEH13] 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*, 7(3):363–373, 2013.

[SEJ09] T. Sommestad, M. Ekstedt, and P. Johnson. Cyber security risks assessment with bayesian defense graphs and architectural models. In *Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS)*, pages 1–10, 2009.

[SEN09] T. Sommestad, M. Ekstedt, and L. Nordstrom. Modeling security of power communication systems using defense graphs and influence diagrams. *IEEE Transactions on Power Delivery*, 24(4):1801–1808, 2009.

[SGL12] S. Sridhar, M. Govindarasu, and C.-C. Liu. Risk analysis of coordinated cyber attacks on power grid. In *Control and Optimization Methods for Electric Smart Grids*, volume 3 of *Power Electronics and Power Systems*, pages 275–294. 2012.

[SHJ+02] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M. Wing. Automated generation and analysis of attack graphs. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 273–284, 2002.

[SHP11] W. Saad, Z. Han, and H. V. Poor. Coalitional game theory for cooperative micro-grid distribution networks. In *IEEE International Conference on Communications (ICC)*, pages 1–5, 2011.

[SHPB12] W. Saad, Z. Han, H. V. Poor, and T. Başar. Game theoretic methods for the smart grid. *IEEE Signal Processing Magazine*, 29(5):86–105, 2012.

[SKTO13] T. Spyridopoulos, G. Karanikas, T. Tryfonas, and G. Oikonomou. A game theoretic defence framework against dos/ddos cyber attacks. *Computers & Security*, 38:39–50, 2013.

[SO07] R. Sawilla and X. Ou. Googling attack graphs. Technical Report TM 2007-205, Defence R&D Canada, 2007.

[SPEC01] L.P. Swiler, C. Phillips, D. Ellis, and S. Chakerian. Computer-attack graph generation tool. In *Proceedings of the Second DARPA Information Survivability Conference & Exposition II (DISCEX)*, pages 307–321, 2001.

[SS03] S. E. Schechter and M. D. Smith. How much security is enough to stop a thief? In *Proceedings of the 7th International Financial Cryptography Conference*, pages 122–137, 2003.

[SSA91] SSADM-CRAMM subject guide for SSADM version 3 and CRAMM version 2. Technical Report Central Computer and Telecommunications Agency, IT Security and Privacy Group, 1991.

[SV14] M. Sola and G. M. Vitetta. Demand-side management in a smart micro-grid: A distributed approach based on bayesian game theory. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 656–661, 2014.

[SVP+12] S. Setty, V. Vu, N. Panpalia, B. Braun, A. J. Blumberg, and M. Walfish. Taking proof-based verified computation a few steps closer to practicality. In *Proceedings of the 21st USENIX Conference on Security Symposium*, 2012.

[SW08a] H. Shacham and B. Waters. Compact proofs of retrievability. In *Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT)*, pages 90–107, 2008.

[SW08b] N. Svendsen and S. Wolthusen. Modeling and detecting anomalies in SCADA systems. In *Critical Infrastructure Protection II*, volume 290 of *The International Federation for Information Processing*, pages 101–113. Springer US, 2008.

[SWB04] J. Salmeron, K. Wood, and R. Baldick. Analysis of electric grid security under terrorist threat. *IEEE Transactions on Power Systems*, 19(2):905–912, 2004.

[TK81] A. Tversky and D. Kahneman. The framing of decisions and the psychology of choice. *Science*, 211(4481):453–458, 1981.

[TL00] S. Templeton and K. Levitt. A requires/provides model for computer attacks. In *Proceedings of the Workshop on New Security Paradigms (NSPW)*, pages 31–38, 2000.

[TLM08] C-W. Ten, C-C. Liu, and G. Manimaran. Vulnerability assessment of cybersecurity for SCADA systems. *IEEE Transactions on Power Systems*, 23(4):1836–1846, 2008.

[Tof] Tofino. URL: http://www.tofinosecurity.com.

[Vac09] Geraldine Vache. *Evaluation Quantitative de la Sécurité Informatique-Approche par les Vulnérabilités*. PhD thesis, Institut National des Sciences Appliquées de Toulouse, 2009.

[VM08] J. Verba and M. Milvich. Idaho national laboratory supervisory control and data acquisition intrusion detection system (SCADA IDS). In *IEEE Conference on Technologies for Homeland Security*, pages 469–473, 2008.

