Cyber Physical Security for Power Grid Protection

By

GEORGIA KOUTSANDRIA

Diploma Degree (Technical University of Crete, Chania, Greece) 2012

THESIS

Submitted in partial satisfaction of the requirements for the degree of

MASTER OF SCIENCE

in

Electrical and Computer Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Anna Scaglione, Chair

Sean Peisert

Soheil Ghiasi

Committee in Charge

2014

To the memory of my dear *παππούς*.

Contents

Georgia Koutsandria
July 2014
Electrical and Computer Engineering

Cyber Physical Security for Power Grid Protection

Abstract

Security of cyber-physical systems (CPSs), such as power systems, is becoming increasingly important due to the fact that these systems are accessible from the Internet, and therefore, are subject to a variety of threats that could have severe impact on their stability and performance. Network intrusion detection systems (NIDSs) are widely used to safeguard CPSs within a network perimeter from external threats by continuously monitoring and analyzing network traffic. This is a step up from the protection provided by traditional firewalls, since NIDS can sometimes go further than simple packet filters by performing more advanced types of deep packet inspection.

This thesis focuses on a novel use of NIDS that we call *hybrid control NIDS* (HC-NIDS). The HC-NIDS is tailored to detect attacks on networks that support hybrid controllers with power grid protection schemes. The security policies, derived form the hybrid automaton that designates the expected operation of the system, combine knowledge of desired communication rules as well as physical laws and limits that the system should obey in order to characterize network traffic as "safe" or "unsafe".

We apply HC-NIDS to two well-known protection mechanisms of power systems: the fault location, isolation and service restoration (FLISR) process in automated distribution systems (ADSs), and the power transformer's overcurrent protection scheme in transmission lines. For each case, we implement a set of specification-based intrusion detection rules based on the hybrid automaton that models the desired behavior of the system. Then, we create several attack scenarios to demonstrate our approach. As part of this evaluation, we developed an experimental framework that consists of a simulation of a physical system, and an emulation of the master controller, which implements the protection mechanism in the power grid, and allows communication via the Modbus TCP industrial control protocol.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor, Anna Scaglione, for giving me the opportunity to work with her, for her continuous support during my Master's degree studies, for her guidance, motivation, and enthusiasm. When I was applying for graduate schools, someone told me that the choice of advisor may be the most important decision I will make for my graduate studies. Almost three years after that moment, I can attest to the truth of his statement. When I came to Davis, I was a baby and Anna taught me how to walk. Her office's door was always open, and she was always more than happy to discuss, not only about research but even personal issues. Anna, has sharp insights on every research topic that she gets interested in, and always comes up with the most brilliant and sometimes crazy ideas that make you admire her more, and feed you with passion and energy to dive head first into a research problem. I cannot be more delighted that I chose Anna as my advisor.

I am very fortunate to have had the opportunity to collaborate with Professor Sean Peisert, who also serves on my committee. Sean, contributed to this thesis with genuine interest, coming up with great ideas, was always more than happy to provide me with honest advice, suggestions for improvements, and encouraging words whenever I needed them. I would like to thank my other thesis committee member, Professor Soheil Ghiasi, for his time, support, and patience.

Besides my advisor and my thesis committee, I would like to express my honest thanks to Dr. Chuck McParland, who was our project collaborator and contributed to the work described in this thesis, for his guidance and insightful comments. Also, my sincere thanks goes to my colleague Vishak Muthukumar, who was my research companion, helped and shared his ideas concerning the technical implementation part, and Masood Parvania for "filling" my knowledge gaps in the power related parts of my research during the last months.

I would like to extend my thanks to my lab mates. Mahnoosh Alizadeh, for all the helpful comments regarding my studies and my life in Davis, and for all the funny conversations that

made me blush. I promise you that I will not get married to a Duke so that you will be able to afford the appropriate cocktail gown. Lorenzo Ferrari, for reintroducing me to the fun side of the life, and reminding me that I can be happy, even in small places like Davis. Saeed Bagheri, for being the "silent strength" in our lab, but always giving me the most "to the point" comments regardless of the topic, and Xiao (Simon) Li for making me feel foolish with his research, but accepting to be my backup "future husband" in case I become forty and remain single. Thanks to the next generation of our lab and proofreaders of this thesis: Wai Hoi To and Reinhard Gentz.

Life is a long journey for which you need to have good friends beside you. I would like to deeply thank my "heart" friend Christina Christodoulakis for all the deep conversations and the laughter that make me so glad I have a friend like you, and remember: "Understand that friends come and go, but with a precious few you should hold on". But also I would like to thank my "little sis", Federica Barbagallo, for making my life in Davis colorful. I know our friendship is still young, but I hope that it will last even if I do not end up coming to the place that you love, Rome.

Last but not least, I would like to thank my family: my brother, Panos, who made me realize than even though life is not easy you should not quit and that every single day you have to fight in order to succeed, and my parents, Giannis and Thoula, for supporting me spiritually throughout my life and their unconditional love. I cannot find any appropriate words to express how grateful I am to all of you for every sacrifice that you have made on my behalf. Mom I love you!

This research was supported in part by the Director, Office of Computational and Technology Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy, under contract number DE-AC02-05CH11231. It is also supported in part by the Department of Energy under Award Number DE-OE0000097, and in part by the National Science Foundation under Grant Number CCF-1018871. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of any of the employers or sponsors of this work.

Chapter 1

Introduction

1.1 Overview and Motivation

The rapid development of the smart grid has a significant impact on today's cyber-physical systems (CPSs), including various elements of power systems. Among the many goals of the smart grid are improved reliability, efficiency, and sustainability of existing power systems. Moreover, a secure, reliable and economic power supply is closely related to a fast, efficient, and dependable communication infrastructure.

While originally CPS systems had minimal networking capabilities, attained through few serial ports per device, over the years they have incorporated Ethernet modems and packet switched communications that allow them to communicate with a large number of devices multiplexing the same communication medium. Numerous protocols are used by various vendors, including various open protocols, such as Modbus TCP/IP [1] and DNP3 [2], as well as various proprietary protocols.

However, by making software, hardware, and physical components accessible from the Internet, CPSs, including power systems communicating via Ethernet protocols, are exposed to traffic from a wide variety of sources. Several possible scenarios for attacks against network-connected cyber-physical systems are described by Zhu, et al. [3]. 'Several possible vectors

for attacks against cyber-physical systems include unauthorized access to the control software or access to the network traffic related to the control hardware that monitors and controls cyber-physical systems are both examples of physical attacks. Some such attacks could be manual, directly by human operation, and others could be automated via malware. For example, one of the most well-known attacks is the Stuxnet worm [4], first publicly discovered in June 2010. By using a combination of vulnerability knowledge, hacking pragmatism, and possible physical security breaches, Stuxnet was the first known malware to subvert industrial systems and rewrite programmable logic controller (PLC) code by including a PLC root kit. Another notable attack against cyber-physical systems is Havex RAT [5], started in spring of 2014 and conducted against industrial control systems to gather valuable information from SCADA systems.

The question arising is, therefore, what can be done to prevent similar hazards in the future? Instead of looking at the problem from a strictly computer-based perspective, when investigating security of CPSs, unlike with computer systems that do not control physical devices, there is additional information available: what can we learn from the physical operation of the device in question [6]? Cyber-physical systems have a particular set of functions that are continuously executed. The "hybrid" and dynamic behavior of cyber-physical systems can be represented by hybrid automata [7, 8] to account for both discrete and continuous aspects of the system behavior.

Power systems utilize protection mechanisms that follow specific hybrid automata models derived from the various designated operational limits. The ANSI/IEEE C37.2 standard [9] provides a taxonomy of these codified elements that are typically called interchangeable devices or functions. The various protection schemes are applied to power systems to ensure their safety, and decide about the state of the system, i.e., whether the system is under a safe or an unsafe operation mode. Therefore, both the network and PLC contexts and also the physical context of information exchanged in the control network can be used to check whether the system is consistent.

In this thesis, we argue that the hybrid automaton model, which designates the acceptable operation of the system, by including both communication and physical rules, can be used to identify traffic that deviates from the expected communication pattern or physical limitations, that could place the system in an unsafe mode of operation. We focus on protection schemes typically utilized to secure power systems, and we implement a set of intrusion detection rules derived from the hybrid automata. We call our use of a network intrusion detection system (NIDS) *a hybrid control NIDS (HC-NIDS)* as it incorporates the security policy, control environment rules, and physical models and constraints that come from the underlying hybrid control system, to define the safe and unsafe states of the cyber-physical system. The approach presented in this thesis, is presented to two of the main chapters of this thesis (Ch. 2, 3), where we focused on two different applications met in the power grid and are based on our accepted conference publications [10, 11].

In the next section, we briefly introduce the general architecture and the major components of CPSs. In Section 1.3, we discuss prior and current research on security of cyber-physical systems, and in Section 1.4 we briefly review common protection systems of the power grid that we focus the demonstration attention of this thesis. Then, in Section 1.5 we introduce our proposed approach on cyber-physical security of the power grid, by implementing intrusion detection rules based on the execution of the hybrid automata that specify the communication rules and physical limits that the system should obey.

1.2 Cyber-Physical Systems Overview

Cyber-physical systems (CPSs) conforming to the SCADA model were developed to reduce the "operational cost", and to allow system monitoring and remote control from a central location. These systems are generally used for large scale, geographically distributed infrastructures and are used in industries that require real-time monitoring and control of physical functions, such as the power grid. Modern power systems are designed to integrate an as-

sortment of machinery aiming to automatically monitor, control, and safeguard system's operation, referred to as "instrumentation and control (I&C) system".

1.2.1 Physical Equipment and Components

A typical architecture of a CPS systems consists of two main divisions that includes a SCADA master station and field devices, interacting with each others through communication networks. SCADA systems are generally used for large scale, geographically distributed infrastructures and are used in industries that require real-time monitoring and control of physical functions, such as the power grid. The SCADA model (Figure 1.2.1) is a hierarchical architecture where at the top of the hierarchy is a Supervisory Control Unit (SCU), interacting with several Remote Terminal Units (RTUs) connected directly to machinery or indirectly, communicating to field devices with embedded computation and communication capabilities, referred to as Intelligent Electronic Devices (IEDs).

The SCU is a rather traditional computer system. It includes a database that is called a "system historian" that provides analytics that feed the display on a Human Machine Interface (HMI). The HMI is the vehicle for the operators to send commands to change the settings of the machinery in the ICS. The RTUs and IEDs are, instead, the robotic arm of the system and are implemented via highly specialized micro-processors called Digital Relays (in power system applications) or, more commonly, Programmable Logic Controllers (PLCs). These are designed to program hybrid automata and vary widely in size and capabilities; typically the least capable are IEDs and the most capable are RTUs.

PLCs are capable of two types of communications: 1) PLC to machinery interactions, that occur through dedicated buses for digital signals typically to control switches (coils) and record their state or for analog signals inputs and outputs, that record sensor measurements and control continuous physical variables; 2) serial or packet switched communications between PLCs or between PLC and a computer that acts as SCU. The first type of interactions

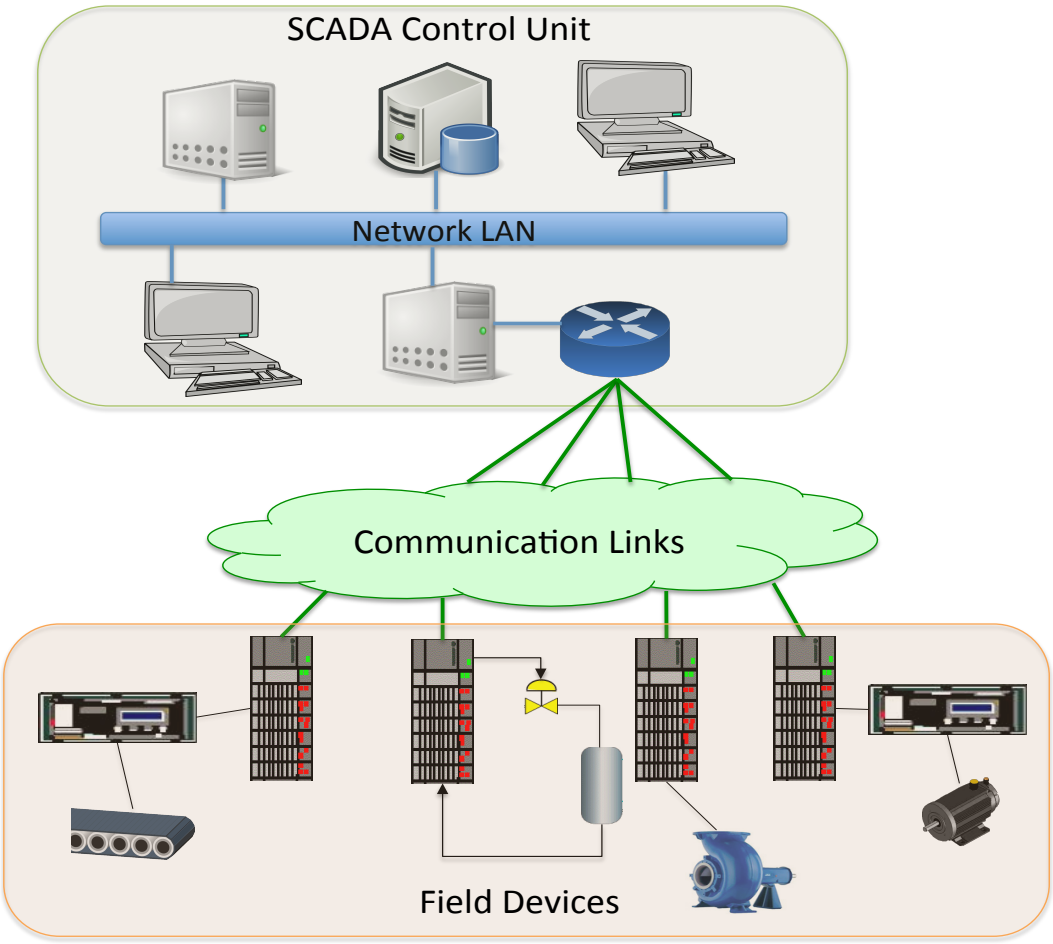

Figure 1.2.1: General architecture of Cyber-Physical Systems

has a one to one mapping with the state of the PLC memory, and this is the mechanism through which it is possible to program robotic applications. In fact, it is by reading and writing registers in the memory of the PLC that the control program is reading the state or altering the value of a physical coil or parameter in the corresponding physical subsystem.