[VVR+10] P. Vytelingum, T. D. Voice, S. D. Ramchurn, A. Rogers, and N. R. Jennings. Agent-based micro-storage management for the smart grid. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 39–46, 2010.

[Wen05] Wenyuan Li. *Risk Assessment Of Power Systems: Models, Methods, and Applications*. Wiley-IEEE Press, 2005.

[WIL+08] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia. An attack graph-based probabilistic security metric. In *Proceedings of The 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec)*, 2008.

[WJ10] D. Wei and K. Ji. Resilient industrial control system (RICS): Concepts, formulation, metrics, and insights. In *Third International Symposium on Resilient Control Systems (ISRCS)*, pages 15–22, 2010.

[WK09] W. W. Weaver and P. T. Krein. Game-theoretic control of small-scale power systems. *IEEE Transactions on Power Delivery*, 24(3):1560–1567, 2009.

[WL13] W. Wang and Z. Lu. Cyber security in the smart grid: Survey and challenges. *Computer Networks*, 57(5), 2013.

[WLJ06] L. Wang, A. Liu, and S. Jajodia. Using attack graph for correlating, hypothesizing, and predicting intrusion alerts. *Computer Communications*, 29(15):2917–2933, 2006.

[WNJ06] L. Wang, S. Noel, and S. Jajodia. Minimum-cost network hardening using attack graphs. *Computer Communications*, 29(18):3812–3824, 2006.

[WSJ07a] L. Wang, A. Singhal, and S. Jajodia. Measuring the overall security of network configurations using attack graphs. In *Proceedings of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, pages 98–112, 2007.

[WSJ07b] L. Wang, A. Singhal, and S. Jajodia. Toward measuring network security using attack graphs. In *Proceedings of the 2007 ACM Workshop on Quality of Protection (QoP)*, pages 49–54, 2007.

[WWRL10] C. Wang, Q. Wang, K. Ren, and W. Lou. Privacy-preserving public auditing for data storage security in cloud computing. In *Proceedings of IEEE INFOCOM*, pages 1–9, 2010.

[XLO+10] P. Xie, J. H. Li, X. Ou, P. Liu, and R. Levy. Using bayesian networks for cyber security analysis. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 211–220, 2010.

[XMS10] L. Xie, Y. Mo, and B. Sinopoli. False data injection attacks in electricity markets. In *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 226 –231, 2010.

[YJ13] S.L.P Yasakethu and J. Jiang. Intrusion detection via machine learning for SCADA system protection. In *First International Symposium for ICS & SCADA Cyber Security Research*, 2013.

[YML+13] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, B. Pranggono, and H.F. Wang. Intrusion detection system for IEC 60870-5-104 based SCADA networks. In *IEEE Power and Energy Society General Meeting (PES)*, pages 1–5, 2013.

[YUH06] D. Yang, A. Usynin, and J. W. Hines. Anomaly-based intrusion detection for SCADA systems. In *Proceedings of the 5th International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human Machine Interface Technologies*, 2006.

[YWW+11] J. Yang, H. Wang, J. Wang, C. Tan, and D. Yu. Provable data possession of resource-constrained mobile devices in cloud computing. *Journal of Networks*, 6(7):1033–1040, 2011.

[ZFBB09] Q. Zhu, C. Fung, R. Boutaba, and T. Başar. A game-theoretical approach to incentive design in collaborative intrusion detection networks. In *International Conference on Game Theory for Networks (GameNets)*, pages 384–392, 2009.

[ZHM90] M. Zviran, J. C. Hoge, and V. A. Micucci. SPAN-A DSS for security plan analysis. *Computers & Security*, 9(2):153–160, 1990.

[Zhu09] J. Zhu. *Optimization of Power System Operation*. Wiley-IEEE Press, 2009.

[ZHZB10] Q. Zhu, L. Husheng, H. Zhu, and T. Başar. A stochastic game model for jamming in multi-channel cognitive radio systems. In *IEEE International Conference on Communications (ICC)*, pages 1–6, 2010.

[ZKL05] C.V. Zhou, S. Karunasekera, and C. Leckie. A peer-to-peer collaborative intrusion detection system. In *IEEE International Conference on Networks*, 2005.

[ZKSY09] S.A. Zonouz, H. Khurana, W.H. Sanders, and T. M. Yardley. RRE: A game-theoretic intrusion response and recovery engine. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 439–448, 2009.