1.2.2 Communication Networks

The communications networks serve as the intermediate instrument between a SCADA master station and the field devices. Various types of physical mediums can be used to conduct the communications within a power system, such as fiber cables, wireless communication, power line communication, direct copper cables, and land line telephone communication. In

order to provide control and automation to power systems under an effective manner, various industrial control protocols were adopted by the industry, including different versions of Modbus, DNP3, and IEC 61850 [12] that is specifically designed for substation automation systems. The goal of these protocols is to conform to the Open System Interconnection (OSI) model and IP standards so as to make it simple to establish connectivity among the devices utilized for networked automation.

In this thesis, we focus on the Modbus industrial control protocol due to practical convenience (inexpensive PLCs, a variety of software tools and a library). Modbus is an application layer protocol based on a request/response messaging model. More specifically, a client-server (or master-slave) transaction is established between devices that communicate over an Ethernet TCP/IP network. Once a TCP connection is established between two (or more) devices, the device that acts as a master can initiate a Modbus transaction by sending a message request to the slave(s). A master can directly communicate to one or more slaves by sending several queries and a slave responds to master queries until the connection is over.

A request consists of a function code that specifies to the slave what kind of action to perform such as read registers (analog quantities) or coils (digital quantities), the starting register/coil, the total number of registers/coils to be read, and any additional information that a slave needs to include in its respond defined by the function code action. On the other side, a response can either be normal or an exception if the slave sends a normal response, the message includes all the information that corresponds to the function code of the requested message. If an exception occurs, the response consists of a function code indicating that it is an exception, and a description of the error.

1.3 Prior Work on Security of Cyber-Physical Systems

Recent research on security for cyber-physical systems ranges from traditional computer security mechanisms, such as encryption, to more sophisticated intrusion detection techniques.

Network intrusion detection systems (NIDS) are common mechanisms used for real-time monitoring and analysis of network traffic. While there are numerous kinds of NIDSs, the most common type typically looks for contents of network packets known to be damaging in some fashion.

Numerous network intrusion detection approaches for cyber-physical systems have been studied, proposed, and built [13, 14]. *Anomaly-based* intrusion detection techniques are used to detect events that deviate from normal behavior by classifying network traffic as normal or abnormal using statistical models. Historically, anomaly detection has not been very effective in standard IP networks and general-purpose computing because of very high false positive rates, the lack of "malicious" training data, and the lack of actionability of the alerts given [15]. These challenges are partially due to the fact that traffic on many IT networks is very "noisy" and malicious activity often does not rise above the level of the noise of "normal" activity. It is also difficult to separate "abnormal but benign" activity from malicious activity.

Morris, et al. [16], introduced a number of *signature-based* intrusion detection rules for the Modbus protocol. While this approach is effective regarding the protocol's requirements and whether the network communication rules related to the Modbus protocol are satisfied, it focuses on a generic approach related to the specifications of a particular protocol rather than distinct devices.

Carcano, et al. [17], focused on attacks that appear legitimate when considered alone, but can harm the system when combined with other actions. However, while the approach focuses on the system's state by using the knowledge of the expected operation of a process, the scope of the process being considered does not extend to physical constraints and laws that should be satisfied to ensure stability of the system. The idea of using the power system physics, e.g., Ohm's and Kirchhoff's Laws, to operate, monitor and protect the grid, is at the heart of power system operation and reliability theories. System physics have previously been used for adequacy and security analysis [18], to filter bad measurement values and to reveal the state of the power grid, as exemplified by state estimation (SE) and energy

management systems (EMS) for the bulk power system [19]. A recent prolific line of work on cyber-physical security has also focused on Byzantine attacks in the SE functionality [20]. This work highlighted vulnerabilities of the bad-data detection step of SE in detecting well-constructed data injection attacks that provide *physically valid* measurements.

Chow, et al. [21], proposed a data-driven approach for cyber-physical security that includes three different cases of using the data of physical processes, i.e., apply anomaly detection to historical data. However, the proposed approach focuses on home area network (HAN) systems rather than the power grid, and handles the physical process data on a different way than the approach presented in this thesis does. Yoon, et al., [22], presented a security framework that combines known monitoring methods in order to identify malicious events through a statistical learning-based mechanism against real-time systems.

Cárdenas, et al. [23], focused on SCADA vulnerabilities and presented a theoretical approach to control systems' security, performing linear feedback control for linear state space equations. This work is close to the approach presented in this thesis with the difference that we focus on real automation applications typically met on power systems, that typically check the conservation of physical limits and laws that designate the stability. Lin, et al., [24], proposed to run contingency analyses to predict future consequences of control commands on a critical power asset in the context of transmission network applications. While effective, given the nature of transmission networks, it assumes information about other parts of the system are readily available, as they typically are in the transmission network, but not necessarily in distribution systems.

Application of intrusion detection in distribution systems has been primarily focused on detection of attacks on the advanced metering infrastructure (AMI) for monitoring purposes. Berthier and Sanders [25] developed a security monitoring system for smart meters. They deployed a specification-based intrusion detection sensor in the field to identify security threats in real time. The sensor monitors traffic between smart meters and the utility network and also ensures that devices are in secure state and prevent energy theft.

Other important work by Velente, et al, [26], does not focus on securing CPSs through intrusion detection techniques, but does highlight the significance of ensuring the trustworthiness of the various components that are included in a cyber-physical system, and are connected to the physical world. The authors proposed an approach that is focused on the well-known attestation technology, a method used to identify unauthorized changes to devices, in order detect deviations on the operation of devices that do not conform to their expected behavior. Wang, et al., [27], studied a specific CPS, i.e., central heating and cooling plant (CHCP), and the various components that make the system, where they confirmed through their conclusions the lack of testing models for CPSs.

In this thesis, we include our work in [10, 11] where we presented our HC-NIDS approach for the power grid, where we focused on automated power distribution systems, and on power transmission systems, respectively.

1.4 Power Grid: Protection Systems

In this section, we briefly introduce power system protection and its basic components that are widely used to protect power systems [28]. Power system protection is the procedure of ensuring that generation, transmission, and distribution of electrical energy is fulfilled safely against failures and abnormalities that could place the power system at risk. Therefore, the objective of power system protection is to provide accurate and fast fault clearing by isolating the faulty sections of the electrical power system from the rest of the "healthy" system in order to minimize damage due to failures or abnormalities, and maintain the stability of the power system.

In order to be able to detect the occurrence of faults in power systems, appropriate fault detection equipment is used to obtain information about the state of the system through observation of the quantities of current and/or voltage measurements. Then, based on the nature of the application under protection, different types of fault clearing equipment can be

used to automatically clear the electrical faults, including fuses, circuit breakers, instrument transformers and relays, aiming to protect various types of components and equipments such as generators, transformers, lines, buses, and capacitors.

1.4.1 Fuses and Circuit Breakers

The operation of a fuse and a circuit breaker aims on accomplishing the same objective: detect the occurrence of faults, and interrupt the current flow. Fuses are low resistance resistor that are made up of metal strip that directly melts when the element is overheated. On the other hand, circuit breaker are electrical switches that are energized (open contacts) once a fault is observed. In comparison to fuses that need to be replaced in case of faults, circuit breakers are reenergized (close contacts) once the fault has been cleared. However, fuses are considered to provide faster tripping actions than circuit breakers.

1.4.2 Instrument Transformers

The major types of instrument transformers are voltage transformers (VT), and current transformers (CT). In high voltage systems, such as transmission systems, it is necessary to scale-down the voltage from primary values to safe secondary values. Voltage transformers are designed to operate in such a way. Similarly, current transformers are utilized to reduce the current on the power system from high primary levels to secondary appropriate levels for supplying devices on the power system.

1.4.3 Protective Relays

Protective digital relays are typically used in power systems in order to perform automated control actions that protect physical equipment within the system, and ensure the stability of the system. These relays are designed to take various actions to detect the occurrence of faults on the grid and isolate the faulted section from the rest of the power system, by

continuously executing the same set of functions based on the system's physical limitations.

Power transmission systems consist of a collection of conductors aiming to transfer high voltage and current magnitudes of electric power from generation stations to customers through substations. Numerous primary and backup protection schemes protect power systems against possible damage by disconnecting the faulted area very soon after a fault is detected.

Several types of protective relays exist, which implement specific protection schemes in order to trip circuit breakers in the case of fault or abnormality. Overcurrent relays, such as instantaneous or time-delay overcurrent delays, are designed to operate in specific current regions. The overcurrent relays respond to magnitude of the current, and if this value exceeds a predefined current magnitude, known as pickup current, then circuit breakers are energized. The difference between the aforementioned relays is the timing of the tripping action, i.e., instantaneous or time-delay activation of the circuit breaker.

1.5 The Proposed Hybrid Control Network Intrusion Detection System Approach

Our work employs a combination of safety engineering principles and computer security practices to provide security to cyber-physical systems, including power systems. In this thesis, in particular, our goal is to identify and implement intrusion detection rules for protective digital relays in power systems based on the knowledge of the hybrid automata executed by the network of relays. Our novel use a NIDS integrates the computer and network security communication rules used by traditional NIDS approaches, with information related to the physical limits of the system, and the expected execution of its hybrid automata models, in order to mitigate an essential category of cyber-physical vulnerabilities.

The hybrid model that characterizes the process of the physical system, i.e., the transmission grid, is a combination of three things: 1) a set of system equations that constrain analog

quantities and depend on the state of the digital variables (hybrid state) or switches states, i.e, circuit breakers; 2) a controller program that routinely acquires sensor variables, compares them with physical conditions and determines the pattern of information exchanged by intelligent electronic devices (IEDs) to gather the sensor data and to issue physical commands or communication commands; 3) an application layer protocol that codifies how the message packets should be formatted and interpreted.

In our approach, we combine the hybrid model of the system and the communication aspects that make up the hybrid model of the system to define a "hybrid" set of NIDS rules of permissive device actions as signatures for the popular Bro Network Security Monitor [29], in order to provide a greater level of protection from cyber-physical vulnerabilities. In addition, we propose an experimental framework that we used to implement physical models in the Simulink simulation environment that interacts with embedded controllers to generate actual network traffic that the HC-NIDS continuously monitors. This allowed us to test our approach by emulating real cyber attacks. We validate our HC-NIDS approach and implementation for an overcurrent protection scheme for a power transformer. Any traffic that deviates from the expected normal operation of the protection scheme, as defined by Kirchoff's laws and implemented in our intrusion detection rules is characterized as a possible threat and triggers the HC-NIDS to raise an alert.

The remainder of this thesis is organized as follows: Chapter 2 discusses our approach of a hybrid control network intrusion detection system (HC-NIDS) for protecting automated distribution systems (ADSs) against certain scenarios of attacks over a computer network. In Chapter 3 we demonstrate our idea on protective digital relays in the power transmission grid. A description of the experimental framework that we built in order to provide realistic evaluation of cyber-physical systems, is presented in Chapter 4. Finally, we provide the conclusions of this thesis and discuss future work in Chapter 5.

Chapter 2

A Hybrid Control Network Intrusion Detection System for Automated Distribution Systems

The work presented in this chapter was first described in an earlier paper by Parvania, et al. [10]. This chapter described the implementation of our approach in an *automated distribution systems (ADS)*, and the evaluation of the idea presented in this thesis through a number of attack scenarios.

2.1 Automated Distribution Systems

Distribution automation refers to a blend of emerging technologies, such as switching technologies, sensor detectors, and communication protocols, that are utilized to control and monitor the operation of a power distribution system in an automated fashion [30]. The vision for an ADS is to facilitate the exchange of both electrical energy and information between system operators, customers, and other parties and equipment [31].

One of the promises of an ADS, is to allow the remote control and switching of the power distribution topology for protection and to improve reliability. In such an application, the

system operator would be able to automatically locate and isolate the faulted distribution component and restore the electrical service to the healthy parts of the distribution system. The process, called *fault location, isolation, and service restoration* (FLISR), is expected to considerably reduce the outage duration for customers [32].

Since ADS applications provide remote access to the critical distribution system components through communication networks, it is of paramount importance to coordinate their development with that of an appropriate cyber network framework that would prevent attackers from gaining control of circuit breakers and switches. Unfortunately, despite heightened attention to computer and network security issues [33, 34, 35], existing ADS structures were not designed with computer and network security in mind. Moreover, a basic challenge ADSs pose is that their various parts use different communication media and protocols, each with different security requirements.

2.2 Threats to and Countermeasures of Automated Distribution Systems

Some concerns were expressed over network security weakness and system fragility of power distribution systems [34, 35]. One of the reasons that exposes ADSs to attacks over computer networks is the fact that development of physical reliability methods has been divorced from a systematic assessment of the attacks that can be delivered over a computer network. In addition, the unconstrained integration of large numbers of communication systems that use both open and proprietary network protocols can expose ADSs to targeted cyber-attacks.

Several information technology-based security standards and systems, including firewalls, encryption schemes, authentication mechanisms, and network intrusion detection systems (NIDS), have been advocated and adopted in order to isolate control networks perimeters from external sources [14, 36, 37]. However, within a network perimeter, even encryption and authentication fail when an attack or simply an erroneous but damaging command is

mistakenly issued by an authorized user [38]. A variety of possible attacks against confidentiality, integrity, and availability of ADSs exist, with the most damaging ones being those in which controllers are made to perform actions that put the system in a physically unsafe state.

2.2.1 Denial-of-Service Attack

Many components of an ADS are sensitive to timing and require real-time communication to ensure secure and optimal operation of system. An attacker could flood a vital communication link with fabricated packets, causing key packets to be dropped, leading to abnormal operation of the system [39]. Therefore, it is important to identify such attacks on the power distribution system as their impacts could severely affect the reliability of the system.

2.2.2 Man-in-the-Middle and Eavesdropping Attacks

The distribution system spans large geographic areas and communication lines may well be physically unprotected in places. PLCs often communicate through unencrypted protocols that can be identified and analyzed by any open source network analysis tool by tapping into the cable in unguarded location. An attacker can modify the sensor values to the controller, potentially causing the controller to give control commands that send the system into an unsafe state [40]. Related to this, eavesdropping attacks may involve passive listeners of network traffic that reveal sensitive information about the status of physical devices, thus potentially enabling more damaging or more stealthy attacks against those devices. [41].

2.2.3 Insider Attack

A person that has some combination of authorized access of or access to a particular system, is commonly considered an *insider* [42]. Not all insiders are inherently malicious but given that they have knowledge and access of a system that others may not have, may have unusually

large ability to damage a system, either maliciously or accidentally. For example, insiders could compromise the system by planting a "logic bomb" or installing malicious software or hardware equipment on systems not easily accessible by others. As an example, an insider could substitute an important component of the system with a manipulated device that executes a source code in order to disable the network IDS and perform any attack attempts.

2.3 Scenario: Fault Location, Isolation and Service Restoration Process

Permanent failures, i.e., faults that are present until the faulty section or component is repaired, of any distribution system equipment, including cables and overhead lines, would cause a power outage for electricity customers. Traditionally, distribution system operators had limited monitoring and control access on distribution equipment, which made it a difficult and time-consuming process to manually locate a fault, dispatch a maintenance crew, and finally restore the service for customers.

Integration of remotely-controlled sectionalizing switches (SSs) and fault detectors (FDs) along with peer-to-peer communication between the protection devices enable the application of an automatic *fault location, isolation and service restoration process* (FLISR) in an ADS. The FLISR function automatically detects feeder faults, determines the fault location, isolates the faulted section of the feeder, and restores service to healthy portions of the feeder [43]. This automation of the process considerably reduces the customers' outage duration and improve the reliability of the distribution system [32]. The typical radial distribution feeder, shown in Figure 2.3.1, is an example of such a FLISR process. The feeder consists of four lines sections (L_i) which are equipped at both sides with SSs and FDs. The main feeder energizes the four load points (LP_i) through a circuit breaker (CB). Consider a permanent fault that occurs on line section L_3 in Figure 2.3.1. The FLISR process operates as follows:

1. Fault Location: The CB of the main feeder detects the fault, operates and de-energizes all the four downstream load points.

2. Fault Isolation: Fault detectors FD_4 and FD_5 report the location of the fault to the master station. Accordingly, an opening command is sent by the master station to sectionalizing switches SS_4 and SS_5 to isolate the faulted section.

3. Service Restoration: The master station sends a closing command to the feeder CB; therefore load points LP_1 and LP_2 are re-energized.

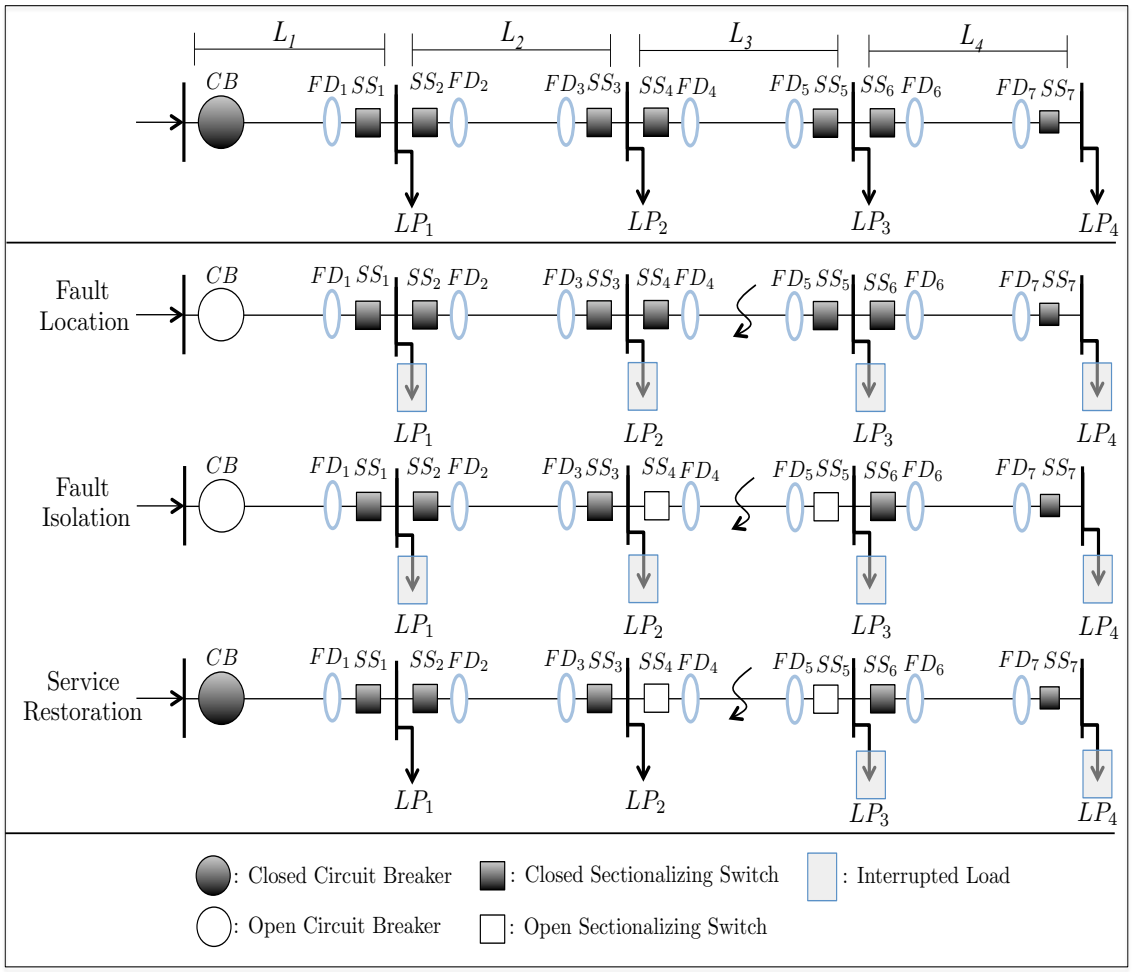

Figure 2.3.1: FLISR operation process

The operation algorithm of the FLISR system for the general case of a radial feeder with I line sections is presented in Algorithm 1.

Algorithm 1 General FLISR Algorithm

$SS_i = 0$, $FD_j = 0$ for $i, j = \{1, \cdots, 2I-1\}$
$L_k = 0$ for $k = \{1, \cdots, I\}$
if $CB = 1$ **then** ▷ Fault Occurs
  for $j = 1 : 2 : 2I-1$ **do**
    if $j > 1$ **then** ▷ Locate the Fault
      if ($FD_{j-1} = 1$ or $FD_j = 1$) **then**
        $k \leftarrow (j+1)/2$
        $L_k \leftarrow 1$, $SS_{j-1} \leftarrow 1$, $SS_j \leftarrow 1$ ▷ Isolate
      end if
    else
      if $FD_1 = 1$ **then** ▷ Locate the Fault
        $k \leftarrow 1$, $L_1 \leftarrow 1$, $SS_1 \leftarrow 1$ ▷ Isolate
        $count\_faults \leftarrow count\_faults + 1$
      end if
    end if
  end for
  $CB \leftarrow 0$ ▷ Close Circuit Breaker
  for $j = 1 : 2 : 2I-1$ **do**
    $k \leftarrow (j+1)/2$
    if $L_k = 0$ **then** ▷ Fault is repaired
      if $k = 1$ **then** ▷ Close the Sectionalizing Switches
        $SS_j \leftarrow 0$
      else
        $SS_{j-1} \leftarrow 0$, $SS_j \leftarrow 0$
      end if
    end if
  end for
end if

2.4 Hybrid Control Network Intrusion Detection Rules

Our "hybrid control" use of a NIDS is designed to perform real-time monitoring and analysis of network traffic and detect actions that do not conform to a set of predefined operational rules and policies [10]. This manner of intrusion detection is called "specification-based intrusion detection" [44]. We leverage the Bro Network Monitoring Framework [29] for a

variety of reasons but our technique could be implemented in other IDS frameworks as well. Bro is a well-established, open-source framework that includes IP packet parsers for two common industrial communication protocols, DNP3 and Modbus TCP. In our case, we assume that the Modbus TCP protocol is utilized as the communication protocol between the controllers in FLISR, although our approach applies equally well to other protocols such as DNP3, and indeed the Bro IDS that we use also contains a DNP3 parser, in addition to a Modbus parser.

We use Bro to monitor the network traffic of the FLISR system, and is responsible for identifying any actions that are not consistent with the physical operation and network communication rules of the Hybrid Control scheme that describes FLISR's normal operation and physical constraints. For this reason, as described earlier, we refer to approach as *Hybrid Control NIDS (HC-NIDS)*. The first layer of the HC-NIDS is an event engine that captures the network traffic, identifies each Modbus TCP packet, and forwards the packets to analyze to the second layer, which acts as the "rules layer". The captured Modbus packets are analyzed to check their consistency to the IDS rules, reflecting the FLISR process and operational physics. If a packet contains a command that would trigger a deviation from the prescribed process, a log entry or alert is triggered.

In this section, we present the set of intrusion detection rules that we implemented for the FLISR system. The intrusion detection rules are the result of the execution of the hybrid automaton that designated the expected behavior of the system. Our security policies consist of rules related to communication patterns, such as IP addresses, as well to the expected physical behavior of the system, e.g., packet sequence.

We want to highlight that the first category of security policies constitutes a well known and studied practice concerning the security of CPSs. However, our work focuses on the consideration of the physical laws in addition to the communication rules that the system should obey. For that reason we also demonstrate rules for several common security policies that check the whether the general communication rules are satisfied, e.g., use of acceptable

IP addresses, to show how rules looking at traditional network security rules and physics-based rules can run alongside each other in the same NIDS. Even though the intrusion detection rules that we implemented are applied to the network traffic exchanged within the network of the controllers, we use both the physical context of the information exchanged in the control network alongside the traditional IP network context.

2.4.1 Intrusion Detection Rule 1: IP Address

A set of of acceptable IP addresses, corresponding to control devices in our system, are specified in the NIDS. Any request packet that has an IP address different than the master controller's IP address or any response packet that has an IP address different than the slave controller's IP address indicates a packets that should be disallowed.

2.4.2 Intrusion Detection Rule 2: Function Code

Based on the normal operation of the FLISR process that we described earlier in this chapter, only "write" to single coils (function code (fc) = 5) commands are allowed. This NIDS rule treats packets that include any other function code as a potentially damaging one. Even though this intrusion detection rule focuses on a communication aspect of the system, the decision of the acceptable function codes is done based on the expected behavior of the system.

2.4.3 Intrusion Detection Rule 3: Packet Sequence

Figure 2.4.1 shows the communication sequence that we expect to observe in the network traffic related to the FLISR process. Our NIDS continuously checks the sequence of the packets exchanged within the network of the controllers and alerts on packets that deviate from the expected packet sequence.

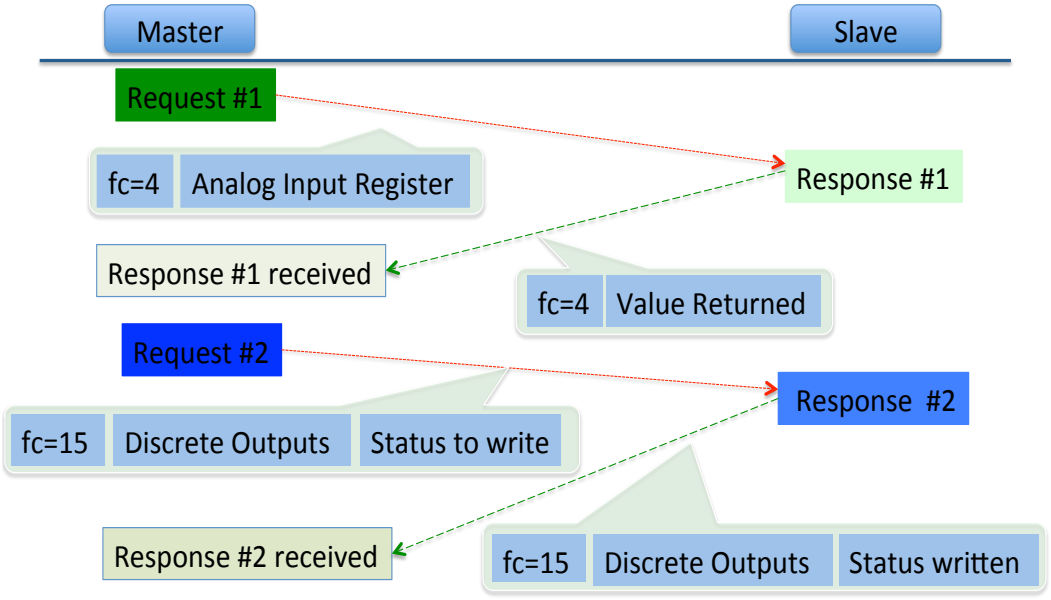

Figure 2.4.1: Communication packet sequence of FLISR

2.4.4 Intrusion Detection Rule 4: Cycle Duration

We define as one cycle the time gap between two "write" commands specified in the expected packet sequence, which is considered to have a relatively constant value. Significant deviations from the average cycle duration trigger an alert.

2.5 Experimental Validation

In order to demonstrate the utility of our approach in protecting the FLISR system, we implemented different attack scenarios whose main purpose is to either confound the system or retrieve important information about the system's state. Our primary goal in describing this assessment is to demonstrate that by leveraging knowledge of the system's expected behavior, our approach can observe a broad range of potential classes of intrusions, in addition to the typical intrusion detection rules that many existing NIDS employ. The attack scenarios introduced later in this section show cases where a traditional NIDS works well by checking that the communication rules are not violated, as well cases that demonstrate capabilities

largely specific to the approach used in the HC-NIDS.

The experimental set-up of our implementation, as shown in Figure 2.5.1, consists of physical devices, PLCs, and the HC-NIDS. We used two Siemens SIMATIC S7-1200 series PLCs [45], model CPU 1212C AC/DC/RLY, that are configured to emulate the FLISR system's tasks and communicate through the Modbus TCP protocol. The master controller emulates the FLISR master station and receives as input data the status of the FDs, that are implemented by digital switches. The slave controller in Figure 2.5.1 emulates the actions of circuit breaker and sectionalizing switches. In order to perform the FLISR functions, the slave controller receives queries from the master controller to enable or disable the circuit breaker and sectionalizing switches. The control algorithm of the FLISR is programmed on both controllers using the ladder logic programming language on the SIMATIC STEP 7 Basic software [46].

As introduced in previously, the Bro IDS [29] is used to implement the HC-NIDS rules, and is the core component of our implementation. We used Bro's scripting language to implement IDS rules in terms of policy scripts that define the allowed scope of the FLISR system. The last part of our experimental implementation is a set of scenarios that demonstrate the capability of the HC-NIDS for detecting several types of attacks. For this purpose, we programmed a libmodbus-based Modbus Master Simulator in C that acts as the attacker in the FLISR system. We assume that the attacker initiates a connection with the slave controller during the periods when there is not any packet exchange between the two controllers.

2.5.1 Denial-of-Service Attempt

The main purpose of the DoS attack is to make the slave controller unavailable to its intended master controller. The specific scenario constitutes an example of attacks typically met on cyber-physical systems, and there is research that already studied these type of attacks. However, for validation purposes we wanted to include a case that corresponds to the traditional security policy that we included in our HC-NIDS. Under the assumption mentioned above,

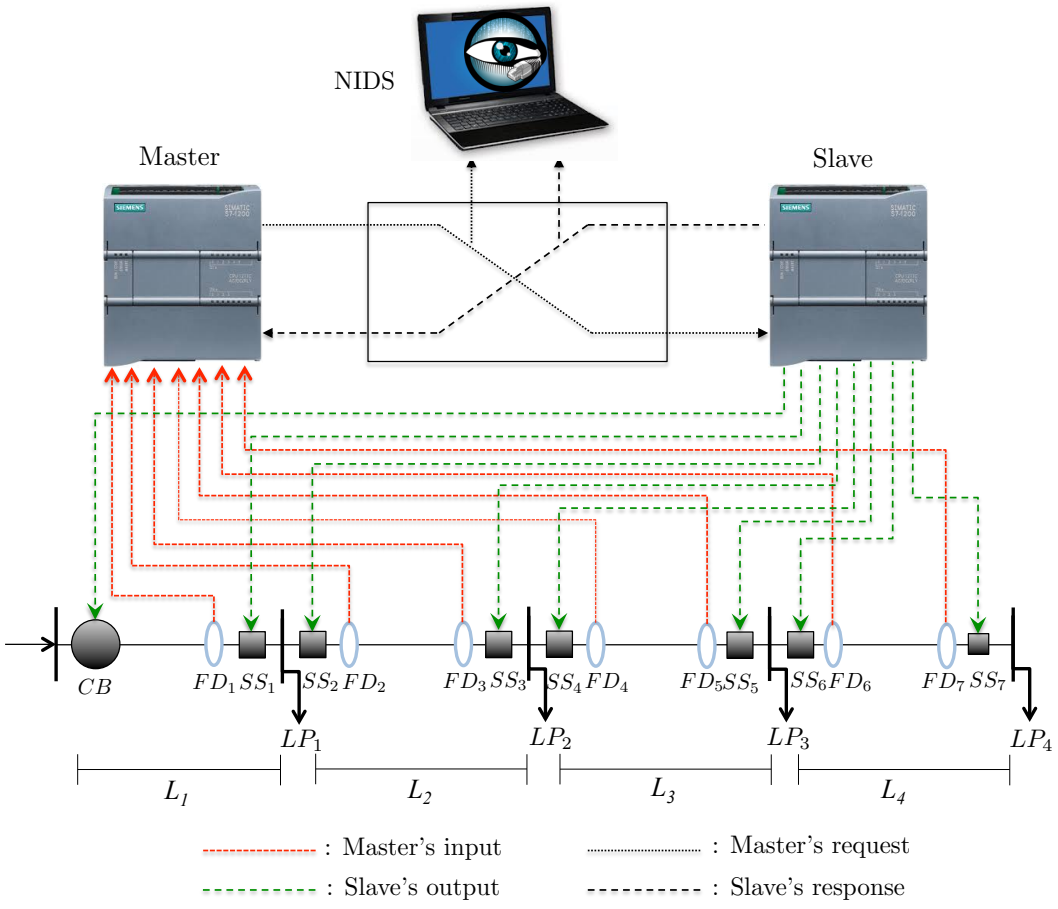

Figure 2.5.1: Configuration of the FLISR emulator

the attacker dispatches queries to the slave controller. The HC-NIDS analyzes the network traffic and checks the expected traffic rules, as described in previous section. In this case, the HC-NIDS determines if a non-acceptable IP address commits queries, and produces an alarm that notifies the network administrator about the suspicious attempt (Chapter 2.4.1). This experiment looks at the scenario presented in Chapter 2.2.1.

2.5.2 Data Memory Access

The objective of this attack scenario is to probe the status of physical devices to determine the state of the devices, and if it is possible to learn more about the devices such that further action can be taken, such as causing power outages. We assume the attacker obtains information about the master controller's IP address and dispatches a "read" command request. The

HC-NIDS determines that a "read" command function code is not in the list of acceptable function codes and generates an alarm indicating an attempted illicit action (Chapter 2.4.2).

2.5.3 De-Energizing the Distribution Feeder

In this attack, the attacker aims to de-energize the whole distribution feeder by opening the main feeder CB. We assume the attacker retrieves information of the FLISR network configuration, including the controllers' IP addresses, the memory allocation, and mapping to the physical devices, and the utilized command function codes. However, we assume that the attacker is not aware of the expected packet sequence. Based on the information obtained from the network traffic, the attacker sends a "write" command request to open the CB. In this case, the malevolent attempt passes the three intrusion detection rules, i.e., controllers' IP addresses, command function codes and cycle's time gap. However, the HC-NIDS detects that the initial packet is not followed by an opening command to the SSs, so it is not consistent with the packet sequence in Figure 2.4.1 (Chapter 2.4.3). More specifically, this attack scenario only includes the exchange of one "write" command query that targets the CB, and thus a faulty packet sequence is observed and the HC-NIDS generates an alert indicating that a malfeasance activity observed.

2.5.4 Causing Power Outage for Intended Load Points

In this attack, the attacker aims to cause a power outage for certain load points by isolating specific line sections. We assume that the attacker is aware of the controllers' IP addresses, the utilized command function codes, and the acceptable packet sequence. However, we assume that the attacker is not aware of the cycle's duration. The attacker generates an acceptable packet sequence with the master controller's IP address and the utilized command function codes. First, the attacker dispatches a "write" command request to activate the CB, and then sends a second "write" command request in order to activate specific SSs. The HC-NIDS observes the time gap between the newly issued queries and detects that the

packets' time gap is not consistent with the expected time gap between two packet requests (Chapter 3.3.4). This observation triggers the HC-NIDS to issue an alert that indicates the occurrence of a possible malevolent action. Figure 2.5.2 shows the timing difference between normal network traffic and several simulated attacks. The attack packets constitute larger time gaps which obviously makes them identifiable in comparison to the normal traffic.

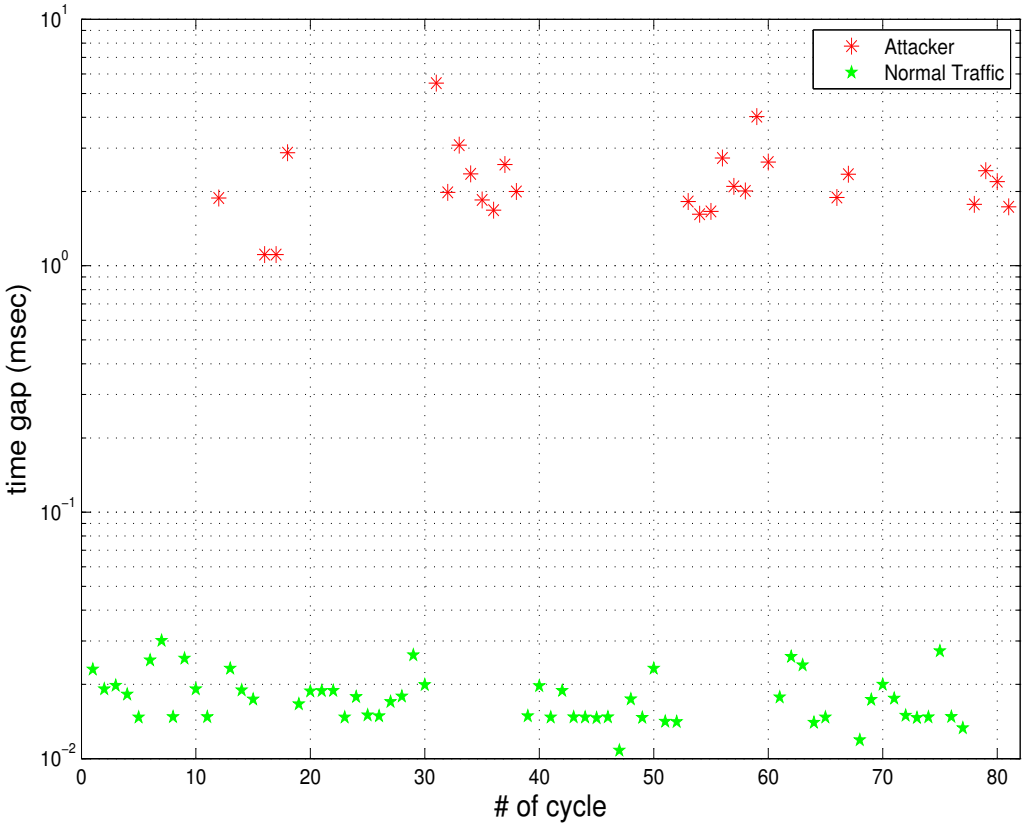

Figure 2.5.2: Timing difference between normal network traffic and attacks

2.5.5 An Insider Attack Scenario

This attack is similar to the attack that causes power outages for intended load points. However, we now assume that the attacker has a complete knowledge about the FLISR process and the network that includes the controllers' IP addresses, the command function codes, the acceptable packet sequence, and the time gaps between packets. Thus, the attacker

is able to dispatch queries that are completely consistent with the system's expected behavior.

We assume that just before the master controller initiates a new block of packets, the attacker takes charge of the connection and initiates a block of packets that follow the observed features of the expected network traffic. The attacker can launch targeted attacks by activating or de-activating different circuit breaker and sectionalizing switches.

Detecting this attack scenario requires more knowledge about the exact state of the distribution feeder. The FLISR process is enabled in cases where fault currents are detected. In our implementation, as described previously, we manually create faults using the digital switches attached to the PLC, and there is no information, related to the current that flows on the distribution line, included in the exchanged traffic. Therefore, since we are only monitoring one communication path, as we have currently implemented it, the HC-NIDS has incomplete information about the physical state, and insufficient knowledge to detect the attack. If we monitored more than one communication path, and two instances of Bro could communicate with each other, then physical values, such as current, would be included in the payload, and our HC-NIDS could be capable of detecting this attack scenario.

Chapter 3

A Hybrid Control Network Intrusion Detection System for Protective Digital Relays in the Power Transmission Grid

The work presented in this chapter was first described in an earlier paper by Koutsandria, et al. [11]. In this chapter, we focus on the implementation of intrusion detection policies for digital relays, which are used in the power transmission grid in order to implement protection mechanisms and serve as *proactive* digital relays. We chose the transformer's overcurrent protection scheme as a second example of protection schemes implemented on the power transmission grid and validate our approach through the implementation of several attack scenarios.

3.1 Threats and Countermeasures of the Power Transmission Grid

The communication infrastructure of modern power systems provides attackers with the ability to remotely issue false events that could damage the physical systems. Numerous

scenarios for traditional network attacks, as well as attacks that can cause actual physical damage are possible [3], by manipulating the hardware that monitors and controls the physical systems. When a fault occurs in the power transmission grid, protective digital relays isolate the faulted equipment by opening the adjacent circuit breakers. An attacker performing the Denial of Service (DoS) attack can paralyze the system by preventing the exchange of valuable information, which could lead to faulty decisions. In addition, knowledge of the transmission grid's configuration can be leveraged to create data injection attacks that bypass "bad data" detection algorithms. In fact, an attacker could fabricate meter measurements in a way that leads to false estimation of power systems' state, such that the current bad data detectors could not detect the attack [47].

The traffic analysis attack [48], constitutes another noteworthy type of attack in the power transmission grid. An attacker could exploit this vulnerability by monitoring the data acquisition packet response being sent by field devices to the control center. Information such as the sender and receiver's addresses, and details of the transmitted messages can be obtained even if the packets are encrypted. In power systems, critical information, including bus voltage magnitude, active and reactive power etc., are sent periodically by the distributed monitors to the control center. Therefore, an attacker could perform traffic analysis attacks in order to obtain crucial information about the power system, and proceed to further attacks aiming to compromise specific parts of the grid.

3.2 Power Transformer's Overcurrent Protection Scheme

We chose the transformer's overcurrent protection scheme as an example of protection schemes typically implemented on the power transmission grid. The example consists of a three-phase, two-winding transformer that connects a generating unit to a transmission line. Two three-phase circuit breakers are connected to both sides of the transformer. We focus on the instantaneous overcurrent relay which provides rapid clearing of severe internal faults and

external through-fault currents. The instantaneous overcurrent relay corresponds to device number 50 in the IEEE C37.2 standard [9], and is responsible for activating circuit breakers whenever the input current exceeds a predefined pickup current. The sufficient margin for the pickup current of the instantaneous overcurrent relay would be between 125%-175% of the maximum low-side three-phase symmetrical fault current [49].

3.3 Hybrid Control Network Intrusion Detection Rules

Our *Hybrid Control NIDS (HC-NIDS)* provides a "hybrid" set of intrusion detection rules by blending common network communication policies with physical constraints that designate the physical system's normal operation. In our approach, we implement these intrusion detection rules in terms of policy scripts applied to the exchanged network traffic within the physical system model. Similar to the cases presented in Chapter 2, to implement the intrusion detection rules, we leverage the Bro Network Security Monitor which includes IP packet parsers for two common industrial communication protocols, DNP3 and Modbus TCP. In our work, communication between the controllers uses Modbus TCP.

The HC-NIDS continuously monitors the network traffic of the physical system and executes the set of the intrusion detection rules that we implemented in order to identify events that deviate from the expected operation of the physical system. Every Modbus packet included in the network traffic is analyzed and characterized as acceptable or suspicious traffic by comparing specific fields of the packet to the HC-NIDS policies. The HC-NIDS triggers an alarm in the form of a log entry whenever a deviation is observed.

The following subsections describe the intrusion detection rules (IDSs) compose our HC-NIDS applied to the power transformer's overcurrent protection that are derived from the expected behavior of the system. Our set of intrusion detection rules includes security policies that focus on the physical aspects of the system, based on the approach presented in this thesis, in addition to common security policies, e.g., check of acceptable IP addresses.

Therefore, our contribution is in the way that we utilize the physical information included in the network traffic exchanged within the network of the controllers, that we take in mind in order to implement a full set of security policies that also includes common security checks. Figure 3.3.1 shows the block diagram of the desired behavior of the power transformer's system that checks both communication rules and whether the controlling actions based on the physical laws are consistent with the hybrid automaton of the system. In our implementation, we do not take under consideration exception responses that occurred due to communication errors that are not related to the operation of the physical system, ex., timeout responses, which is shown in dashed lines in Figure 3.3.1.

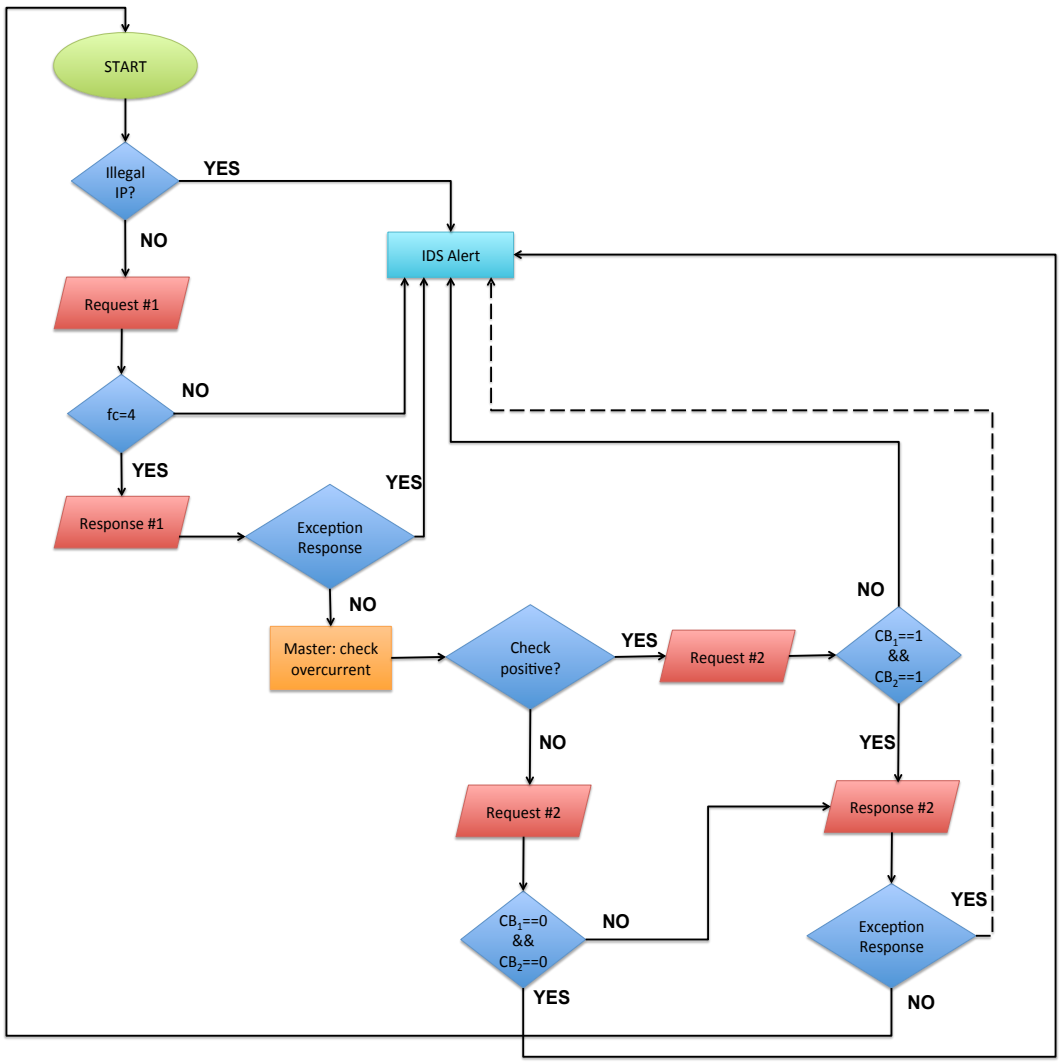

Figure 3.3.1: Block diagram of the expected behavior of the system

3.3.1 Intrusion Detection Rule 1: IP Address

The master and slave controllers' IP addresses are included in the list of acceptable IP addresses in our HC-NIDS. The predefined IP addresses are unique and any occurrence of unknown IP addresses signifies a possible threat.

3.3.2 Intrusion Detection Rule 2: Function Code

Only read input register ($fc = 4$), and write multiple coils ($fc = 15$) commands are accepted by our HC-NIDS. The first type of commands is used to obtain the value of the current that flows on the transmission line, whereas the second one indicates the status to be given to the circuit breakers, i.e. activate or not the circuit breakers.

3.3.3 Intrusion Detection Rule 3: Packet Sequence

The master controller continuously issues read requests in order to obtain the value of the current that flows on the transmission line, and write requests that set the circuit breakers aside to the transformer on a specific condition, which reflects the result of the power transformer's overcurrent protection scheme. The expected packet sequence of the transformer's overcurrent protection is shown in Figure 3.3.2. Since in the network traffic a response packet does not always appear after the associated query packet, we also use the transaction ID of the packets in order to check whether the appropriate pairs of packets are observed in the network traffic.

3.3.4 Intrusion Detection Rule 4: Time Gap

The master controller issues queries with a specific rate, and the average time gap between one read and one write request within a cycle should fall within a specific range. Any deviation of this range indicates an attempt of possible illicit action, such as injecting packets that could activate the circuit breakers when it is not expected.

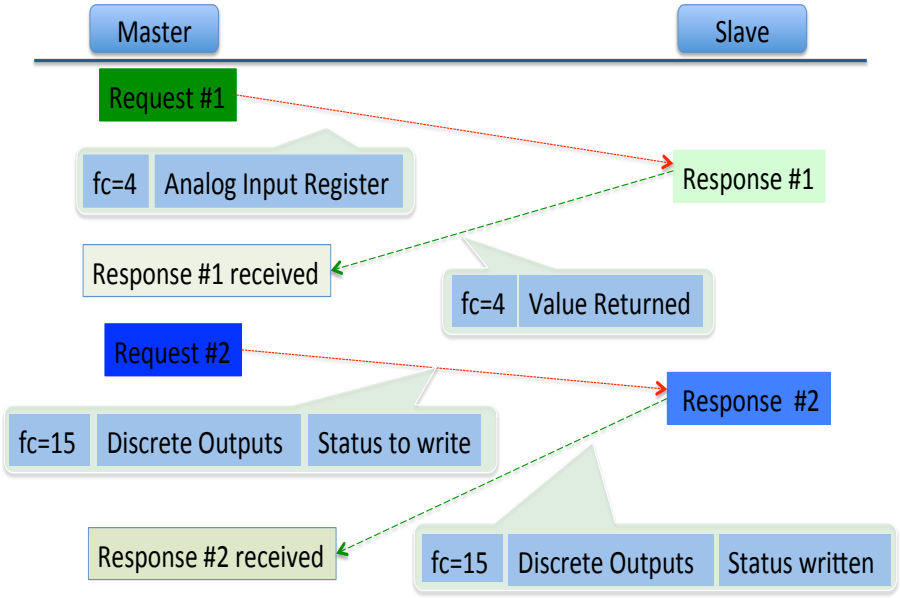

Figure 3.3.2: Expected communication packet sequence

3.3.5 Intrusion Detection Rule 5: Physical Constraints

The overcurrent protection scheme that we studied in this work is characterized by a pickup current that shows whether a fault happened and the transformer should be isolated by activating the circuit breakers. In our implementation, we assume that the pickup current of the instantaneous overcurrent relay is 125% of the maximum low-side three-phase symmetrical fault current.

3.4 Threat Model

Before describing our threat model, we first define several levels of knowledge that might characterize an attacker's abilities to deceive our HC-NIDS and confuse or even damage the physical system. In order to evaluate the performance of our HC-NIDS in protecting the power transformer's physical model, we examine several attack scenarios aimed at perturbing the normal operation of the system. Since our approach is focused on leveraging physical constraints and information that ensure the stability of the system, our evaluation focuses on attack scenarios that our HC-NIDS is able to identify through intrusion detection rules

related to physical operation of the system.

However, given that we are building our detection leveraging the Bro IDS, as with any general-purpose NIDS, Bro is capable of detecting attacks relevant to specified communication policies, including the IP addresses of the master and slave controllers. In this subsection, we present a description of the attack scenarios that we implemented. We provide additional details about their implementation along with results regarding the transformer's overcurrent protection in the subsequent section.

3.4.1 Attacker's Knowledge Level

In our attack scenarios, we consider various knowledge levels that might gauge an attacker's ability to penetrate into the system and bypass some or all of the intrusion detection rules. The diversification of the various levels is based on the information that an attacker could obtain by penetrating the network in order to confuse or damage the electrical physical system.

1. *Zero or Low:* Limited knowledge about the network communication rules, such as IP addresses of the controllers and the communication protocol used within the system.

2. *Moderate:* The attacker fingerprints the master and slave controllers, identifying information about IP addresses, and memory mappings to physical devices.

3. *Privileged:* The attacker obtains knowledge about the IP addresses used by the controllers, the command function codes used, and partial physical information, such as the expected packet sequence.

4. *Sophisticated:* This level constitutes the most challenging category to be identified and defended by a NIDS. In this case, the attacker gains complete access to the network traffic and also has access to sophisticated tools that assist in analyzing the network traffic, and extracting integrated information about the physical process.

3.4.2 Attack Scenario 1: Memory Access

An attacker performing this attack seeks to acquire information about the status of the circuit breakers adjacent to the power transformer. The specific attack scenario does not have any physical impact to the system, however when the attack is performed successfully, an attacker could then proceed to further actions and then cause actual physical harm the system, such as by activating the circuit breakers. We assume that the attacker only has information about the network and physical configuration of the system corresponding to the "moderate" level of knowledge. However, we assume that the attacker is not aware of the expected/normal sequence of function codes in the network traffic. Our HC-NIDS detects the specific type of attacks via the IDS Rule 2 (Ch. 3.3.2).

3.4.3 Attack Scenario 2: Injecting Malfeasant Packets

The main purpose of this attack is to disable the power transformer by activating the circuit breakers. The attacker obtains a "moderate" level of awareness, which does not include information about the average time gap or the expected packet sequence of commands. Therefore, our HC-NIDS is capable of detecting this type of attack by checking the consistency of the expected packet sequence. This is addressed by IDS Rule 3 (Ch. 3.3.3).

3.4.4 Attack Scenario 3: Imitating the Master Controller

This attack scenario is similar to the attack that intend to isolate the power transformer when it is not required by the overcurrent protection scheme. The attacker obtains a "privileged" level of awareness, which includes information about the expected packet sequence of commands. However, we assume, as it is inferred by the definition of the knowledge level, that the attacker is not aware of the overall system's expected operation and does not know about the physical constraints applied to the overcurrent protection scheme. This type of attack is directly related to our "hybrid" set of intrusion detection rules included in our

HC-NIDS, which is identified through IDS Rule 5 (Ch. 3.3.5).

3.5 Evaluation of Attack Scenarios

Our experimental results and observations demonstrate the capabilities of our HC-NIDS for detecting a wide range of attacks by using the physical constraints and the overall expected behavior of the studied physical system in addition to the common communication rules that are included in our HC-NIDS.

3.5.1 Attack Scenario 1

The attacker issues "read" command requests in order to obtain information about the status of the circuit breakers. However, the circuit breakers are digital outputs and the function code that corresponds to the specific type of command is not included in the acceptable function codes defined in our HC-NIDS. Therefore, the IDS Rule 2 included in our HC-NIDS detects the specific type of attack and our HC-NIDS generates an alarm indicating the illicit action.

3.5.2 Attack Scenario 2

We assume that the attacker attempts to inject false data in the form of queries aiming to isolate the power transformer on the transmission line. The packet sequence of the issued commands that appear in the network traffic do not conform to the expected packet sequence that our intrusion detection rules in the HC-NIDS specify. Figure 3.5.1 shows an instance of the network traffic of the exchanged communication packets between the slave controller and the polling devices, i.e., master controller and attacker, in the form of Modbus TCP/IP queries and responses. We expect to observe a repetitive pattern of packets in the network traffic, which consists of two pairs of packets, i.e. read query-response (green bars) and write query-response (blue bars). In Figure 3.5.1, red bars indicate the identified illicit actions,

in the form of injected packets, in between the packets that correspond to normal network traffic. As a result, our HC-NIDS identifies the attempt to activate the circuit breakers beside the transformer on the transmission line, and issues alerts indicating an unacceptable packet sequence of issued commands.

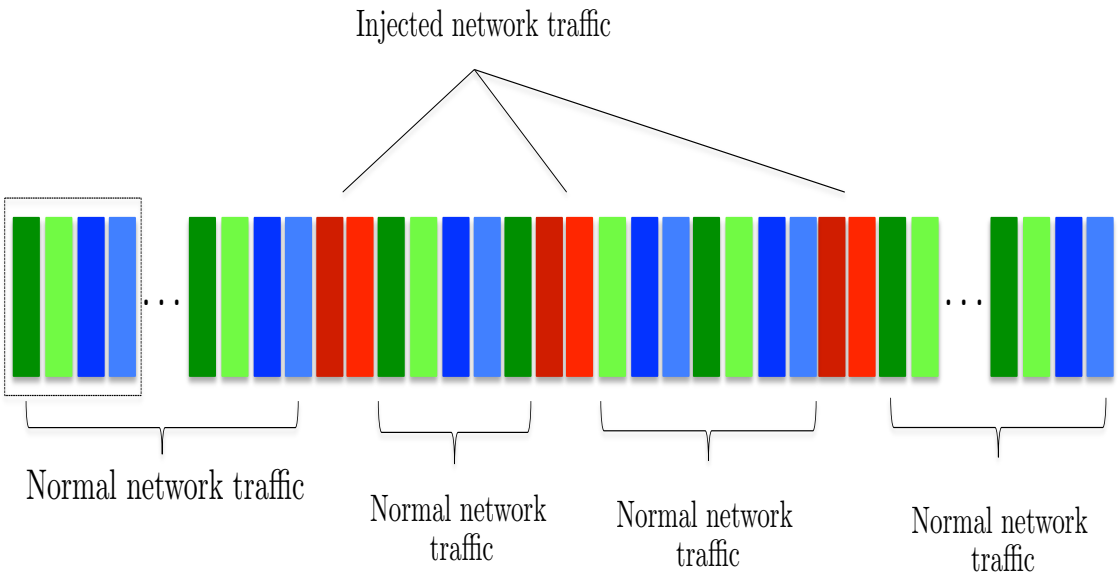

Figure 3.5.1: Sequence of packets in the network traffic

3.5.3 Attack Scenario 3

The attacker imitates the master controller's behavior by issuing packets that follow the expected communication packet sequence. The "write" command requests that the attacker launches aim either to activate the circuit breakers (status = 0), and isolate the power transformer when it is not necessary, or to prevent an activation of the circuit breakers (status = 1) when the power transformer's current exceeds the predefined pick-up current. Figure 3.5.2 shows the measured current of the power transformer by the sensors, and the subsequent actions concerning the status of the circuit breakers that are located beside the power transformer. Even though each packet constitutes a "legal" event on its own, the combination of the packets constitutes an illicit event since the physical constraints specified

by the normal operation of the physical system are not met. Our HC-NIDS identifies the attacker's activity by checking the consistency between the measured current of the power transformer and the status of the circuit breakers, and generates alerts that indicate the occurrence of activity that should be disallowed.

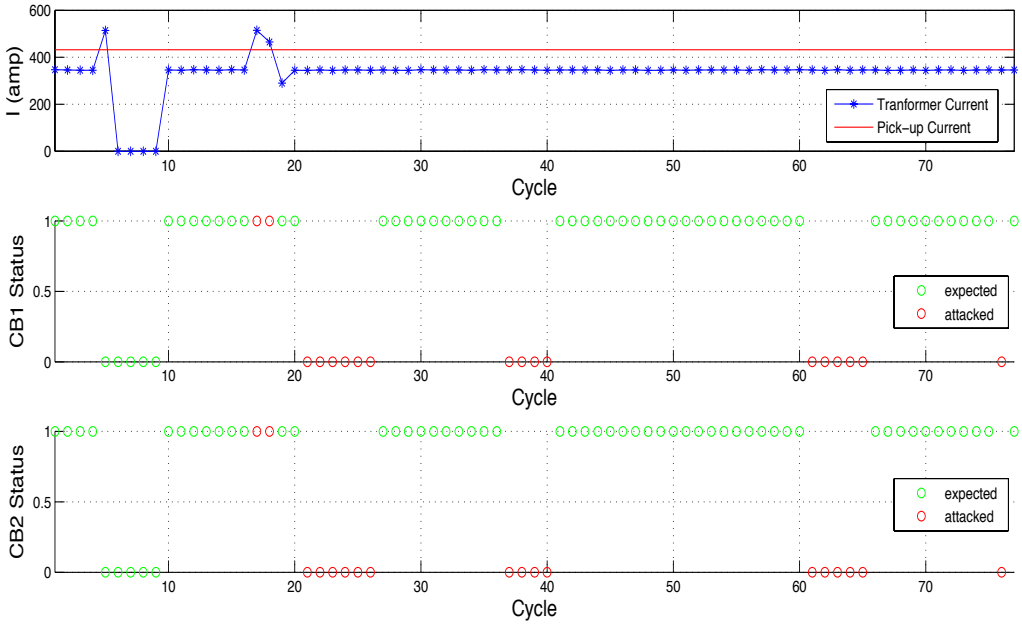

Figure 3.5.2: Transformer's current and status of circuit breakers

Chapter 4

An experimental Framework for Automation and Control Processes

4.1 State-of-the-Art

Despite the growing trend of employing cyber-physical systems (CPSs) in modern industry, such as the power grid, there are many open questions as to how best to perform experimental security studies on them. Testing security mechanisms, such as intrusion detection methods for cyber-physical systems, can often be inconclusive at best [50]. However, when attempting to draw any valid conclusions at all about security mechanisms, particularly mechanisms for cyber-physical systems in order to provide an adequate evaluation of security mechanisms, an experimental framework must include both "hardware-in-the-loop" and "cyber-in-the-loop" capabilities through a realistic implementation of the components of CPSs.

Therefore, in order to be able to analyze these types of systems, while also considering the physical behavior of the system, it is necessary to have an accurate and realistic mathematical model that describes the normal operation of the system [51, 52]. In addition to that, a testbed must either contain an actual physical device or be able to mimic the composite interactions that emulate the orderly behavior of cyber-physical systems, such as power

systems.

Recent research focuses on the implementation of frameworks that simulate the physical process and the control mechanism implemented in programmable logic controllers (PLCs) [53], rather than the implementation of a complete framework aiming on testing security mechanisms of CPSs. However, we believe that a bridge between theory and practice can only be useful when considering both sides of a cyber-physical system.

The Matlab/Simulink environment is a well known software for modeling and analyzing real-time dynamic systems, such as power systems. In addition, Matlab/Simulink supports a number of low-level communication protocols, such as TCP/IP. In this chapter, we present the experimental framework for testing intrusion detection systems for systems that we developed by leveraging the capabilities of the Matlab/Simulink environment in order to establish communication over industrial control protocols, i.e. Modbus TCP, with real PLCs.

4.2 Architecture of the Experimental Framework

Our experimental framework allows us to establish communication between a simulated physical process and a real PLC through an Ethernet interface that sends information via the Modbus TCP industrial control protocol. While many power systems rely on more advanced options, the choice of Modbus for our experimental validation is dictated by practical convenience, PLCs are inexpensive and a variety of software tools and libraries exist to communicate with them. However, the ideas discussed are applicable to many (it won't work the same way if they are encrypted communications!) of the application layer protocols used in industrial and control systems. The general architecture of our experimental framework is shown in Figure 4.2.1, and includes the following levels:

1. Simulink model of physical application
2. C MEX S-function that allows communication through the Modbus protocol
3. Emulation of control mechanism.

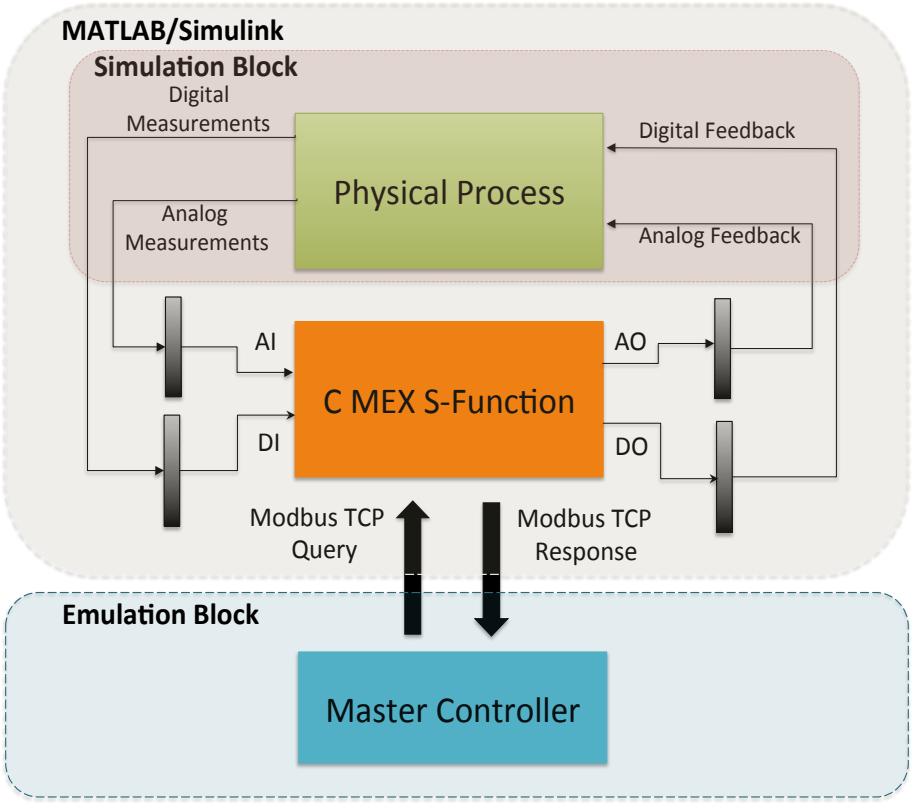

Figure 4.2.1: General architecture of the experimental framework

4.2.1 First Level: Simulink Model of the Physical Process

In the real world, power systems obey to physical laws that are completely known, such as Kirchhoff's and Ohm's laws, which must be satisfied by the physical systems in order to ensure that the systems are working properly. Therefore, a set of differential equations can be derived from the physical limits and laws of the system, that could be represented with ordinary differential equations (ODEs). However, these equations are typically too complex to handle, since second or higher order equations are rarely exactly solvable, especially in cases where the behavior of the system is complicated.

Therefore, common modelers handle the behavior of the complicated expressions that describe the behavior of the system, though linear-time invariant (LTI) systems on the frequency domain, a well-known theory called frequency response. In power systems, inputs

coming from the physical world are typically voltage and current signals. The type of signals in power systems, are narrowband signals centered at a frequency generally of 50 or 60Hz. Then, LTIs can be viewed as input/output systems where their response to the sinusoidal inputs is a sinusoidal output at the same frequency as the input. It is then enough to express the dynamics of the system through LTIs and model the frequency response from physical data.

The Matlab/Simulink environment provides a number of libraries that assist on representing and simulating the mathematical model that characterizes the behavior of a physical system. The Simulink models are a combination of graphical block diagrams that can be derived from the physical laws and behavior of dynamical systems. For that reason, the Matlab/Simulink environment is widely used for modeling dynamical systems, such as power systems, and for implementing realistic simulations of a physical process.

In this section, we present the first level that includes the development of a Simulink model of the physical system that should be controlled and protected by the master controller. Industrial controlled processes typically include a number of sensor and detector outputs, that are utilized to either indicate analog, i.e., current, or digital measurements, i.e., state of a switch. Afterward, the outputs of the physical process are given as inputs to the next level, which is responsible for the formation of Modbus TCP packets and data exchange with other controllers.

A number of actuator inputs are assigned to the simulation block and connects the master controller's outputs to the physical process (Figure 4.2.1). We refer to this type of interaction either as analog or digital feedback, based on the nature of the output signal, since these signals point out the result of the control mechanism implemented in the master controller.

In our work, as described in Chapter 3.2, we implemented the operation of a power transformer in the Simulink simulation environment where we assume that the transmission line, including a three-phase transformer, circuit breakers, measurement sensors, etc., corresponds to the physical process that we want to protect using the overcurrent protection function.

In addition, we consider current sensors that continuously record measurement samples of the current that flows on the transmission line, and feed the block that is responsible for the Modbus TCP packet transmission (S-function block) and which constitutes the slave controller. The Simulink model of the studied case that we described in Chapter 3.2 is shown in Figure 4.2.2.

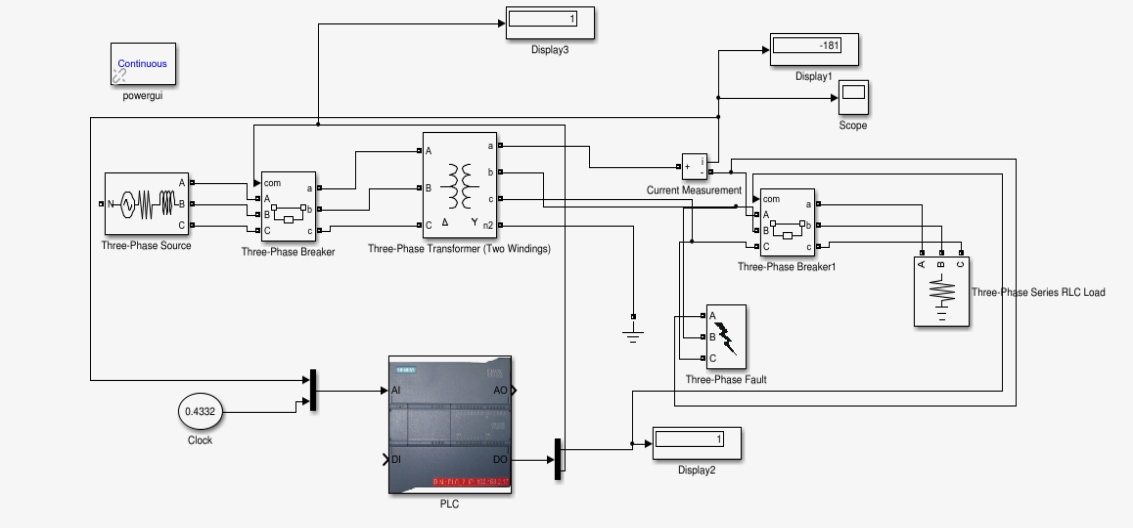

Figure 4.2.2: Simulink model of the power transformer on a transmission line

4.2.2 Second Level: C MEX S-Function

In the second level, we developed an S-function block that we implemented in ANSI C and which utilizes the open source Modbus TCP protocol library `libmodbus` [54] in order to simulate the communication behavior of a controller on a linux operating system. The S-function code is compiled using the Matlab mex compiler [55], which also creates a dynamically-loadable executable used by the Simulink model. The same approach could be used to allow communication through other industrial control protocols by including the S-function code for the appropriate protocol libraries instead of the Modbus TCP protocol library.

The S-function block receives as inputs the outputs of the physical process implemented using the Simulink environment, and allows communication with a second PLC. The sensors and detectors included in the dynamic model provide the S-function block with the appropri-

ate physical measurements, which are used in the next level in order to check the consistency of the physical model. Then, the S-function block is responsible for formulating packets that include the input measurements, and dispatches the Modbus TCP packets to the master controller whenever a "read" command query is received. The S-function block also performs the feedback control actions specified in a "write" query, which the master controller sends after the protection and control mechanisms are executed (Figure 4.2.1). In the case of the power transformer that we studied, the time of each measurement sample is also given as an input to the S-function block in order to process a specific number of samples and calculate an estimate of the specific measured variable within a given time frame.

The S-function begins by including and defining the necessary libraries, headers, and variables, including the following:

- libmodbus library
- width of analog input/output ports
- width of digital input/output ports
- client's IP
- sampling frequency.

Then, the S-function method *mdlInitializeSizes* is executed to set up the characteristics of the S-function block:

- assignment of analog inputs/outputs to specific ports
- assignment of digital inputs/outputs to specific ports
- data types of analog inputs/outputs
- data types of digital inputs/outputs.

The S-function method *mdlInitializeSampleTimes* indicates the sample rates, where in our case the value CONTINUOUS_SAMPLE_TIME is used for continuous sample rate, and then the *mdlStart* method is executed only once for initialization purposes. In the next step, the S-function *mdlOutputs* function method is executed and is responsible for acquiring the measurements from the Simulink model, formulating the Modbus packets, and computing the output signals of the S-function block that corresponds to the feedback of the control mechanism. The S-function function method *mdlTerminate* is called at the end of the code to perform actions such as emptying the memory. Figure 4.2.3 shows the sequence of the functions executed within the S-function block.

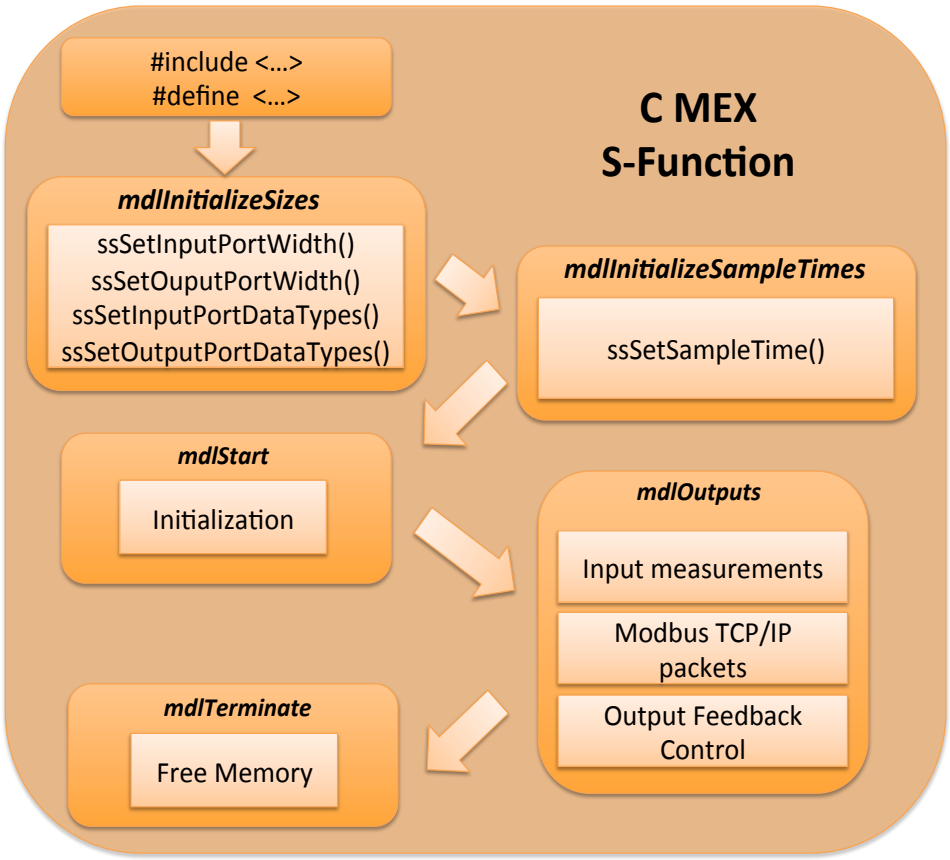

Figure 4.2.3: Sequence of methods executed in the S-function block

4.2.3 Third Level: Implementation of the Control Mechanism

The third level of our experimental framework corresponds to the master controller that performs the protection and control mechanisms related to the physical system. For the studied case described in Chapter 3.2, we implemented an emulation of a master controller in Ladder logic using a real PLC. In addition, in order to create a modular version of our experimental framework, we also created a simulation of the control and protection mechanism that we implemented in C. However, the basic concept can be applied to a variety of different types of control and protection mechanisms for power systems, and can be implemented either by emulation or simulation of the PLCs.

4.2.3.1 Emulation Using a Real PLC

Our implementation of emulation using a real PLC uses a Siemens SIMATIC S7-1200 PLC that acts as the master controller and performs the protection function and control mechanism of the physical process. This PLC supports both Ethernet and TCP/IP based communication protocols, such as the Modbus TCP/IP, allowing communication either as a Modbus TCP client (master) or server (slave). We implemented the control mechanism in Ladder logic using the SIMATIC Software Step 7 Basic.

In our case, we consider a master controller that initiates a connection and exchanges data with the simulated slave controller (as discussed in Chapters 4.2.1- 4.2.2). The master controller polls the slave controller in order to acquire the value of the input measurements obtained by the related sensors, such as the value of the current. Then, the protection control algorithm is executed and, based on the result, the master controller sends "write" queries to the slave controller indicating which control action should be performed, i.e., activate a circuit breaker in the case of fault on the power transmission line. Figure 4.2.4 shows the function that we used. The function block that we used is implemented in ladder logic and is included in the SIEMENS S7-1200 communication library, as well as the communication between the TIA Portal software and the S-function block introduced in Chapter 4.2.2.

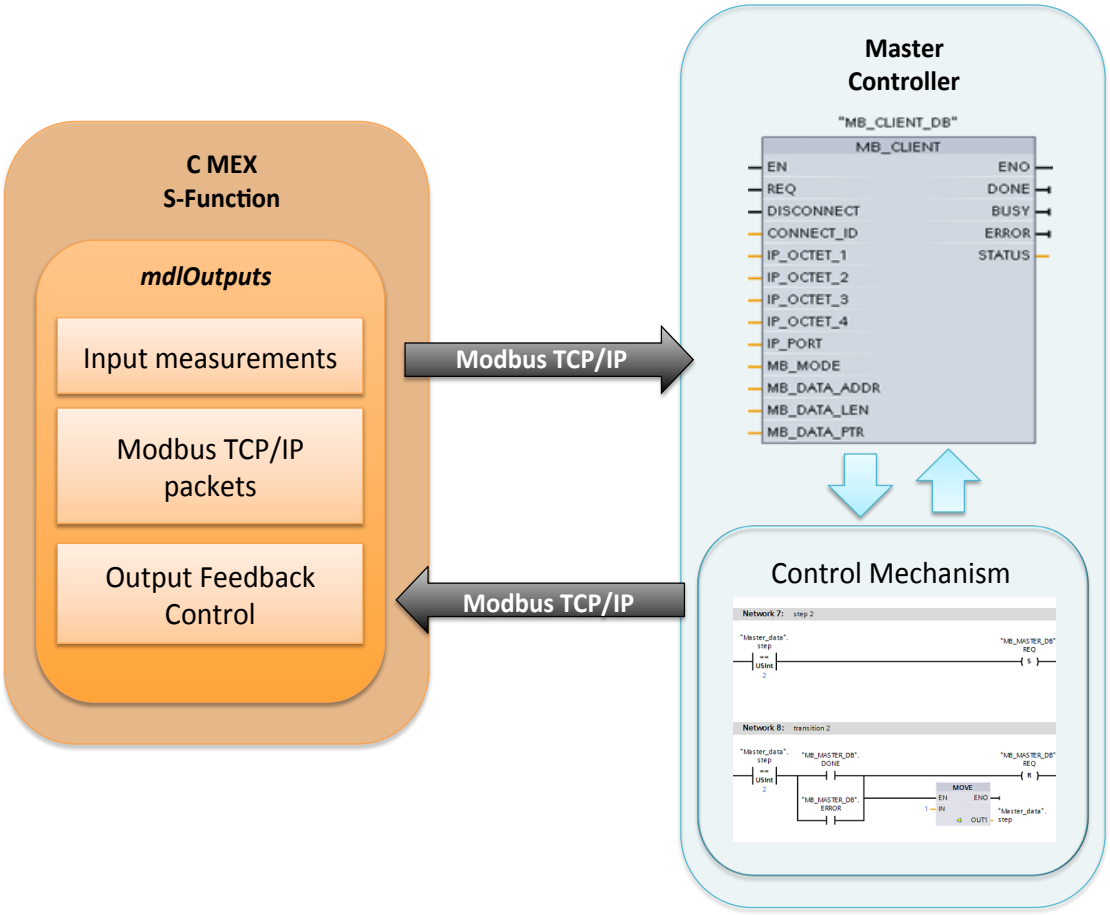

Figure 4.2.4: Communication between Matlab/Simulink and PLC

4.2.3.2 Simulation of a PLC Using libmodbus

The simulation implementation of the master controller was implemented in the C programming language and utilizes the Modbus protocol library `libmodbus` in order to allow communication through the Modbus industrial control protocol. The specific implementation constitutes a replica of the implementation of the protection and control mechanism in ladder logic using the real PLC.

The use of a simulation of the master controllers is a more flexible route when no access to real devices is possible, and it also provides substantially greater freedom in when choosing the industrial control protocol that will be used to establish communication and exchange data within the network of the controllers.

Similarly to the procedure described in Chapter 4.2.4, the master controller establishes a connection with the slave controller and dispatches "read" requests in order to obtain the value of current measurements. Then, the simulated master controllers perform the protection control algorithm related to the protection of the power transformer and dispatches "write" queries indicating the result of the protection mechanism via control actions. Also, in between the two different types of requests, the master controllers releases the connection. Figure 4.2.5 shows the communication between the simulated PLC written using libmodbus and the S-function block introduced in Chapter 4.2.2. In addition, Algorithm 2 shows the generalized sequence of the main functions executed in the C code for the studied case, where we assume that the master controller gets as inputs analog values, and returns digital values. However, the C code can be customized for different cases based on the requirements of the application, as well as the control mechanism, the code of which is not shown in Algorithm 2 since it can be only applied to a specific example.

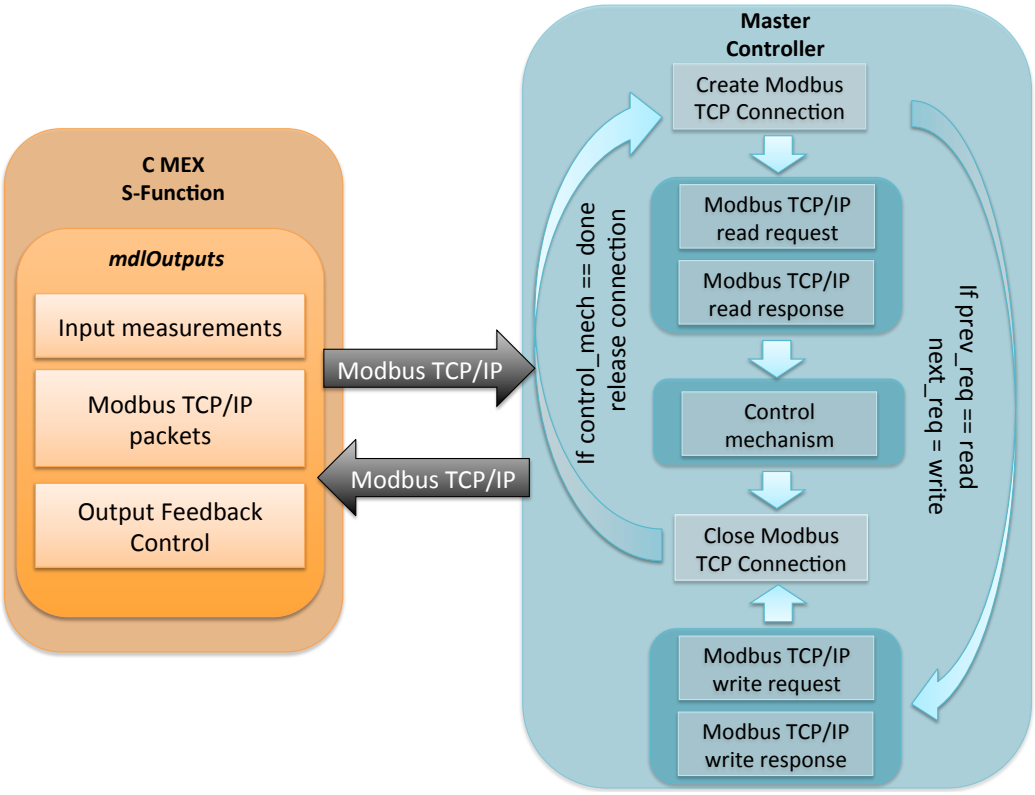

Figure 4.2.5: Communication between Matlab/Simulink and PLC

Algorithm 2 Simulation of a PLC

```
modbus_t *ctx
int rc
bool sts
char *modbusClientIP ← CLIENT_IP
int modbusPort ← PORT

ctx ← modbus_new_tcp(modbusClientIP, modbusPort)
if ctx == NULL then
    fprintf(Unable to allocate libmodbus context)
    return ← −1
end if
if modbus_connect(ctx) == −1 then
    fprintf(Connection failed)
    sts ← false
end if

rc ← modbus_read_input_registers


C code of control mechanism


modbus_close(ctx)
modbus_free(ctx)

ctx ← modbus_new_tcp(modbusClientIP, modbusPort)
if ctx == NULL then
    fprintf(Unable to allocate libmodbus context)
    return ← −1
end if
if modbus_connect(ctx) == −1 then
    fprintf(Connection failed)
    sts ← false
end if

rc ← modbus_write_bits

modbus_close(ctx)
modbus_free(ctx)
```

Chapter 5

Conclusions and Future Work

In conclusion, securing cyber-physical systems, such as power systems, is of undoubtedly great importance, due to the fact that damage to these systems could have severe impact not just on computers but devices, systems, and even the safety of people in the physical world. Therefore, in order to assess the effects of potential threats, and protect cyber-physical systems, it is useful to leverage important information related to the physical part of the system in addition to network communication.

This thesis highlights the importance of combining both network and physical information in the form of intrusion detection rules in order to ensure the reliable and secure operation of components of the power grid. The goal of our work is to augment, and not to replace, existing NIDS, by monitoring the physical activity within the network of the controllers, in addition to the communication rules that the system should obey. Although in this thesis we focus on power systems, we believe that is could be extended to cyber-physical systems that present a similar operational behavior.

However, while the approach presented in this thesis appears to be effective when considering part of a cyber-physical system, we should keep in mind that typically these types of systems consist of a large number of components that need to be coordinated in order to ensure the safe operation of the entire system. In most of our attack scenarios, we assume

that an attacker does not have complete knowledge about the physics of the system, where in the opposite situations our HC-NIDS should be enriched with more "detailed" policies that could allow us to detect more sophisticated attacks. In addition, our approach requires the exchange of valuable information related the physical part between the various controllers.

In addition to the assumptions and considerations mentioned above, there are some challenges, limitations, and open questions related to our approach and the way that we validated our HC-NIDS. In our experimental validation, we did not consider cases where the network traffic includes noise, i.e., traffic that is not related to the system under monitoring. Therefore, we believe that under noise, our HC-NIDS will produce a higher rate of false positives related the expected packet sequence. Another challenge of our approach, is that the hybrid automaton and intrusion detection rules based on that automaton must be customized for individual applications that obey different physical limits and settings.

Therefore, as a future effort, we plan to research means for generalizing our approach. Instead of focusing on the implementation of security policies for specific applications, by creating models based on the hybrid automata that designate the expected behavior of CPSs, we seek to research means for building generalized models that will still take both cyber and physical aspects of CPSs under consideration, as well the coordination orchestration of the various components that compose such systems. Then, the generalized models could be used to derive a set of security policies that follow the approach described in thesis, and could be included in a NIDS to enhance the stability of cyber-physical systems.

References

[1] Modbus-IDA, "Modbus application protocol specification v.1.1b," *Hopkinton, Massachusetts*.

[2] K. Curtis, "A DNP3 protocol primer," *DNP User Group*, 2005.

[3] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber attacks on SCADA systems," in *Proceedings of the 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computin Internet of Things (iThings/CPSCom)*. IEEE, 2011, pp. 380–388.

[4] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier," *White paper, Symantec Corp., Security Response*, 2011.

[5] Havex, http://www.f-secure.com/weblog/archives/00002718.html.

[6] C. McParland, S. Peisert, and A. Scaglione, "Monitoring Security of Networked Control Systems: It's the Physics," *IEEE Security & Privacy*, vol. 12, no. 6, 2014.

[7] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho, *Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems*. Springer, 1993.

[8] T. A. Henzinger, *The theory of hybrid automata*. Springer, 2000.

[9] "IEEE Standard Electrical Power System Device Function Numbers, Acronyms, and Contact Designations," *IEEE Std C37.2-2008 (Revision of IEEE Std C37.2-1996)*, pp. 1–48, Oct 2008.

[10] M. Parvania, G. Koutsandria, V. Muthukumar, S. Peisert, C. McParland, and A. Scaglione, "Hybrid Control Network Intrusion Detection Systems for Automated Power Distribution Systems," in *Proc. 1st International Workshop on Trustworthiness of Smart Grids (ToSG)*, 2014.

[11] G. Koutsandria, V. Muthukumar, M. Parvania, S. Peisert, C. McParland, and A. Scaglione, "A Hybrid Network IDS for Protective Digital Relays in the Power Transmission Grid," in *Proceedings of IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2014.

[12] R. Mackiewicz, "Overview of iec 61850 and benefits," in *Proceedings of the 2006 IEEE PES Power Systems Conference and Exposition*, 2006, pp. 623–630.

[13] R. Mitchell and I.-R. Chen, "A Survey of Intrusion Detection Techniques for Cyber Physical Systems," *ACM Computing Surveys*, vol. 46, no. 4, 2013.

[14] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Tech. Rep., 2000.

[15] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," *Proceedings of the IEEE symposium on security and privacy*, 2010.

[16] T. H. Morris, B. A. Jones, R. B. Vaughn, and Y. S. Dandass, "Deterministic intrusion detection rules for MODBUS protocols," in *Proceedings of the 46th Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2013, pp. 1773–1781.

[17] A. Carcano, I. N. Fovino, M. Masera, and A. Trombetta, "State-based network intrusion detection systems for SCADA protocols: a proof of concept," in *Critical Information Infrastructures Security*. Springer, 2010, pp. 138–150.

[18] R. Billinton and R. N. Allan, *Reliability Evaluation of Power Systems*. Plenum press: New York, 1996.

[19] A. Abur and A. G. Exposito, *Power system state estimation: theory and implementation*. CRC Press, 2004.

[20] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures," in *Proceedings of the IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2010.

[21] R. Chow, E. Uzun, A. A. Cárdenas, Z. Song, and S. Lee, "Enhancing cyber-physical security through data patterns," in *Proceedings of the Workshop on Foundations of Dependable and Secure Cyber-Physical Systems (FDSCPS)*, 2011, p. 25.

[22] M.-K. Yoon, S. Mohan, J. Choi, J.-E. Kim, and L. Sha, "Securecore: A multicore-based intrusion detection architecture for real-time embedded systems," in *Proceedings of the 19th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 2013, pp. 21–32.

[23] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, "Attacks against process control systems: risk assessment, detection, and response," in *Proc. ACM Symposium on Computer and Communications Security*, 2011, pp. 355–366.

[24] H. Lin, A. Slagell, Z. Kalbarczyk, P. W. Sauer, and R. K. Iyer, "Semantic security analysis of SCADA networks to detect malicious control commands in power grids," in *Proc. of the First ACM Workshop on Smart Energy Grid Security*, 2013, pp. 29–34.

[25] R. Berthier and W. H. Sanders, "Specification-based intrusion detection for advanced metering infrastructures," in *Proc. 17th IEEE Pacific Rim International Symposium on Dependable Computing*, 2011.

[26] J. Valente, C. Barreto, and A. A. Cardenas, "Cyber-physical systems attestation," in *Proceedings of the 2014 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2014, pp. 354–357.

[27] Z. Wang, X. Li, V. Muthukumar, A. Scaglione, S. Peisert, and C. McParland, "Networked loads in the distribution grid," in *Proceedings of the 2012 APSIPA Annual Summit and Conference*, vol. 2012, no. 1, 2012.

[28] J. D. Glover, M. Sarma, and T. Overbye, *Power System Analysis & Design, SI Version*. Cengage Learning, 2011.

[29] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer networks*, vol. 31, no. 23, pp. 2435–2463, 1999.

[30] G. T. Heydt, "The next generation of power distribution systems," *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 225–235, 2010.

[31] R. E. Brown, "Impact of smart grid on distribution system design," in *Proceedings of the 2008 IEEE Power and Energy Society General Meeting*, 2008, pp. 1–4.

[32] A. Abiri-Jahromi, M. Fotuhi-Firuzabad, M. Parvania, and M. Mosleh, "Optimized sectionalizing switch placement strategy in distribution systems," *IEEE Trans. Power Delivery*, vol. 27, no. 1, pp. 362–370, 2012.

[33] M. Hadley, N. Lu, and D. A. Frincke, "Smart-grid security issues," *IEEE Security and Privacy*, vol. 8, no. 1, pp. 81–85, 2010.

[34] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on cyber security for smart grid communications," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 998–1010, 2012.

[35] E. Bompard, P. Cuccia, M. Masera, and I. N. Fovino, "Cyber vulnerability in power systems operation and control," in *Critical Infrastructure Protection.* Springer, 2012, pp. 197–234.

[36] I. Lim, S. Hong, M. Choi, S. Lee, T. Kim, S. Lee, and B. Ha, "Security protocols against cyber attacks in the distribution automation system," *IEEE Trans. Power Delivery*, vol. 25, no. 1, pp. 448–455, 2010.

[37] D. Yang, A. Usynin, and J. W. Hines, "Anomaly-based intrusion detection for SCADA systems," in *Proc. of the 5th Intl. Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies (NPIC&HMIT 05)*, 2006, pp. 12–16.

[38] D. Wei, Y. Lu, M. Jafari, P. M. Skare, and K. Rohde, "Protecting smart grid automation systems against cyberattacks," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 782–795, 2011.

[39] S. Sridhar, A. Hahn, and M. Govindarasu, "Cyber–physical system security for the electric power grid," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 210–224, 2012.

[40] T. T. Tesfay, J.-P. Hubaux, J.-Y. Le Boudec, and P. Oechslin, "Cyber-Secure Communication Architecture for Active Power Distribution Networks," in *Proc. of the 29th ACM Symposium On Applied Computing (SAC)*, 2014.

[41] W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Computer Networks*, vol. 57, no. 5, pp. 1344–1371, 2013.

[42] M. Bishop and C. Gates, "Defining the insider threat," in *Proc. of the 4th Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW)*. ACM, 2008.

[43] EPRI, *Technical and System Requirements for Advanced Distribution Automation*. Palo Alto, CA, 2004.

[44] C. Ko, M. Ruschitzka, and K. Levitt, "Execution monitoring of security-critical programs in distributed systems: A specification-based approach," in *Proceedings of the 1997 IEEE Symposium on Security and Privacy*. IEEE, 1997, pp. 175–187.

[45] Siemens, "S7-1200 programmable controller," https://w3.usa.siemens.com/us/internet-dms/ia/AutomationComm/Automation/DOCS2/S7-1200_System_Manual_enUS.pdf.

[46] SIMATIC, "Programming with step 7," https://www.automation.siemens.com/doconweb/pdf/SINUMERIK_SINAMICS_02_2012_E/S7P.pdf?p=1.

[47] L. Xie, Y. Mo, and B. Sinopoli, "False data injection attacks in electricity markets," in *Proceedings of the First IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2010, pp. 226–231.

[48] X. Fu, B. Graham, R. Bettati, and W. Zhao, "Active traffic analysis attacks and countermeasures," in *Proceedings of the 2003 International Conference on Computer Networks and Mobile Computing (ICCNMC)*. IEEE, 2003, pp. 31–39.

[49] M. V. Deshpande, *Elements of electrical power station design*. I. Pitman & Sons, 1966.

[50] J. McHugh, "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by the Lincoln Laboratory," *ACM Transactions on Information and System Security (TISSEC)*, vol. 3, no. 4, pp. 262–294, November 2000.

[51] A. Kusiak, *Computational intelligence in design and manufacturing*. John Wiley & Sons, 2000.

[52] S. B. Morriss, *Automated manufacturing systems: Actuators, controls, sensors, and robotics*. Glencoe/McGraw-Hill, 1994.

[53] J. Martins, C. Lima, H. Martínez, and A. Grau, "A matlab/simulink framework for plc controlled processes," *Matlab-Modelling, Programming and Simulations*, p. 211, 2010, ABM Nasiruzzaman.

[54] Libmodbus, http://libmodbus.org.

[55] MATLAB, "Matlab mex compiler," http://www.mathworks.com/help/matlab/ref/mex.html.