[ZL96] D. Zerkle and K. Levitt. NetKuang - A multi-host configuration vulnerability checker. In *Proceedings of the 6th USENIX Unix Security Symposium*, 1996.

[ZMPB10] X. Zheng, P. Martin, W. Powley, and K. Brohman. Applying bargaining game theory to web services negotiation. In *IEEE International Conference on Services Computing (SCC)*, pages 218–225, 2010.

[ZMST11] R.D. Zimmerman, C.E. Murillo-Sáchez, and R.J. Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, 2011.

[ZWH+10] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S. Yau. Efficient provable data possession for hybrid clouds. In *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS)*, pages 756–758, 2010.

[ZWS+11] Y. Zhang, Li. Wang, W. Sun, R. C. Green, and M. Alam. Distributed intrusion detection system in a multi-layer network architecture of smart grids. *IEEE Transactions on Smart Grid*, 2(4):796–808, 2011.

[ZYB12] D. Zheng, F. R. Yu, and A. Boukerche. Security and quality of service (QoS) co-design using game theory in cooperative wireless ad hoc networks. In *Proceedings of the Second ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet)*, pages 139–146, 2012.

Optimal Defense Strategies to Improve the Security and Resilience of Smart Grids

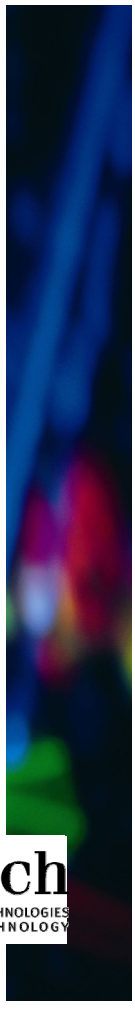
Ziad Ismail

RESUME : Du fait de l'évolution des menaces, la gestion des risques de sécurité dans le contexte d'un réseau électrique dit intelligent, ou smart grid, représente un défi. Cette thèse traite cette problématique en proposant des solutions basées sur la théorie des jeux non-coopératifs, les graphes d'attaques et les processus de décision markovien sous contraintes.
Dans la première partie de cette thèse, nous proposons et résolvons des modèles en théorie des jeux non-coopératifs pour optimiser le déploiement des ressources de défense dans le smart grid. Nous identifions le choix optimal des modes de sécurité sur les équipements d'une infrastructure relative aux compteurs intelligents, ou Advanced Metering Infrastructure (AMI), permettant de protéger la confidentialité des données clients. En outre, nous présentons un modèle analytique permettant d'identifier et de renforcer les équipements de communication les plus sensibles du réseau électrique.
Afin d'améliorer la sécurité des systèmes de contrôle industriel, la stratégie de défense a besoin d'être à la fois proactive, en anticipant les cibles potentielles des attaquants, et réactive en ajustant le type et l'intensité de la réponse en fonction du niveau de la menace. Dans la deuxième partie de la thèse, nous abordons ce défi et présentons une solution qui calcule la politique de sécurité optimale garantissant que les objectifs du défenseur sont satisfaits. Cette politique est obtenue par la résolution d'un processus de décision markovien sous contraintes construit à partir d'un graphe d'attaque généré préalablement et représentant l'évolution de l'état de l'attaquant dans le système.

MOTS-CLEFS : smart grid, théorie des jeux, politique de sécurité, optimisation.

ABSTRACT : The evolution of the threat landscape has made the security risk management in the smart grid a challenging task. This thesis addresses this problem and proposes solutions based on non-cooperative game theory, attack graphs and Constrained Markov Decision Processes (CMDPs).
In the first part of this thesis, using the framework of non-cooperative game theory, we define and solve models to optimize the deployment of defense resources in the smart grid. We find the optimal choice of security modes to enable on each equipment in the Advanced Metering Infrastructure (AMI) to protect the confidentiality of customers' data. In addition, we present an analytical model for identifying and hardening the most critical communication equipment used in the power system.
In order to improve the security of industrial control systems, the defense strategy needs to be both proactive by anticipating potential targets of adversaries, and reactive by adjusting the type and strength of the response to the threat level. In the second part of this thesis, we address this challenge by presenting a solution that computes the optimal security policy that guarantees that the defender's objectives are satisfied. This policy is obtained by solving a CMDP built using information in an attack graph generated beforehand that represents the evolution of the attacker's state in the system.

KEY-WORDS : smart grid, game theory, security policy, optimization.



TELECOM ParisTech

EDiTE

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY