Doctoral theses at NTNU, 2022:169

Livinus Obiora Nweke

Using Formal Methods for Modelling Cyber-Physical Systems Security

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of Philosophiae Doctor
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of Science and Technology

Livinus Obiora Nweke

Using Formal Methods for Modelling Cyber-Physical Systems Security

Thesis for the Degree of Philosophiae Doctor

Gjøvik, May 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU

Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

ISBN 978-82-326-6216-6 (printed ver.)
ISBN 978-82-326-5875-6 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2022:169

Printed by NTNU Grafisk senter

"Whatever your mind can conceive and believe, it can achieve."

(Napoleon Hill)

Declaration of Authorship

I, Livinus Obiora Nweke, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Signed:

(Livinus Obiora Nweke)

Date:

Preface

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Information Security and Communication Technology at the Norwegian University of Science and Technology, Norway. The study was carried out during the period from February 2019 to November 2021. The thesis is written on the basis of the published research papers and a research paper under review. The articles are reformatted to fit the thesis's structure and the contents of the original articles are self-contained.

Summary

The recent years have witnessed an increasing integration of physical systems with information and communication technology (ICT). This emerging field is usually referred to as cyber-physical systems (CPS) and has generated a lot of attention. These systems are sometimes called real-time systems because they have stringent quality of service (QoS) requirements. Also, the coupling of the physical and the cyber components entails that any malicious activity in the cyber components would have a devastating effect on the physical components which in turn may endanger the lives of the humans and the environment. For this reason, some CPS are also known as safety-critical systems. The application of CPS spans through several domains including power stations, large interconnected infrastructure, traffic systems, etc. Thus, this thesis explored how these physical systems can be incorporated and formally modelled to be able to capture the behaviour of CPS, specifically under attack.

First, this research focused on the communication channel while treating the internal state of CPS as a black box. The communication channel has been shown to be vulnerable to cyber attacks and for a number of these types of attacks, the timing behaviour and particularly the buffering capacity of the communication channel is critical. To address this problem, we employed queueing networks because they are appropriate models for capturing this sort of error conditions that leads to breaching of real-time constraints. We extended the existing models to include ways of representing the interaction of the different types of traffic and to understand how that can breach the QoS requirements. This is because the probability density function for the adversarial flow need not be the same as that of the regular traffic. We then applied one of the proposed models to study the effect of adversarial flow in software-defined industrial control networks.

Moreover, there are scenarios where it is necessary to study the internal state for the internal behaviour of the CPS, which means this cannot be captured using a queueing network type of model. Our interest in this regard was to describe the internal behaviour of CPS in such a way that the asynchronous communication takes a prominent role. Hence, instead of having a model that relies on state machine, where we have global knowledge of the state; we employed process calculus model which effectively allowed us to look at the distributed state and how one state can influence another process including the uncertainty of the distributed state. We first developed an ad-

versary model based on the process calculus model which allows reasoning over process states, placement of adversarial entities and communication behaviour. Then we extended the process calculus model by embedding an algebraic representation of Attack-Defence Trees (ADT) and augmenting this by the addition of a partial ordering over the constituents of the ADT within the embedding, offering an elegant mechanism to extend ADT to ordering and time-related attacks. Lastly, we expanded on the process calculus model to provide a finer grain model of the actual interactions inside the physical systems of CPS.

Acknowledgments

> Gratitude is not only the greatest of virtues, but the parent of all others.
>
> CICERO, 106-43 BC

I would like to express my sincere gratitude to my principal supervisor Professor Stephen D. Wolthusen for his commitment, guidance, support and advice throughout the period of this research and writing of the thesis. He also allowed me to explore additional research areas of my interest which has enabled me to establish my personal stance within the academic community. I could not have imagined having a better supervisor and mentor for my PhD study.

Also, I have been extremely lucky to have great support system during my PhD journey. Starting with my amazing colleagues with whom I shared the same office space, to both the academic and administrative staff of the Department of Information Security and Communication Technology at the Norwegian University of Science and Technology (NTNU), Norway and the entire Norwegian society. Even though the journey was sometimes difficult and lonely, their great support made it a lot more easier. And for this, I will always be immensely indebted.

Finally, my profound gratitude goes to my biggest cheerleading team: my daughter, wife, mother, family and friends for supporting me throughout the journey. It has not been a smooth ride, however, their support made the heavy load much more lighter to carry.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> There is nothing more difficult to take in hand, more perilous to conduct, or more uncertain in its success, than to take the lead in the introduction of a new order of things.
>
> N. MACHIAVELLI

1.1 Motivation

Over the past few years, there has been massive deployment of systems which combine computation and communication capabilities to control and monitor physical systems. This new emerging field is referred to as cyber-physical systems (CPS) and has generated widespread interest. The main goal of CPS research is "*to integrate knowledge and engineering principles across the computational and engineering disciplines (networking, control, software, human interaction, learning theory, as well as electrical, mechanical, chemical, biomedical material science, and other engineering disciplines) to develop new CPS science and supporting technology*" [13]. CPS are also the main components of critical infrastructure and as such, are essential for the maintenance of vital societal functions. Any malicious activity that could disrupt the proper functioning of CPS would lead to devastating consequences. Therefore, CPS security has become a major concern not only for the government, academia and industry but also for the general public.

CPS security incidents are increasingly becoming pervasive and according to Gartner analysts [23] would continue to rise in the coming years. For example, a recent ransomware attack forced the closure of the largest United States (U.S) fuel pipeline [70]. This attack is not strictly against CPS, however, CPS suffered as a result of not being properly isolated from IT (information technology) systems. If a non-targeted ransomware incident could cause CPS to shutdown, it would be difficult to imagine if the attack was actually targeted at the CPS. This demonstrates the far-reaching threats to CPS and the need to rethink how these threats may be addressed considering the impact they would have if realised.

The impact of a successful CPS attack goes beyond the financial ramifications to include loss of human life, litigation, regulatory fines, reputation loss and more. Gartner predicts that the financial impact of CPS attacks resulting in fatal causalities will reach $50 billion by 2023 [23]. A practical example of such impact is the astronomical rise in the average U.S gasoline price following the cyber attack that forced the closure of the largest fuel pipeline [20]. This attest to the fact that the impact of a successful CPS attack could be far-reaching. Consequently, there is an urgent need to understand the risks that CPS represent and the need to dedicate more efforts towards the design and implementation of a more secure CPS.

Nonetheless, several attempts have been made towards addressing CPS security. For example, the design challenges of CPS have been identified in [41] and a discussion on the past, present and future of CPS with focus on models has been presented in [40]. An analysis of the security issues at various layers of the CPS architecture, risk assessment and the different techniques for securing CPS has been discussed in [10]. And a survey, which captures and systematizes the existing research on CPS security under a unified framework has been presented in [31]. The goal of these attempts is to pave the way for building models that are abstract enough to be applicable to study CPS security.

Moreover, modelling CPS security has also been attempted in several studies [46, 47, 50, 49, 48]. There is a huge body of literature that discusses different methods - both formal and informal methods - for modelling CPS security. These different modelling techniques have their merits and demerits as they try to capture the security issues in CPS at different levels of abstraction. However, given the complexities in modelling CPS security, it still remains an open research problem.

1.2 Aim and Scope

In this thesis, we have studied the way that adversaries behave inside CPS and have tried to model selective aspects of these adversarial behaviours. First, we focused on the communication channel while treating the internal state of CPS as a black box. The communication channel has been shown to be vulnerable to cyber attacks. For a number of these types of attacks, the timing behaviour and particularly the buffering capacity of the communication channel is critical. Therefore, the timing related aspects of CPS have been investigated in this study, to understand how they can breach the stringent quality of service requirements.

Also, the behaviour of the internal state of CPS was considered in this study. The internal state of CPS consists of two components: the internal state of the cyber system (the computational elements of CPS) and the internal state of the physical system (sensors and actuators used to monitor and

control physical objects). Whilst the internal state of the cyber system of CPS refers to the interaction inside the computational elements of CPS governed by control laws, the internal state of the physical system of CPS represents the dynamics inside the physical processes influenced by the measurements that are communicated. In this study, we were interested in describing the internal behaviour of CPS in such a way that the asynchronous communication takes a prominent role. Hence, instead of having a model that relies on a state machine, where we have global knowledge of the state; we considered models which effectively allow us to look at the distributed state and how one state can influence another process including the uncertainty of the distributed state.

1.3 Research Questions

This study was guided by the following research questions, which were formulated in line with the above-mentioned motivation and scope.

- 1st Research Question (RQ1): How can queueing network models be used as a formal method for modelling timing related aspects of CPS e.g., the buffering capacity of the communication channel?
- 2nd Research Question (RQ2): How can process algebraic methods be used as a formal method for modelling the synchronisation issues between the internal state of the cyber system and the internal state of the physical system of CPS?
- 3rd Research Question (RQ3): How can the models developed in RQ1 and RQ2 be adapted to investigate adversarial actions in CPS?

1.4 Background

This section presents a general discussion on CPS and CPS security. It also provides useful background on the formal methods employed in this thesis to facilitate a better understanding of the remaining materials presented in this study.

1.4.1 Cyber-physical systems (CPS)

The term CPS refers to an emerging class of systems which are concerned with the intersection of computation, communication, and physical objects. It was first coined by Helen Gill at the National Foundation of Science, United States in 2006 [40] and ever since, have attracted a lot of attention from the government, industry and academia. Several efforts have been made to define CPS but there is still not a generally accepted definition.

The common feature of the existing definitions of CPS is that they are systems which combine computation and communication capabilities to control and monitor physical systems. This definition indicates that CPS consist of computational elements, which analyses and processes information received about the state of the physical system and issues appropriate control commands to ensure that the desired state of the physical system is maintained; the communication elements, which facilitates the interaction between the computational and physical elements; and the physical elements, which captures the state of the physical system and executes the control commands from the computational elements. The interaction of simplified CPS components is depicted in Figure 1.1, where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ denote the plant's state and input vectors, respectively, while $y \in \mathbb{R}^p$ is the plant's output vector obtained from measurements of p sensors from the set $S = \{1, 2, ..., p\}$. Also, $w_t \in \mathbb{R}^n$ is the process noise and $e_t \in \mathbb{R}^m$ is the measurement noise.

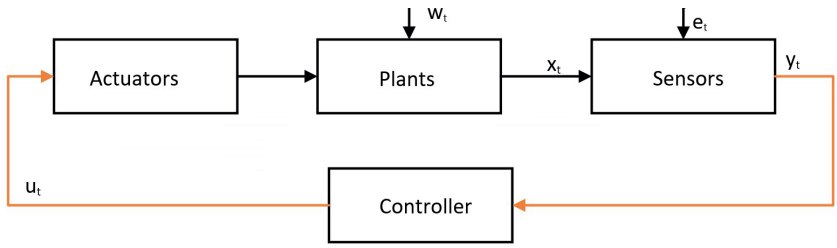


Figure 1.1: Interaction of simplified CPS components

In recent years, there has been an exponential growth in the number of physical objects controlled by information and communication technologies (ICT), which attests to the increasing popularity of CPS. This can be attributed to the proliferation of low-cost and increased-capability sensors of increasingly smaller form-factor; the availability of low-cost, low-power, high-capacity, small form-factor computing devices; large Internet bandwidth; and the continuous improvements in energy capacity, energy harvesting and alternative energy sources [65]. Hence, it is easy to notice that CPS have permeated every aspects of our lives as their applications span through several domains including electrical power grids, water and wastewater management, oil and gas sector, traffic systems, implantable medical devices, aerospace, building and environmental control, factory automation, and many other domains.

Additionally, there are popular terms such as Internet of Things (IoT), the Internet of Everything and Industry 4.0, that are closely associated with CPS. This is because these terms are related to the exponential growth in the integration of physical systems with ICT. However, Lee in [40] observe that

"*the term "CPS" is more foundational and durable than all these, because it does not directly reference either implementation approaches (e.g., the "Internet" in IoT) nor particular applications (e.g., "Industry" in Industry 4.0). It focuses instead on the fundamental intellectual problem of conjoining the engineering traditions of the cyber and physical worlds*".

CPS have several properties which make their design and analysis difficult to achieve using the existing methods. For example, CPS have time constraint because the physical processes being controlled are usually time-aware and deadline sensitive [72]. Also, the analysis and design of CPS need to take into consideration the upstream and downstream dependencies of the component systems [27]. This is further exacerbated by the unreliable and unpredictable environment that the CPS would have to operate [41], which in turn creates uncertainty in the general operation of CPS. Thus, these properties - timing, uncertainty, and dependencies that exist between the entities of CPS - have given rise to security challenges that need to be considered in the design and implementation of CPS.

1.4.2 CPS security

CPS security has generated widespread interest in recent years. A substantial amount of effort have been made in both the industry and academia towards addressing the security challenges in CPS. This is because of the pervasiveness of CPS in every aspects of our lives. The implication of this is that any security incident that affects CPS could lead to physical harm to people, destruction of property or environmental disasters. In addition to the most recent security incident described in section 1.1, the following paragraphs describe other real life examples of security incidents in CPS to further motivate why the analysis, detection, and identification of security issues in CPS are of utmost important.

One of the widely reported security incidents in CPS is the attack against Natanz uranium enrichment plant in Iran, where the infamous malware known as Stuxnet escaped the digital realm and wreaked physical damage to a CPS [85]. Another earlier reported security incident in CPS is the sewage attack of 2000 [76]. In this attack, a disgruntled contractor exploited SCADA radio-controlled sewage equipment for the Marcoochy Shire Council in Queensland, Australia, to dump 800,000 litres of raw sewage into local parks . These two examples of the earlier reported security incidents in CPS are indications that security issues in CPS are not new but have evolve in scale and magnitude as shown in recent reported security incidents.

Some of the recently reported security incidents in CPS include the successful attack against a power substation, north of the city of Kiev, Ukraine [26]. This attack blacked out a portion of the Ukrainian capital equivalent to a fifth of its total power capacity. Also, ransomware attacks have been deployed in recent years against CPS and these have blocked natural gas

pipeline supplies [18], shut down logistics operations [29] and disrupted steel production operations [25]. A much more recent reported security incident in CPS is the attack on a water treatment plant in Florida, USA [44]. In this attack, an unknown hacker remotely accessed a computer at the water treatment plant and attempted to increase the amount of sodium hydroxide in the water supply to potentially dangerous level. These few examples of the reported security incidents in CPS indicate that there is an urgent need to rethink how CPS security should be addressed.

To provide an overview of the security challenges in CPS, lets consider a simplified CPS architecture showing the possible attack points. The attack points 1-8 depicted in Figure 1.2 represents the possible means an attacker could exploit to compromise a CPS. These attack points have been described as follows by Cardenas in [16]:

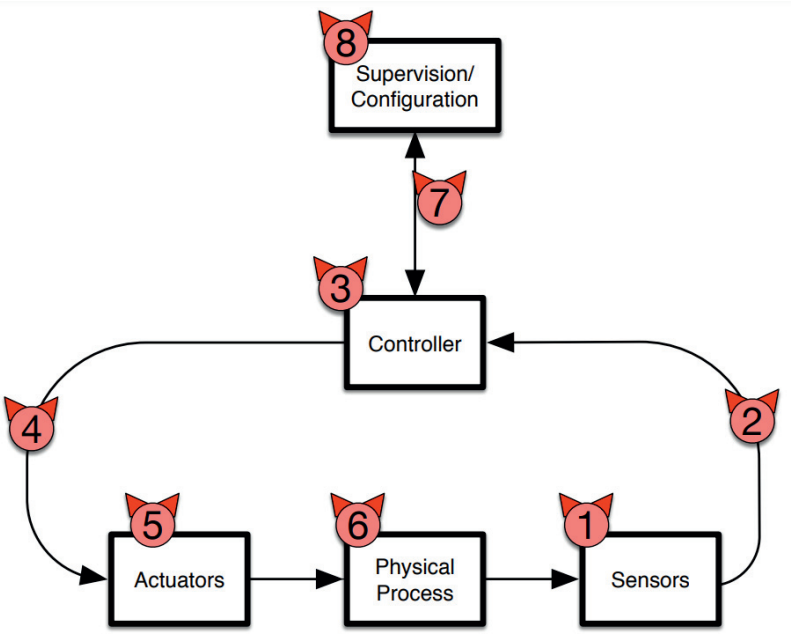


Figure 1.2: Attack Points in a CPS

[16]

- Attack 1 represents an attacker who has compromised a sensor (e.g., if the sensor data is unauthenticated or if the attacker has the key material for the sensors) and injects false sensor signals, causing the control logic of the system to act on malicious data. An example of this type of attack has been considered by Huang et al. [30].

- Attack 2 represents an attacker in the communication path between the sensor and the controller, who can delay or even completely block the information from the sensors to the controller, so the controller loses observability of the system (loss of view), thus causing it to operate with stale data. Examples of these attacks include denial-of-service attacks on sensors [8] and stale data attacks [37].

- Attack 3 represents an attacker who has compromised the controller and sends incorrect control signals to the actuators. An example of this attack is the threat model considered by McLaughlin [51].

- Attack 4 represents an attacker who can delay or block any control command, thus causing a denial of control to the system. This attack has been considered as a denial-of-service to the actuators [8].

- Attack 5 represents an attacker who can compromise the actuators and execute a control action that is different to what the controller intended. Notice that this attack is different to an attack that directly attacks the controller, as this can lead to zero dynamics attacks. These types of attacks are considered by Teixeira et al. [78].

- Attack 6 represents an attacker who can physically attack the system (e.g., physically destroying part of the infrastructure and combining this with a cyber-attack). This type of joint cyber and physical attack has been considered by Amin et al. [9].

- Attack 7 represents an attacker who can delay or block communications to and from the supervisory control system or configuration devices. This attack has been considered in the context of supervisory control and data acquisition (SCADA) systems [32].

- Attack 8 represents an attacker who can compromise or impersonate the SCADA system or the configuration devices, and send malicious control or configuration changes to the controller. These types of attacks have been illustrated by the attacks on the power grid in Ukraine where the attackers compromised computers in the control room of the SCADA system [26] and attacks where the configuration device of medical devices has been compromised [28].

The main objective of CPS security is to ensure that an attacker is unable to exploit these attack points to compromise the operation of a CPS. This has proven to be a difficult undertaking because most of the existing methods for modelling CPS security are designed for IT (information technology) systems where the behaviour of the system is well-understood. In CPS however, we have timing, uncertainty and dependencies that exist between the entities of the system that must be taken into consideration. As a result of

these properties, several approaches have been proposed for modelling CPS security [46, 47, 50, 49, 48]. For this thesis, queueing networks and process algebraic methods are used as the formal methods for modelling CPS security. Therefore, the following subsections provide a background on these methods.

1.4.3 Queueing networks

A queueing network can be defined as a collection of service centres representing the system resources which are used to provide service to a collection of customers that represent the users [75]. The basic queueing systems have been defined in queueing theory and applied to analyse networks. Queueing theory started by the study of queues, devising analytical mechanisms and tools for the design and evaluation of the performance of queueing systems [75]. To characterise a queueing system, there is a need to identify the probabilistic properties of the arrival processes, service times and service disciplines [75]. Conventionally, the arrival process is characterised by the distribution of the inter-arrival times of the customers. These inter-arrival times are usually assumed to be independent and identically distributed random variables. They are denoted by *A(t)*:

$$A(t) = P(\textit{interarrival time} < t).$$

The service time is used to express how long the service will take. It is usually assumed that the service time for a customer is independent and does not depend upon the arrival process. Another common assumption about service time is that it is exponentially distributed. Its distribution function is denoted by *B(x)*:

$$B(x) = P(\textit{service time} < x).$$

The service discipline is used to decide how the next customer waiting on the queue is served. The most common methods used include: First-in, First-out (FIFO); Last-come, First-out (LIFO); Random Service (RS); Priority.

Kendall's notation is the standard notation used to describe and classify queuing systems [75]. It is given as:

$$A/B/m/K/n/D,$$

where; A is the distribution function of the inter-arrival times, B is the distribution function of the service times, m is the number of servers, K is the capacity of the system, n is the population size, and D is the service discipline.

It is common practice to denote exponentially distributed random variables by M meaning Markovain or memoryless [63]. If the population size

and the capacity are infinite, the service discipline is FIFO and is usually omitted. Thus, M/M/1 denotes a system with Poisson arrivals, exponentially distributed service times and a single server [75]. Basic queueing systems are used to describe the system as a unique resource, but to describe the system as a set of interacting resources; queueing networks are usually used [75]. Queueing networks are models where service requests arrive at service stations to be served and when service requests arrive, a free service station handles the requests. However, if all the service stations are busy when the service requests arrive, the service requests are queued for a waiting time until one of the service stations is free. Similar to the basic queueing systems, queueing networks employ number of service requests and servers, the size of the waiting queues and the queueing discipline as the important parameters for performance evaluation. Queueing networks have been shown to be an appropriate tool for system performance evaluation and have also been shown as being helpful in the modelling of attacks against distributed systems.

1.4.4 Process algebra

Process algebra is a mathematical framework for formal modelling concurrent systems. It provides several formalisms for describing the behaviour of a system and can be used to describe the evolution of the system in terms of labelled transitions. The origin of process algebra can be traced back to the early seventies of the twentieth century, and a brief history has been provided in [12]. Whilst there are different varieties of process algebra, they all share the following common features [57]:

- A minimal set of carefully chosen operators capturing the relevant aspect of systems behaviour and the way systems are composed in building process terms;
- A transition system associated with each term via structural operational semantics to describe the evolution of all processes that can be built from the operators;
- An equivalence notion that allow one to abstract from irrelevant details of systems descriptions.

In this thesis, π-calculus proposed by Robin Milner [53] is used as the process algebraic method for modelling CPS security. The π-calculus provides a formal mechanism for modelling communication among processes over dynamic links [71] and has since been extended and applied in several studies including modelling different types of security processes [1, 2]. A system in the π-calculus is made up of independent processes that communicate via channels. A channel is an abstraction of the communication link

and is referred to by name. Names are the simplest entities of the π-calculus and there are infinite number of names, represented by lowercase letters (x, y, z, etc.).

Processes in the π-calculus evolve by performing actions. These capabilities for action are expressed via the prefixes, of which there are four kinds:

$$\pi := \ \overline{x}y \mid x(z) \mid \tau \mid [x = y]\pi$$

The first capability is to send the name y via name x, and the second to receive any name via x. The third capability refers to an internal action or an unobservable action. And lastly, the fourth is a conditional capability where the capability π is executed if x and y are the same. The set of processes can also be defined by the syntax given in Table 1.1.

Table 1.1: Syntax of π-calculus

Term	Semantics
$P ::=$	Processes
0	empty process
$\overline{x}z.P$	output
$x(y).P$	input
$P + Q$	choice
$P\|Q$	parallel composition
$!P$	replication
$\nu x.P$	restriction
τ	silent function/action

- A composition $P|Q$ behaves as if processes P and Q are running in parallel. This implies that the two processes can evolve separately at the same time and can operate on the channels to communicate with each other and with the outside the network.
- The basic interaction is defined using $\overline{x}z.P$ that defines an output process that is ready to output on channel x, or $x(y).P$ that defines an input process that is ready to receive a value over channel x.
- The replication $!P$ behaves as an infinite number of copies of P running in parallel.
- The name restriction operator ($\nu x.P$) is a process that makes a new, private name x, and then behaves as P.
- τ represents the internal (silent) action of a process that is not observable outside the scope of the process.

- 0 is the empty process.

To briefly describe the use of the π-calculus for modelling systems, lets consider the following example which is similar to the illustration provided by Parrow in [61]. Suppose we have a system which consists of three processes, namely: a controller, a resource and an agent. The controller controls access to the resource and the agent needs to access it. We can represent the original state of the controller using a communication link x. The agent interacts with the controller via another link y to have access to the resource. After this interaction, access to the resource will be transferred to the agent. We can express the communication among these processes using the π-calculus as follows: the controller that sends x along y is $\overline{y}x.C$; the agent that receives some link along y and then uses it to send data along it is $y(a).\overline{a}z.A$. The interaction we have described so far can be formulated in the π-calculus as follows:

$$\overline{y}x.C \,|\, y(a).\overline{a}z.A \;\; \xrightarrow{\tau} \;\; C \,|\, \overline{x}z.A$$

1.5 Related Work

This section presents techniques for modelling CPS security that are related to the methods adopted for this thesis, a discussion on their limitations and justification for using queueing networks and process algebra as the formal methods in this study. It also discusses the existing literature on the use of queueing networks and process algebra for modelling CPS security.

1.5.1 Modelling CPS Security

Modelling CPS security is a very challenging task as we have observed already. This is because of the complex interaction between the cyber and physical components of CPS. Several approaches have been proposed over the years for modelling CPS security. In this subsection, we are interested in the approaches that allows for the derivation of quantitative results or obtaining a parameter that would be useful in understanding the behaviour of CPS under adversarial influences. For example, hybrid automata facilitate the combination of finite state transition systems with discrete variables (cyber components) and continuous variables (physical components) [15]. They have been deployed to reason about the security properties of CPS [5]. Timed automata [7] have also been used to model timing properties of CPS. They are similar to hybrid automata but are embedded with a finite set of real-valued clocks. Stochastic hybrid automata has been proposed in [43] as a framework to model the interaction between continuous dynamics, discrete dynamics and probabilistic uncertainty. It has also been employed in

[56] for impact analysis of cyber-physical attacks on a water tank system via statistical model checking.

Another approach for modelling CPS security that is related to the methods adopted for this thesis is the use of linear temporal logic (LTL) [64] to specify the behaviour of CPS. LTL is used to specify abstract constraints on the behaviour of dynamic systems, which can then be used to express the temporal properties of the systems for model checking. The goal of the model checking is to verify if the system satisfies a set of required properties. LTL formulas are usually translated to automata so as to be checked in an explicit state-based model checker. This formalism has been applied in [58] to study security related issues in CPS.

Unfortunately, most of these approaches described in the preceding paragraphs are not able to express the timing aspects of CPS in a simplified and focused way. This is because they employ state machines in modelling CPS and it is difficult to enumerate all the state space of CPS (in CPS, all the required variables usually are not present only state estimates). Although stochastic hybrid automata is able to provide support for quantitative analysis of CPS, it can only bound the error of the analysis which does not guarantee a completely correct analysis [56]. For this reason, a queueing network model is deployed in this thesis because it has been shown to be not only effective for representing the timing related aspects of systems but also, is able to do so in a simplified and at a fairly abstract level. Further, a queueing network model facilitates the efficient bounding of parameters than those obtainable using the stochastic hybrid automata. A queueing network model also allows for discussing the interplay between the regular traffic and the adversarial traffic, which is an important prerequisite for understanding the behaviour of CPS under adversarial influences.

Moreover, these approaches introduce a level of determinism that do not exist in CPS. This is because they are not able to properly model pattern regularity, rules or substate machines. They lack the expressive power of showing that not only there is a transition between states but that the states follows processes that transition, which leads to other processes taking on certain roles. Further, they do not capture the internal behaviour of CPS in a realistic way to show the internal transitions that are not driven by external influences. We have tried to address this by saying that the internal behaviour of CPS are only visible from the observation (from measurements) and there is a need for an explicit way of synchronising them because global timing does not exist in distributed systems such as CPS. Consequently, given the expressive power of the process algebraic methods, they are used in this thesis as the explicit way of understanding these synchronisation issues between the internal state of the cyber system and the internal state of the physical system of CPS.

1.5.2 Modelling CPS security using queueing networks

In the past few years, software-defined networks (SDN) have been proposed for enhancing CPS security. SDN is a current network architecture in seeking to decouple the control plane from the data plane. Unlike in a traditional network architecture where control and data plane are embedded in the networking devices, SDN separates roles such that networking devices can become purely forwarding devices with the forwarding instructions pushed to them via the control plane and allowing the use of commodity components [22]. They are being adopted widely and are also likely to be deployed as the infrastructure of systems with critical real-time properties such as industrial control systems (ICS). However, their ability to guarantee the quality of service (QoS) requirements of such networks in the presence of adversarial flow still needs to be investigated. One of the methods that can be employed for such investigation is to consider attacks that can be analysed at the flow abstraction level using queueing networks.

Queueing networks have long been employed to study the performance and QoS characteristics of networks [75]. Understanding the performance of SDN through its analysis using queueing networks is of particular importance for systems with strong QoS requirements. These requirements include particularly the delay, loss, and jitter parameters and the respective requirements for different types and classes of service. This thesis concentrates on queueing distributions for modelling SDN as a method for characterisation and categorisation as this is highly pertinent to the adaption for modelling CPS security.

Modelling CPS security involves taking into account the complex interaction between the cyber and physical components of CPS. To model this interaction, it would require consideration at different levels of abstraction. For example, the timing behaviour of the interaction between the cyber and physical components of CPS can be modelled using queueing networks while hiding other details. In this regards, the approach employed in this thesis is to consider SDN applied to industrial control networks and the strong periodic network pattern of the network traffic using queueing networks. This periodic behaviour and occasionally some bursty traffic as seen in CPS such as industrial control networks is captured using probability density function to consider the messages being generated by sensors and consumed by actuators and how they can be handled. The physical system in this scenario is considered as a process that consumes and generates messages. Using queueing networks, this thesis then model how many messages are there, at what interval do the messages arrive, and is there a risk that the system will be overwhelmed by messages (that can come from attackers but can also come from the physical process itself). This is very useful for dealing with questions relating to denial of services and quality of services types of attacks that may be targeted at CPS. A comprehensive

survey of the existing methods in the analysis of SDN using queueing networks has been provided in [59], which is one of the research outcomes of this thesis. For completeness, a summary of these methods is presented in the following paragraphs.

In most of the existing approaches for the analysis of SDN using queueing networks, modelling based on a Poisson arrival, one exponential sever, infinite FIFO queue and unlimited customer population (M/M/1 distribution) are usually deployed [33]. Even though these are very strong assumptions that may not be satisfied by real systems, they provide useful insights which may be used to study how real systems will perform given certain parameters. Attempts have been made to use other type of distributions to address the limitations of M/M/1 distribution. For example, the G/M/1/K distribution has been used in [73, 74]. In this type of queueing distribution, the inter-arrival times are independent and identically distributed with general distribution, and service times are independent and exponentially distributed. Also, the system is made of a finite waiting space and the arriving customers are served on a FIFO basis.

The remaining type of queueing distributions that have been used in the existing literature include the $M^X/M/1$ distribution [81], the M/Geo/1 distribution [77] and the M/G/1 distribution [34]. The $M^X/M/1$ distribution is a type of M/M/1 distribution where the arrival stream forms a Poisson process and the batch size is a random variable. Whilst the M/Geo/1 distribution comprises of Poisson arrival process and a service time that obeys geometric distribution, the M/G/1 distribution also has an arrival process with a Poisson distribution but the service time for each customer is generally distributed.

So far, the existing methods described in the preceding paragraphs, have focussed on relatively straightforward models for characterising the SDN behaviour. This appears to be a gap as a more precise characterisation of both arrival processes and service time distributions for regular operation, configuration changes, and adversarial action is thus far not being considered. To address this issue, this thesis proposed ways in which the models can be extended to study attacks that are based on arrival rates and service time distributions of flows in SDN. This is critical to understand vulnerabilities and security requirements for different deployments of SDN architectures such as software-defined industrial control networks.

Moreover, the behaviour of different deployments of SDN architectures has been studied using queueing networks. The SDN-based cloud computing architecture [82] and SDN-based network function parallelism in the could [17] have been examined using queueing networks. Also, queueing networks have been utilised for studying the behaviour of SDN in ultra dense networks [14] and satellite communication networks [42] and for the performance modelling and analysis of SDN under bursty multimedia traf-

fic in [52]. However, none of these works have considered software-defined industrial control networks and the strong periodic patterns of network traffic. Even though the authors in [52] use Markov modulated Poisson process (MMPP) in their study, they made simplifications about packet departure process of the MMMP queue to allow for a tractable analytical model. In contrast to all these works, this thesis exploits queueing networks to analyse the behaviour of network traffic in software-defined industrial control networks.

The authors in [83] develop a model of intelligent attacks which can employ primitives to infer the internal state of the flow table of SDN architecture to plan attacks more efficiently. While the internal state of the flow table of SDN architecture was considered in the work, the research done in this thesis treated the internal state of the flow table of SDN architecture as a black box. Also, the work relates only to performance design of SDN architecture while this thesis went further to consider SDN application in industrial control networks and the strong periodic pattern of network traffic in such networks. In addition, generative adversarial networks (GAN) have been deployed to study flow-based network traffic in the existing literature. For example, the authors in [67] propose a method for generating a realistic flow-based network traffic. However, GAN are only suitable for processing continuous attributes while the queueing network model employed in this thesis is able to model the periodic and occasionally bursty network traffic as seen in CPS such as industrial control networks.

SDN applied to industrial control networks is referred to as software-defined industrial control networks. Recent advances in networking technology have witnessed a gradual shift in industrial control networks from serial based communication to Ethernet technology. This provides the opportunity of leveraging the benefits of SDN for industrial control networks because the correctness of industrial control system protocols rely on the guaranteed QoS. However, SDN not only offers benefits but also raises questions about the ability to satisfy the guaranteed QoS in the presence of adversarial flow. To investigate this, well-established results from queueing theory and recent results from [79] are used to obtain a realistic characterization of interaction between the data plane and the control plane. Then, the analytical model is combined with one of the proposed queueing network models for modelling adversarial flow to study the effect of adversarial flow in software-defined industrial control networks.

1.5.3 Modelling CPS security using process algebra

Process algebraic methods have been deployed in several studies for modelling CPS security [39, 38]. They provide a high-level description of the interactions between the processes within CPS, which allows reasoning over adversarial influences. This subsection provides a review of the literature in

modelling CPS security using process algebra that are related to the problems addressed in this thesis.

One of the methods that can be used to study CPS security is to consider the assumptions built into the adversarial models employed for such a study. Adversarial models are well-established for cryptographic protocols, but distributed real-time protocols used in CPS have requirements that these abstractions are not intended to cover. Attacks against these distributed real-time protocols are becoming prevalent and have been studied over the past years. Hence, it is imperative to rethink these adversarial models taking into account the unique features CPS environment presents. For example, the adversary model described by Dolev and Yao in [19] assumes that the attacker have complete control over the network. It has been shown that this assumption can no longer hold in CPS environment [46]. Thus, efforts have been made in the past by several authors [39, 38] to formalise attacks for CPS given the stringent QoS requirements.

A formal adversary capability model for SCADA environments has been described in [46]. The authors utilise π-calculus variant to reason about adversarial actions and argue that the Dolev-Yao model and its variants are not suitable for capturing the capabilities of an attacker in a SCADA environment because of the segmented network architecture and real-time processing. Another interesting work on formalization of attacks was presented in [80] where the authors proposed an adversary model which could be used to study the security promises of real-time systems. In this work, the attacker is assumed to be able to compromise both physical and cyber weaknesses of the systems and the adversary model was able to capture the capabilities and spatial distribution of the adversary.

In [60] the authors used a state-based stochastic model to formalize the security properties of real-time systems. They assumed that the system had Markovian property and considering that general probability distributions are assigned to its transitions, the resulting model is a semi-Markov chain. Further, the proposed model is then parametrized based on a time distribution describing the attacker and the system behaviours over time. A generalized attacker and attack models for real-time systems were presented in [4]. The authors described an attacker model for real-time systems and used the attack models that were obtained from the attacker model to generate parametrized attack methods for real-time systems. The authors in [24] used the formalism of discrete event systems modelled as a finite state automata to reason about the problem of synthesizing an attack strategy for real-time systems. The model presented in the work was able to capture a class of deception attacks, where the attacker is capable of modifying a subset of sensor reading in order to mislead the supervisor and forcing the system into an undesirable state.

Furthermore, a formal approach for characterizing attacks in real-time

systems was presented in [39]. The authors deployed formal methods to capture interactions in a real-time system and to reason about how the system may be attacked. They used a hybrid process calculus to characterize both the system and the attacks against the system. The adversary model used in this work assumed that the adversary is not able to compromise the communication, but may compromise physical devices. Different from the works presented so far, this thesis presents an adversary model specifically for the IEC 61850 environment. Like most of the works, it argues that the Dolev-Yao model is not suitable for modelling attacks against real-time systems. Hence, π-calculus variant is used to first capture the multicast, publish-subscribe model and then reason about how adversarial actions can compromise the system.

A significant amount of research effort has been dedicated towards the analysis, detection and identification of security issues in CPS. For example, Mo et al. [55] develop a model-based techniques capable of detecting integrity attacks on the sensors of a control system. Also, Pasqualetti et al. in [62] present attack detection and identification in CPS and analyse the core monitoring limitations for CPS under attack modelled by linear time-invariant descriptor systems with exogenous inputs. Several other works that have considered security issues in CPS, such as denial-of-service attacks [8], replay attacks [69], and false data injection attacks [54]. However, the threat model used in most of these works employs a custom construct which makes them difficult to use in different environments. This thesis proposes a model which offers a set of constructs that can be used to decompose threats in CPS.

Threat modelling of CPS has been attempted in several works in the literature. One of the earliest attempts to threat model CPS came from Zalewski et al. [84]. They propose the use of a discrete time Markov chain (DTMC) model to characterize the transitions between the secure and insecure states of CPS. The authors argue that quantifying the probabilities of transitions between secure and insecure states will allow for the derivation of important inferences about the security related features of CPS. Then, the conventional threat modelling techniques (STRIDE, DREAD, CVSS) are applied in the work, to assign the probabilities of transitions between the states. These techniques capture threats at a certain level of abstraction which does not allow for reasoning over the communication between assets and their timing property.

Martins et al. in [45] present a tool to perform a systematic threat modelling for CPS using a real-world temperature monitoring system as a case study. The authors use the Generic Modelling Environment for the creation of domain-specific modelling for threat analysis CPS. Also, they extended and deployed Microsoft SDL Threat Modelling Tools to model, identify, and mitigate threats in a systematic way for the proposed CPS. A model to repre-

sent CPS threats using patterns that are related to architectural aspects of the CPS is described in [21]. The author shows how to extend the misuse pattern to characterise cyber-physical threats and how to enumerate and unify cyber-physical threats.

A threat modelling framework for CPS using STRIDE is presented in [35]. The authors demonstrate the applicability of the proposed framework using a real synchrophasor-based synchronous islanding test-bed in the laboratory. They show that an adversary can achieve a specific malicious goal by exploiting threats at different locations in the system. Also, they illustrate that by identifying component level vulnerabilities and their potential physical consequences, STRIDE can be applied to address such challenges. Almohri et al. in [6] present threat modelling of medical CPS. The authors consider the roles of stakeholders and system components. They use this understanding to sketch an abstract architecture of medical CPS and then show the various threat modelling options.

CPS threats and vulnerabilities analysis for train control and monitoring systems is presented in [66]. The authors evaluate vulnerabilities and characteristics of railway threat landscape including potential threats, threat actors and motivations. Also, they examine the direct impacts and cascading consequences of threats on the whole system as well as risk produced. Atif et al. in [11] describe cyber threat analysis for CPS. They employ a data-driven approach to threat model CPS. A machine learning algorithm based on K-Nearest-Neighbour (K-NN) is used in this work, to ascertain the threat category faced by the CPS considered.

Attacker models for CPS have been discussed in [68]. The authors present a literature review of the attacker models for CPS and define a taxonomy of ten different features that they applied to the literature. Also, a generalized attacker and attack models for CPS has been proposed in [4] and have been employed to investigate the impact of single-point cyber attacks on a Secure Water Treatment (SWaT) system in [3]. Unlike these attacker and attack models presented in these works where the authors utilise descriptive threat modelling techniques, the approach adopted for this thesis allows us to be analytical. It facilitates the description of the adversary behaviour at a level of details that allows us to effectively explore the range of parameters or the behaviour of a system. This enables us to infer not only the necessary conditions for an attack to be successful but also the sequencing of events or the way events have to be order for the attack to be successful.

In contrast to all the works described so far, this thesis presents threat modelling of CPS using an applied π-calculus. The applied calculus has also been used for attack modelling in [39]. The main differentiator of this approach to the existing literature on threat modelling and attack modelling in CPS is twofold. First, the method has the potential to be automated i.e., to use a mechanism that allows us to efficiently explore the proof space. This

is because it is possible to take an applied π-calculus model and translate it into a theorem prover or prover assistance and then perform the reasoning automatically. However, it requires that the specification of CPS is sufficiently precise that it can be used to reason over the semantics. To generate the applied π-calculus model that represent the CPS, we consider the behaviour of the cyber and physical system and encapsulating that into the applied π-calculus model as processes. This is done by hand because one needs to understand enough of both the cyber and physical system to say that this level of abstraction is sufficient to capture the relevant behaviour. Second, the approach allows us to analyse the threats to CPS in a more precise way. It enables us to capture the pre-conditions that are applicable to certain types of threats. This is because a CPS will not always be vulnerable: there will be some states where manipulating a variable will have an effect and there will be other states where manipulating the same variable will not have an effect. The method allows us to represent these states in the form of processes and the interaction between these processes and to reason about the likelihood of an attacker finding the system in a critical state to launch an attack for adverse effects.

Also, cyber-physical attacks and physics-based attacks have been studied in recent works using process algebra as the formal method. The work in [39] use process algebraic method to present theoretical foundations that can facilitate reasoning about CPS and cyber-physical attacks. Similar to the approach used in this thesis is the formal approach to physics-based attacks in CPS presented in [38]. The authors consider both integrity and denial of service attacks to sensors and actuators of CPS, and on the timing aspects of these attacks. However, the synchronization behaviour of the cyber and the physical system described in their work is not precise, and the interaction within the physical system is not properly captured. Unlike the work in [38], this thesis uses a π-calculus to model a CPS so as to capture the synchronization behaviour of the cyber and the physical system and to provide a more finer grain model of the physical system. This then allows us to describe the influences of the variables within the physical system on the operation of the CPS.

1.6 Research Methodology

The research methodology employed in this thesis follows the conventional scientific research process described in [36] and it is depicted in Figure 1.3. Starting with the research problem, which is to investigate how CPS security can be modelled; the related literature was extensively reviewed. This led to the formulation of the research questions in section 1.3. Afterwards, the system model and assumptions were defined. Queueing networks and process algebraic methods were then used as the formal methods for the

analysis of the model to obtain the results presented in this thesis. Although simulations and experimentations are alternative approaches that may be used for this study, formal modelling using queueing networks and process algebraic methods permits the study of a wider range of configurations and parameters as well as the optimization not only of performance but also an understanding of the severity of attacks. Also, simulation-based security studies are not able to explore every possible choice of parameters as obtainable using queueing networks and process algebraic methods, and they usually rely on computer codes that are prone to errors, poor programming and can be computationally expensive.

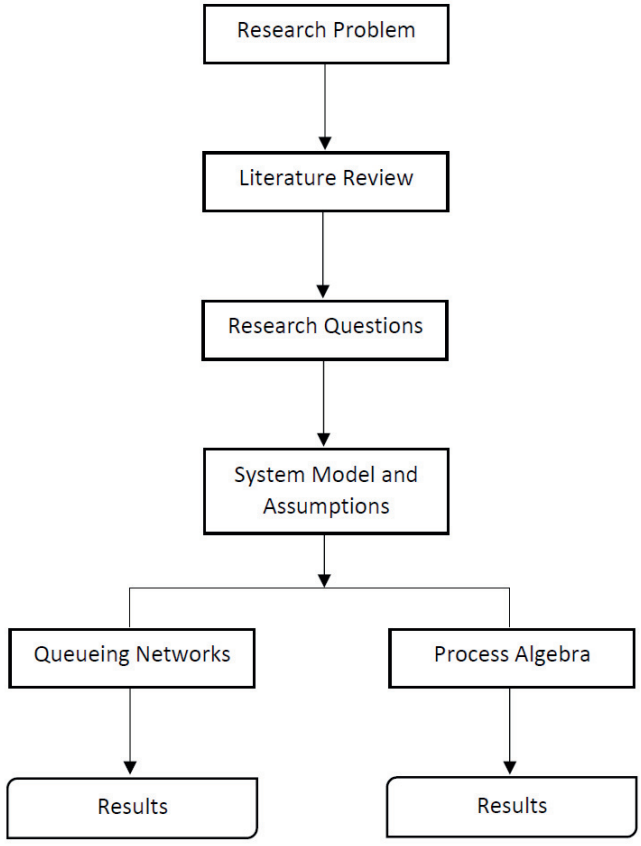


Figure 1.3: Research Methodology

1.6.1 Queueing networks

Understanding the behaviour of CPS through their analysis using queueing networks is highly pertinent to the adaption for modelling CPS security. This study first reviewed the existing literature on the analysis of SDN using queueing networks. The research questions that were addressed in this literature review are as follows:

- What are the current knowledge gaps and limitations in the use of queueing networks for modelling SDN?
- How can the existing queueing network models used for analysing SDN be extended to study attacks that are based on arrival rates and service time distributions?

Although the article search for the literature review was not restricted to a specific time, it was conducted between May 2019 to July 2019. The following key terms were employed in the search: modelling OR software-defined networking OR using AND queueing network models, in English. The retrieved articles were then evaluated and selected based on the research questions. Studies not related to the research questions, not published or written in other languages (not English) were excluded from the review. And to ensure rigour and quality, all the selected studies were indexed in Scopus and Web of Science database.

Based on the literature review, a classification scheme was developed to categorize the existing approaches. Subsequently, the constraints of the existing methods were identified and ways in which they can be extended to study attacks that are based on arrival rates and service time distributions of flows in SDN were proposed. One of these proposed models was then used to study the effect of adversarial flow in software-defined industrial control networks. Among the arguments that motivated the proposed models is that the distribution of a malicious traffic need not follow the distribution assumed for the normal traffic because of the way the attacker would craft the malicious packet to evade detection. This is because it would not be beneficial for an attacker to use the same distribution as the normal traffic and the need for the defender to be aware that the distributions of both the malicious and normal traffic may not be the same.

To study the effect of adversarial flow in software-defined industrial control networks using queueing networks, an appropriate queueing network model was developed. This model took into consideration the strong periodic patterns of network traffic in software-defined industrial control networks. Recent results from [79] were used to obtain a realistic characterization of interaction between the data plane and the control plane. Finally, the analytical model was combined with one of the proposed queueing network

models for modelling adversarial flow to evaluate the effect of adversarial flow in software-defined industrial control networks.

1.6.2 Process algebra

There are scenarios where to facilitate CPS security, it is necessary to study the internal state for the internal behaviour of the CPS, which means this cannot be captured using a queueing network type of model. Our interest in this regard is to describe the internal behaviour of CPS in such a way that the asynchronous communication takes a prominent role. Hence, instead of having a model that relies on state machine, where we have global knowledge of the state; this thesis employed process algebra which effectively allows us to look at the distributed state and how one state can influence another process including the uncertainty of the distributed state. Firstly, an adversary model based on the process algebraic model was developed which allowed reasoning over process states, placement of adversarial entities and communication behaviour. The developed model is difficult to automate because automation requires an extremely precise specification of the target system - which is nearly impossible to obtain for most protocols. Also, the proposed model does not include all attacks to the IEC 61850 that could potentially exist as new attacks are likely to be discovered in the future. To demonstrate the use of the proposed model, a simple case of a replay attack against the publish/subscribe GOOSE/SV subprotocol, showing bounds for non-detectability of such an attack was deployed. This is because a replay attack is relevant to the behaviour of IEC 61850 protocol. And to consider other attacks, the relative positions of the attacker in the adversary model would have to change.

Secondly, a review of the existing asset-centric threat modelling methods was conducted. The literature review aimed to address the following research questions:

- What are the current knowledge gaps or limitations in asset-centric threat modelling methods?
- What additional approach can be employed to increase the effectiveness of asset-centric threat modelling methods?

The article search for the literature review was not restricted to a specific time, but it was conducted between November 2019 to December 2019. The following key terms were employed in the search: asset-centric AND threat modelling methods, in English. Articles retrieved from the search were then evaluated and selected based on the research questions. Studies not related to the research questions, not published or written in other languages (not English) were excluded from the review. And to ensure rigour and quality, all the selected studies were indexed in Scopus and Web of Science database.

From the literature review, it was observed that the intuitive reasoning approach employed in the threat modelling approaches is not sufficient to threat model CPS where uncertainty, timing and dependencies between the entities exist. To address this limitation, a model was developed which extended the process calculus model by embedding an algebraic representation of Attack-Defence Trees (ADT) and augmenting this by the addition of a partial ordering over the constituents of the ADT within the embedding, offering an elegant mechanism to extend ADT to ordering and time-related attacks. The ADT as employed in this thesis was not used as a formal tool in its own right but rather, a part of the ADT was chosen and then the applied π-calculus model was manually created for the problem under consideration. The ADT was constructed before the π-calculus semantics because it allows us to perform direct search of the state space of the CPS we have identified as particularly interesting. Also, we express the ordering of the leaves of the ADT by imposing partial ordering over the interaction between the leaves. The modelling approach was illustrated using the case of an electrical substation fragment in which components communicate via the IEC 61850 protocol.

Lastly, another model was developed which expanded on the process calculus model to provide a finer grain model of the actual interactions inside the physical systems of CPS. This is because it allows us to describe the influences of the variables within the physical systems. To show the utility of the proposed model, a case study of on-load tap-changer (OLTC) control mechanism was used. The justification for this is that it exhibits the timing and state dependencies properties, which is a common feature shared by most CPS. The interactions between the different components of the OLTC control mechanism were captured using the process calculus. Then, execution traces was used to show the soundness of the system. In addition, the possible physical attacks against the OLTC control mechanism was investigated using the process calculus model. The main threats to validity observed for the method described in the previous paragraph and this paragraph is that the internal transitions of physical system of CPS are not driven only by external influences; there are also dynamics inside the physical system that are difficult to capture.

1.7 Summary of Contributions

In the bid to address the research questions specified in section 1.3, taking into account the background presented in section 1.4 and the limitations of the existing literature identified in section 1.5, and using the research methodology described in section 1.6; this thesis has resulted in the contributions outlined in this section.

1.7.1 List of included publications

The following research papers were either published or under review as a result of the research conducted during this PhD work and figure 1.4 shows how they are related to the research questions.

A. Nweke, L. O. and Wolthusen, S. D. Resilience Analysis of Software-Defined Networks Using Queueing Networks. International Conference on Computing, Networking and Communications (ICNC), IEEE, 2020. DOI: 10.1109/ICNC47757.2020.9049712

B. Nweke, L. O. and Wolthusen, S. D. Modelling Adversarial Flow in Software-Defined Industrial Control Networks Using a Queueing Network Model. 2020 IEEE Conference on Communications and Network Security (CNS), IEEE, 2020. DOI: 10.1109/CNS48642.2020.9162191

C. Nweke, L. O.; Weldehawaryat, G. K. and Wolthusen, S. D. Adversary Model for Attacks Against IEC 61850 Real-Time Communication Protocols. 2020 16th International Conference on the Design of Reliable Communication Networks DRCN 2020, IEEE, 2020.
DOI: 10.1109/DRCN48652.2020.1570604241

D. Nweke, L. O. and Wolthusen, S. D. A Review of Asset-Centric Threat Modelling Approaches. International Journal of Advanced Computer Science and Applications, 11(2), 1-6, 2020.
DOI: 10.14569/IJACSA.2020.0110201

E. Nweke, L. O.; Weldehawaryat, G. K. and Wolthusen, S. D. Threat Modelling of Cyber-Physical Systems Using an Applied π-Calculus. International Journal of Critical Infrastructure Protection 35 (100466), 2021.
DOI: 10.1016/j.ijcip.2021.100466

F. Nweke, L. O.; Weldehawaryat, G. K. and Wolthusen, S. D. Corrigendum to Threat Modelling of Cyber-Physical Systems Using an Applied π-Calculus. International Journal of Critical Infrastructure Protection, 2022.

G. Nweke, L. O. and Wolthusen, S. D. A process algebraic approach to modelling cyber-physical systems security, (Under Review).

1.7.2 Additional publications not included

The following research papers and technical report were either published or under review as a result of the parallel studies conducted during this PhD work, but are not included in this thesis.

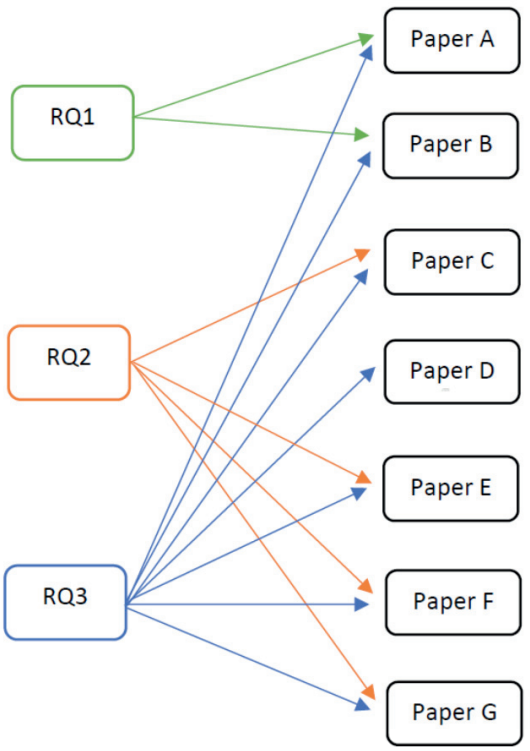


Figure 1.4: Relation between included papers and research questions

1. Nweke, L. O.; Wolthusen, S. D. and Mancini, L. V. A Framework for the Validation of Network Artifacts. 12th Norwegian Information Security Conference (2019)

2. Shukla, A.; Katt, B. and Nweke, L. O. Vulnerability discovery modelling with vulnerability severity. In: 2019 IEEE Conference on Information and Communication Technology. pp. 1–6. IEEE (2019). DOI: 10.1109/CICT48419.2019.9066187

3. Nweke, L. O. and Wolthusen, S. D. Ethical implications of security vulnerability research for critical infrastructure protection. 15th International Conference on Wirtschaftsinformatik (2020)

4. Nweke, L. O.; Yeng, P.; Wolthusen, S. D. and Yang, B. Understanding attribute-based access control for modelling and analysing healthcare professionals' security practices. International Journal of Advanced Computer Science and Applications 11(2), 683–690 (2020). DOI: 10.14569/ijacsa.2020.0110286

5. Nweke, L. O. and Wolthusen, S. Legal issues related to cyber threat information sharing among private entities for critical infrastructure protection. NATO CCDCOE 12th International Conference on Cyber Conflict (2020)

6. Yeng, P. K.; Nweke, L. O.; Woldaregay, A. Z.; Yang, B. and Snekkenes, E. A. Data-driven and artificial intelligence (ai) approach for modelling and analyzing healthcare security practice: A systematic review. In: Intelligent Systems Conference (IntelliSys) 2020 (2020)

7. Nweke, L. O. and Wolthusen, S. D. A Holistic Approach for Enhancing Critical Infrastructure Protection: Research Agenda. International Conference on Emerging Applications and Technologies for Industry 4.0 (EATI 2020)

8. Øver, H.; Øverlier, L.; Franke, K. and Nweke L. O. Technical Report. Analyse: Mørketallsundersøkelsen 2020.

9. Bokolo, A. J; Nweke, L. O. and Al-Sharafi, M. A. Applying Software-Defined Networking to Support Telemedicine Health Consultation During and Post Covid-19 Era. Health and Technology 11(2):395–403 (2020). DOI: 10.1007/s12553-020-00502-w

10. Abomhara, M.; Yayilgan, S. Y.; Nweke, L. O. and Székely, Z. A comparison of primary stakeholders' views on the deployment of biometric technologies in border management: Case study of SMart mobILity at the European land borders. Technology in Society 64:1-12 (2021). DOI: 10.1016/j.techsoc.2020.101484

11. Nweke, L. O. A Survey of Specification-based Intrusion Detection Techniques for Cyber-Physical Systems. International Journal of Advanced Computer Science and Applications 12(5) (2021)

12. Bock, A.; Gulden, J.; España, S.; Jahn, K.; Nweke, L. O. and Richter, A. The Ethics of Information Systems: The Present State of the Discussion and Avenues for Future Work. 29th European Conference on Information Systems (ECIS 2021)

13. Yeng, P. K.; Nweke, L. O.; Yang, B.; Fauzi, M. A. and Snekkenes, E. A. Artificial Intelligence–Based Framework for Analyzing Health Care Staff Security Practice: Mapping Review and Simulation Study. JMIR Medical Informatics (2021). DOI: 10.2196/19250

14. Nweke, L. O.; Bokolo, A. J.; Gibson, M. and Nwigwe E. Investigating the effectiveness of a HyFlex cyber security training in a developing country: A case study. Education and Information Technologies (2022). DOI: 10.1007/s10639-022-11038-z

15. Nweke L. O.; Abomhara M.; Yayilgan S. Y.; Comparin D.; Heurtier O. and Bunney C. A LINDDUN-Based Privacy Threat Modelling for National Identification Systems. IEEE Nigeria Section 4th International Conference on Disruptive technologies for Sustainable Development (NIGERCOM 2022)

16. Shukla, A.; Katt, B.; Nweke, L. O.; Yeng, P. K. and Weldehawaryat, G. K. System Security Assurance: A Systematic Literature Review. ACM Computing Surveys, (Under Review)

1.7.3 List of major contributions

A. *Resilience Analysis of Software-Defined Networks Using Queueing Networks*

In this article, a literature review on the use of queueing networks in the analysis of software-defined networks (SDN) is conducted. The goal of this review is to identify gaps in the existing literature and to propose ways to address the identified gaps. SDN are being adopted widely and are also likely to be deployed as the infrastructure of systems with critical real-time properties such as Industrial Control Systems (ICS). This raises the question of what security and performance guarantees can be given for the data plane of such critical systems and whether any control plane actions will adversely affect these guarantees, particularly for quality of service in real-time systems. So far, many studies have applied queueing networks to general performance analysis of most SDN architectures. However, there appears to be a lacuna in existing work as a more precise characterization of both arrival processes and service time distributions for regular operation, configuration changes, and adversarial action is thus far not being considered. This article then shows ways in which the existing models need to be extended to study attacks that are based on arrival rates and service time distributions of flows in SDN. The contributions of this article can be summarized as:

i. Analysis of the existing literature on the use of queueing networks in the study of SDN.

ii. Identification of the appropriate models for characterizing the behaviour of SDN controllers and switches.

iii. Extension of the existing models to include ways of representing the interaction of different types of traffic (regular and adversar-

ial traffic) and to understand how that can breach the quality of service requirements.

iv. Illustration of how the models proposed may be applied to study attacks that based on arrival rates and service time distributions of flows in SDN.

B. *Modelling Adversarial Flow in Software-Defined Industrial Control Networks Using a Queueing Network Model*

This article employs queueing networks as the formal method to investigate the ability of SDN to guarantee the quality of service (QoS) requirements of industrial control networks in the presence of adversarial flow. SDN has been proposed for enhancing the security of industrial control networks. However, its ability to guarantee the QoS requirements of such networks in the presence of adversarial flow still needs to be investigated. Queueing theory and particularly queueing network models have long been employed to study the performance and QoS characteristics of networks. It involves the use of well-established results from queueing theory to study the interaction between the forwarding plane switches and the control plane controllers. Although queueing network models have been used to study the behaviour of different application of SDN architectures, none of the existing works have considered the strong periodic network traffic in software-defined industrial control networks. In this article, we use queueing theoretical approach to develop a queueing network model for software-defined industrial control networks, and considered the strong periodic patterns of the network traffic in the data plane. We derive the performance measures for the analytical model using recent results from queueing theory and then apply one of the models proposed in Article 1 to study the effect of adversarial flow in software-defined industrial control networks. The contributions of this article can be summarized as:

i. Development of a queueing network model for software-defined industrial control networks, taking into account the strong periodic patterns of the network traffic in the data plane.

ii. Derivation of the performance measures for the analytical model using recent results from queueing theory.

iii. Modelling the effect of adversarial flow in software-defined industrial control networks using the developed queueing network model in combination with one of the models proposed in Article 1.

iv. Evaluation of the model to study the effectiveness of adding an adversarial flow in breaching QoS parameters.

C. *Adversary Model for Attacks Against IEC 61850 Real-Time Communication Protocols*

In this article, we utilize process calculus as the formal method to develop an adversary model for attacks against IEC 61850 communication protocols. Adversarial models are well-established for cryptographic protocols, but distributed real-time protocols have requirements that these abstractions are not intended to cover. The IEC 61850 standard for communication networks and systems for power utility automation in particular not only requires distributed processing, but in case of the generic object-oriented substation events and sampled value (GOOSE/SV) protocols also hard real-time characteristics. This motivates the desire to include both quality of service (QoS) and explicit network topology in an adversary model based on a π-calculus process algebraic formalism based on earlier work. This allows reasoning over process states, placement of adversarial entities and communication behaviour. The article then demonstrates the use of the proposed model for the simple case of a replay attack against the publish/subscribe GOOSE/SV subprotocol, showing bounds for nondetectability of such an attack. The contributions of this article can be summarized as:

i. Development of adversary model for attacks against IEC 61850 real-time communication protocols based on π-calculus process algebraic formalism.

ii. Formalization of IEC 61850 GOOSE messaging service using a π-calculus variant.

iii. Demonstration that the relative positions of the adversary in relation to the publisher, event notification service and subscriber determine the type of attacks that can be launched by an attacker.

iv. Illustration of the use of the developed model for the simple case of a reply attack against the publish/subscribe GOOSE/SV subprotocol, showing bounds for nondetectability of such an attack.

D. *A Review of Asset-Centric Threat Modelling Approaches*

This article presents a review of the existing asset-centric threat modelling approaches. The goal of this study is to identify gaps in the existing literature and to propose the use of formal methods to address the identified gaps. Threats are events that could cause harm to the confidentiality, integrity, or availability of information systems, through unauthorized disclosure, misuse, alteration, or destruction of information or information system. The process of developing and applying a representation of those threats, to understand the possibility of the threats being realized is referred to as threat modelling.

Threat modelling approaches provide defenders with a tool to characterize potential threats systematically. They include the prioritization of threats and mitigation based on probabilities of the threats being realized, the business impacts and the cost of countermeasures. In this article, we observe that the intuitive reasoning approach employed in the existing threat modelling approaches are not sufficient to threat model CPS where uncertainty, timing, and dependencies between the entities exist. The contributions of this article can be summarized as:

i. Analysis of the existing literature on asset-centric threat modelling approaches.

ii. Identification of the gaps in the existing literature and the fact that they are insufficient to threat model CPS where uncertainty, timing and dependencies between the entities exist.

E. *Threat Modelling of Cyber-Physical Systems Using an Applied π-Calculus*

This article employs an applied π-calculus as the formal method for threat modelling of cyber-physical systems (CPS). CPS are distributed systems in which the state of the physical system is generally not observable in non-trivial cases, and where state transitions of this physical system can also occur without resulting in immediate changes to observable variables. Threats to CPS from cyber-attacks are, however, often instantiable only where conditions on the CPS state during the attack meet certain conditions such that they drive the system state outside a desirable or safe space. In this article, an extension to an applied π-calculus in which we can capture both the behaviour of the CPS as well as modelling possible adversary behaviour is proposed. This is achieved by embedding an algebraic representation of Attack-Defence Trees (ADT) in the applied π-calculus and augmenting this by the addition of a partial ordering over the constituents of the ADT within the embedding. The modelling approach is illustrated for the case of an electrical substation fragment in which components communicate via the IEC 61850 protocol. The contributions of this article can be summarized as:

i. Extension to an applied π-calculus in which we can capture both the behaviour of the CPS as well as modelling possible adversary behaviour.

ii. Using the ADT and extending it with partial ordering of events to show causality relationship. This allows for the representation of not only the necessary conditions for an attack to be successful but also the sequencing of events or the way events need to be order for the attack to be successful.

iii. Translation of the ADT with partial ordering into the applied π-calculus using the message synchronization primitives for partial ordering, which enables us to make an argument for equivalence.

iv. Illustration of the proposed modelling approach for the case of an electrical substation fragment in which components communicate via the IEC 61850 protocol.

F. *Corrigendum to Threat Modelling of Cyber-Physical Systems Using an Applied π-Calculus*

This article is an addendum to Threat Modelling of Cyber-Physical Systems Using an Applied π-Calculus, where several additional and related contributions to the applied calculus are noted. It replaces the first, the second and the fourth paragraphs of Section 5.1, page 5 of the original article.

G. *A process algebraic approach to modelling cyber-physical systems security*

This article expands on the formal methods developed in Article 3 and 4 to provide a finer grain model of the actual interactions inside the physical systems of cyber-physical systems (CPS). A CPS has unique properties - for example, timing and state dependencies - which make its modelling and analysis difficult to achieve using the existing methods. This calls for novel approaches for the modelling and analysis of a CPS. In this article a formal method for the modelling and analysis of a CPS is presented. The approach employed allows for the description of the interaction between the cyber and physical system and the reasoning about scenarios where adversarial actions could occur and under what conditions such adversarial actions could have adverse effects. A case study of on-load tap-changer control mechanism modelled as a CPS is used to show the effectiveness of the proposed method. The contributions of this article can be summarized as:

i. Illustration of how a π-calculus model can be adapted so as to provide a good representation of a CPS.

ii. Provision of a finer grain model of the actual interactions inside the physical system, which allows for the description of the influence of the variables within the physical system.

iii. Investigation of situations where adversarial actions could occur in the CPS and under what conditions such adversarial actions could have adverse effects.

iv. Demonstration of the applicability of the proposed model using on-load tap change control (OLTC) mechanism and the possible physical attacks against the OLTC control mechanism.

1.8 Recommendations for Future Work

A possible extension of this work within modelling CPS security using a queueing network model is to layer a particular CPS application domain on top of the queueing network model. For example, it is possible to consider the conditions under which adversarial actions in a power substation could breach the performance bounds specified by the relevant standards. This would require the development of a queueing network model which takes into account the set of requirements as described by the relevant standards such as IEC 61850. Another research direction is to investigate refinements of the probability density function of the regular traffic in CPS and to derive its performance metrics. This could then be referenced when the regular traffic is combined with adversarial traffic and would facilitate the study of adversarial influences in CPS.

Also, there are several directions for future research within modelling CPS security using process algebraic methods presented in this work. One direction is to investigate how to optimally represent the internal state of both the cyber system and the physical system of CPS using process algebraic methods. This is because there are a number of assumptions made in the existing methods about the internal state of both the cyber system and the physical system that can no longer be fully supported. For example, one of these assumptions is that the internal transitions of CPS are driven only by external influences. For this, process algebraic methods can be used to capture these internal transitions of CPS by modelling the internal dynamics of the physical systems. Another direction is to develop an automated tool for modelling CPS security using the theoretical foundations presented in this work. This will support the modelling of security issues of a much more complex CPS.

Lastly, an interesting research direction that may be considered for future studies is to investigate how to unify the queueing network models and process algebraic methods to obtain a further-reaching formal representation of attacks against CPS. This framework would combine the advantages of queueing theoretical approaches as a well-established field and the expressive power of process algebraic methods to provide a much more detailed model to investigate security issues in CPS.

1.9 Conclusions

This thesis advances the state of the art in modelling CPS security by using queueing theoretical and process algebraic methods as the formal methods for understanding the behaviour of CPS with emphasis on real-time communication and capturing adversarial actions. It studied the way that adversaries behave inside CPS and has tried to model selective aspects of

these adversarial behaviours. For a number of these types of attacks, the timing behaviour and particularly the buffering capacity of the communication channel is critical. To address this problem, it employed queueing networks because they are appropriate models for capturing this sort of error conditions that leads to breaching of real-time constraints. It extended the existing models to include ways of representing the interaction of the different types of traffic and to understand how that can breach the quality of service requirements. This is because the probability density function for the adversarial flow need not be the same as that of the regular traffic. It then applied one of the proposed models to study the effect of adversarial flow in software-defined industrial control networks.

Furthermore, there are scenarios where it is necessary to study the internal state for the internal behaviour of CPS, which means this cannot be captured using a queueing network type of model. The main interest in this regard is to describe the internal behaviour of CPS in such a way that the asynchronous communication takes a prominent role. Hence, instead of having a model that relies on state machine, where we have global knowledge of the state; this thesis employed process calculus model which effectively allows us to look at the distributed state and how one state can influence another process including the uncertainty of the distributed state. It developed an adversary model based on the process calculus model which allows reasoning over process states, placement of adversarial entities and communication behaviour. Then, it extended the process calculus model by embedding an algebraic representation of Attack-Defence Trees (ADT) and augmenting this by the addition of a partial ordering over the constituents of the ADT within the embedding, offering an elegant mechanism to extend ADT to ordering and time-related attacks. And it expanded on the process calculus model to provide a finer grain model of the actual interactions inside the physical systems of CPS.

1.10 Bibliography

[1] ABADI, M., AND FOURNET, C. Mobile values, new names, and secure communication. *ACM SIGPLAN Notices 36*, 3 (mar 2001), 104–115. 9, 122, 134

[2] ABADI, M., AND GORDON, A. D. A Calculus for Cryptographic Protocols: The Spi Calculus. *Information and Computation 148*, 1 (jan 1999), 1–70. 9, 122, 148

[3] ADEPU, S., AND MATHUR, A. An Investigation into the Response of a Water Treatment System to Cyber Attacks. In *2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE)* (jan 2016), IEEE. 18, 125

[4] ADEPU, S., AND MATHUR, A. Generalized attacker and attack models for cyber physical systems. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)* (2016), vol. 1, IEEE, pp. 283–292. 16, 18, 84, 125

[5] ADEPU, S., AND MATHUR, A. From Design to Invariants: Detecting Attacks on Cyber Physical Systems. In *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (jul 2017), IEEE. 11

[6] ALMOHRI, H., CHENG, L., YAO, D., AND ALEMZADEH, H. On threat modeling and mitigation of medical cyber-physical systems. In *Proc. Systems and Engineering Technologies (CHASE) 2017 IEEE/ACM Int. Conf. Connected Health: Applications* (July 2017), pp. 114–119. 18, 125

[7] ALUR, R. Timed Automata. In *Computer Aided Verification*. Springer Berlin Heidelberg, 1999, pp. 8–22. 11

[8] AMIN, S., CÁRDENAS, A. A., AND SASTRY, S. S. Safe and Secure Networked Control Systems under Denial-of-Service Attacks. In *Hybrid Systems: Computation and Control*. Springer Berlin Heidelberg, 2009, pp. 31–45. 7, 17, 124

[9] AMIN, S., LITRICO, X., SASTRY, S., AND BAYEN, A. M. Cyber Security of Water SCADA Systems—Part I: Analysis and Experimentation of Stealthy Deception Attacks. *IEEE Transactions on Control Systems Technology 21*, 5 (sep 2013), 1963–1970. 7

[10] ASHIBANI, Y., AND MAHMOUD, Q. H. Cyber physical systems security: Analysis, challenges and solutions. *Computers & Security 68* (jul 2017), 81–97. 2

[11] ATIF, Y., JIANG, Y., JIANGUO, D., JEUSFELD, M., LINDSTRÖM, B., ANDLER, S., BRAX, C., HAGLUND, D., AND LINDSTRÖM, B. Cyber-threat analysis for cyber-physical systems, 2018. 18, 113, 125

[12] BAETEN, J. A brief history of process algebra. *Theoretical Computer Science 335*, 2-3 (may 2005), 131–146. 9

[13] BAHETI, R., AND GILL, H. Cyber-physical systems. *The impact of control technology 12*, 1 (2011), 161–166. 1

[14] BILEN, T., AYVAZ, K., AND CANBERK, B. QoS-based distributed flow management in software defined ultra-dense networks. *Ad Hoc Networks 78* (2018), 24–31. 14, 52, 63, 64

[15] BURMESTER, M., MAGKOS, E., AND CHRISSIKOPOULOS, V. Modeling security in cyber–physical systems. *International Journal of Critical Infrastructure Protection 5*, 3-4 (dec 2012), 118–126. 11

[16] CARDENAS, A. CYBER-PHYSICAL SYSTEMS SECURITYKNOWLEDGE AREA, 2019. 6

[17] CHOWDHARY, A., AND HUANG, D. SDN based Network Function Parallelism in Cloud. In *Proc. Networking and Communications (ICNC) 2019 Int. Conf. Computing* (Feb. 2019), pp. 486–490. 14, 63, 64

[18] CISOMAG. Ransomware Blocks U.S. Natural Gas Pipeline Supplies, 2020. Available from: `https://cisomag.eccouncil.org/ransomware-blocks-u-s-natural-gas-pipeline-supplies/`. 6

[19] DOLEV, D., AND YAO, A. On the security of public key protocols. *IEEE Transactions on Information Theory 29*, 2 (Mar. 1983), 198–208. 16, 78, 84, 88, 137

[20] EATON, C., AND RAMKUMAR, A. Colonial pipeline shutdown: Is there a gas shortage and when will the pipeline be fixed?, 2021. Available from: `https://www.wsj.com/articles/colonial-pipeline-cyberattack-hack-11620668583`. 2

[21] FERNANDEZ, E. B. Threat Modeling in Cyber-Physical Systems. In *Proc. nd Intl Conf Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech) 2016 IEEE 14th Intl Conf Dependable, Autonomic and Secure Computing, 14th Intl Conf Pervasive Intelligence and Computing* (Aug. 2016), pp. 448–453. 18, 113, 121, 125

[22] OPEN NETWORKING FOUNDATION. SDN architecture. *Technical Report*, 2014. 13, 46, 63

[23] GARTNER. Gartner predicts 75cyber-physical security incidents by 2024, Sept. 2020. Available from: `https://www.gartner.com/en/newsroom/press-releases/`. 1, 2

[24] GÓES, R. M., KANG, E., KWONG, R., AND LAFORTUNE, S. Stealthy deception attacks for cyber-physical systems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)* (2017), IEEE, pp. 4224–4230. 16, 85

[25] GOUD, N. BlueScope Steel operations disrupted due to Ransomware Cyber Attack, 2020. Available from: `https://www.cybersecurity-insiders.com/`. 6

[26] GREENBERG, A. 'crash override': The malware that took down a power grid, 2017. Available from: https://www.wired.com/story/crash-override-malware/. 5, 7, 154

[27] GRIFFOR, E. R., GREER, C., WOLLMAN, D. A., AND BURNS, M. J. Framework for cyber-physical systems: volume 1, overview. Tech. rep., jun 2017. 5

[28] HALPERIN, D., HEYDT-BENJAMIN, T. S., RANSFORD, B., CLARK, S. S., DEFEND, B., MORGAN, W., FU, K., KOHNO, T., AND MAISEL, W. H. Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses. In *2008 IEEE Symposium on Security and Privacy (sp 2008)* (may 2008), IEEE. 7

[29] HOPE, A. Toll Group's Operations Shut Down by Yet Another Ransomware Attack, 2020. Available from: https://www.cpomagazine.com/cyber-security/. 6

[30] HUANG, Y.-L., CÁRDENAS, A. A., AMIN, S., LIN, Z.-S., TSAI, H.-Y., AND SASTRY, S. Understanding the physical and economic consequences of attacks on control systems. *International Journal of Critical Infrastructure Protection 2*, 3 (oct 2009), 73–83. 6

[31] HUMAYED, A., LIN, J., LI, F., AND LUO, B. Cyber-Physical Systems Security—A Survey. *IEEE Internet of Things Journal 4*, 6 (dec 2017), 1802–1831. 2

[32] JAKARIA, A. H. M., YANG, W., RASHIDI, B., FUNG, C., AND RAHMAN, M. A. VFence: A Defense against Distributed Denial of Service Attacks Using Network Function Virtualization. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)* (jun 2016), IEEE. 7

[33] JARSCHEL, M., OECHSNER, S., SCHLOSSER, D., PRIES, R., GOLL, S., AND TRAN-GIA, P. Modeling and performance evaluation of an OpenFlow architecture. In *Proceedings of the 23rd international teletraffic congress* (2011), International Teletraffic Congress, pp. 1–7. 14, 49, 64, 67, 72

[34] JAVED, U., IQBAL, A., SALEH, S., HAIDER, S. A., AND ILYAS, M. U. A stochastic model for transit latency in OpenFlow SDNs. *Computer Networks 113* (2017), 218–229. 14, 53, 56

[35] KHAN, R., MCLAUGHLIN, K., LAVERTY, D., AND SEZER, S. STRIDE-based threat modeling for cyber-physical systems. In *Proc. IEEE PES Innovative Smart Grid Technologies Conf. Europe (ISGT-Europe)* (Sept. 2017), pp. 1–6. 18, 125

[36] KOTHARI, C. R. *Research methodology: Methods and techniques*. New Age International, 2004. 19

[37] KROTOFIL, M., CÁRDENAS, A. A., LARSEN, J., AND GOLLMANN, D. Vulnerabilities of cyber-physical systems to stale data - Determining the optimal time to launch attacks. *Int. J. Crit. Infrastructure Prot. 7*, 4 (2014), 213–232. 7, 126

[38] LANOTTE, R., MERRO, M., MUNTEANU, A., AND VIGANÒ, L. A Formal Approach to Physics-based Attacks in Cyber-physical Systems. *ACM Trans. Priv. Secur. 23*, 1 (2020), 3:1–3:41. 15, 16, 19, 148, 149, 150, 155, 157, 161, 171

[39] LANOTTE, R., MERRO, M., MURADORE, R., AND VIGANÒ, L. A formal approach to cyber-physical attacks. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)* (2017), IEEE, pp. 436–450. 15, 16, 17, 18, 19, 85, 122, 126, 129, 148, 149, 150, 155, 157

[40] LEE, E. The Past, Present and Future of Cyber-Physical Systems: A Focus on Models. *Sensors 15*, 3 (feb 2015), 4837–4869. 2, 3, 4

[41] LEE, E. A. Cyber Physical Systems: Design Challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)* (may 2008), IEEE. 2, 5

[42] LI, T., ZHOU, H., LUO, H., QUAN, W., AND YU, S. Modeling software defined satellite networks using queueing theory. In *Proc. IEEE Int. Conf. Communications (ICC)* (May 2017), pp. 1–6. 14, 63, 64, 65

[43] LYGEROS, J. AND PRANDINI, M. Stochastic Hybrid Systems: A Powerful Framework for Complex, Large Scale Applications. *European Journal of Control 16*, 6 Elsevier BV (2010), 583–594. 11

[44] MARQUARDT, A., LEVENSON, E., AND TAL, A. Florida water treatment facility hack used a dormant remote access software, sheriff says, 2021. Available from: `https://edition.cnn.com/2021/02/10/us/florida-water-poison-cyber/index.html`. 6

[45] MARTINS, G., BHATIA, S., KOUTSOUKOS, X., STOUFFER, K., TANG, C., AND CANDELL, R. Towards a systematic threat modeling approach for cyber-physical systems. In *Proc. Resilience Week (RWS)* (Aug. 2015), pp. 1–6. 17, 125

[46] MCEVOY, T. R., AND WOLTHUSEN, S. D. A formal adversary capability model for SCADA environments. In *International Workshop on Critical Information Infrastructures Security* (2010), Springer, pp. 93–103. 2, 8, 16, 79, 84

[47] MCEVOY, T. AND WOLTHUSEN, S. A Plant-Wide Industrial Process Control Security Problem. *Critical Infrastructure Protection V*, Springer Berlin Heidelberg, 2011, 47-56 2, 8, 156

[48] MCEVOY, T. R. AND WOLTHUSEN, S. D. Algebraic Analysis of Attack Impacts and Countermeasures in Critical Infrastructures. *Critical Information Infrastructures Security*, Springer Berlin Heidelberg, 2013, 168-179 2, 8, 156

[49] MCEVOY, T. AND WOLTHUSEN, S. Agent Interaction and State Determination in SCADA Systems. *Critical Infrastructure Protection VI*, Springer Berlin Heidelberg, 2012, 99-109 2, 8, 156

[50] MCEVOY, T. R. AND WOLTHUSEN, S. D. A Formal Adversary Capability Model for SCADA Environments. *Critical Information Infrastructures Security*, Springer Berlin Heidelberg, 2011, 93-103 2, 8, 157

[51] MCLAUGHLIN, S. CPS: stateful policy enforcement for control system device usage. In *Proceedings of the 29th Annual Computer Security Applications Conference* (dec 2013), ACM. 7

[52] MIAO, W., MIN, G., WU, Y., WANG, H., AND HU, J. Performance modelling and analysis of software-defined networking under bursty multimedia traffic. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 12*, 5s (2016), 77. 15, 50, 63, 65, 66, 69

[53] MILNER, R., PARROW, J., AND WALKER, D. A calculus of mobile processes, I. *Information and Computation 100*, 1 (sep 1992), 1–40. 9, 122

[54] MO, Y., GARONE, E., CASAVOLA, A., AND SINOPOLI, B. False data injection attacks against state estimation in wireless sensor networks. In *49th IEEE Conference on Decision and Control (CDC)* (dec 2010), IEEE. 17, 124

[55] MO, Y.; CHABUKSWAR, R. AND SINOPOLI, B. Detecting Integrity Attacks on SCADA Systems. *IEEE Transactions on Control Systems Technology,*, IEEE, 2014, 22, 1396-1407 17, 124

[56] MUNTEANU, A.; PASQUA, M. AND MERRO, M. Impact Analysis of Cyber-Physical Attacks on a Water Tank System via Statistical Model Checking. In *Proceedings of the 8th International Conference on Formal Methods in Software Engineering* (Sep 2020), ACM. 12

[57] NICOLA, R. A gentle introduction to Process Algebras, 2013. Available from: `https://www.pst.ifi.lmu.de/Lehre/fruhere-semester/sose-2013/`

formale-spezifikation-und-verifikation/intro-to-pa.pdf. 9

[58] NIU, L., AND CLARK, A. Secure Control Under Linear Temporal Logic Constraints. In *2018 Annual American Control Conference (ACC)* (jun 2018), IEEE. 12

[59] NWEKE, L. O., AND WOLTHUSEN, S. D. Resilience Analysis of Software-Defined Networks Using Queueing Networks. In *2020 International Conference on Computing, Networking and Communications (ICNC)* (feb 2020), IEEE. 14, 63

[60] OROJLOO, H., AND AZGOMI, M. A. A method for modeling and evaluation of the security of cyber-physical systems. In *2014 11th International ISC Conference on Information Security and Cryptology* (2014), IEEE, pp. 131–136. 16, 84

[61] PARROW, J. An Introduction to the pi-Calculus. In *Handbook of Process Algebra*. Elsevier, 2001, pp. 479–543. 11, 123

[62] PASQUALETTI, F., DORFLER, F., AND BULLO, F. Attack Detection and Identification in Cyber-Physical Systems. *IEEE Transactions on Automatic Control 58*, 11 (nov 2013), 2715–2729. 17, 124

[63] PINSKY, M., AND KARLIN, S. *An introduction to stochastic modeling*. Academic press, 2010. 8, 48

[64] PNUELI, A. The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)* (sep 1977), IEEE. 12

[65] RAJKUMAR, R., LEE, I., SHA, L., AND STANKOVIC, J. A. Cyber-physical systems: the next computing revolution. In *Proceedings of the 47th Design Automation Conference, DAC 2010, Anaheim, California, USA, July 13-18, 2010* (2010), S. S. Sapatnekar, Ed., ACM, pp. 731–736. 4

[66] REKIK, M., GRANSART, C., AND BERBINEAU, M. Cyber-physical Threats and Vulnerabilities Analysis for Train Control and Monitoring Systems. In *Proc. Computers and Communications (ISNCC) 2018 Int. Symp. Networks* (June 2018), pp. 1–6. 18, 113, 125

[67] RING, M., SCHLÖR, D., LANDES, D. AND HOTHO, A. Flow-based network traffic generation using Generative Adversarial Networks. In *Computer and Security*, Vol. 82 Elsevier, 2019, pp. 156–172. 15

[68] ROCCHETTO, M., AND TIPPENHAUER, N. O. On Attacker Models and Profiles for Cyber-Physical Systems. In *Computer Security – ESORICS 2016*. Springer International Publishing, 2016, pp. 427–449. 18, 125

[69] SÁNCHEZ, H. S., ROTONDO, D., ESCOBET, T., PUIG, V., SALUDES, J., AND QUEVEDO, J. Detection of replay attacks in cyber-physical systems using a frequency-based signature. *Journal of the Franklin Institute 356*, 5 (mar 2019), 2798–2824. 17, 124

[70] SANGER, D. E., KRAUSS, C., AND PERLROTH, N. Cyberattack forces a shutdown of a top u.s. pipeline, 2021. Available from: `https://www.nytimes.com/2021/05/08/us/politics/cyberattack-colonial-pipeline.html`. 1

[71] SANGIORGI, D., AND WALKER, D. *The pi-calculus: a Theory of Mobile Processes*. Cambridge university press, 2003. 9, 82, 93, 122, 140, 159

[72] SHRIVASTAVA, A., DERLER, P., LI-BABOUD, Y., STANTON, K. B., KHAYATIAN, M., ANDRADE, H. A., WEISS, M., EIDSON, J. C., AND CHANDHOKE, S. Time in cyber-physical systems. In *Proceedings of the Eleventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, CODES 2016, Pittsburgh, Pennsylvania, USA, October 1-7, 2016* (2016), ACM, pp. 4:1–4:10. 5

[73] SINGH, D., NG, B., LAI, Y., LIN, Y., AND SEAH, W. K. G. Modelling Software-Defined Networking: Switch Design with Finite Buffer and Priority Queueing. In *Proc. IEEE 42nd Conf. Local Computer Networks (LCN)* (Oct. 2017), pp. 567–570. 14, 51, 56, 64

[74] SINGH, D., NG, B., LAI, Y.-C., LIN, Y.-D., AND SEAH, W. K. Modelling Software-Defined Networking: Software and hardware switches. *Journal of Network and Computer Applications 122* (2018), 24–36. 14, 51, 56, 64, 67

[75] SMITH, J. M. *Introduction to Queueing Networks*. Springer, Cham, 2018. 8, 9, 13, 48

[76] SMITH, T. Hacker jailed for revenge sewage attacks, 2001. Available from: `https://www.theregister.com/2001/10/31/hacker_jailed_for_revenge_sewage/`. 5

[77] SOOD, K., YU, S., AND XIANG, Y. Performance Analysis of Software-Defined Network Switch Using $M/Geo/1$ Model. *IEEE Communications Letters 20*, 12 (Dec. 2016), 2522–2525. 14, 52, 64, 67

[78] TEIXEIRA, A., SHAMES, I., SANDBERG, H., AND JOHANSSON, K. H. Revealing stealthy attacks in control systems. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)* (oct 2012), IEEE. 7

[79] TIAN, W. Analysis and efficient provisioning of access networks with correlated and bursty arrivals. *Int. J. Communication Systems 27*, 4 (2014), 551–570. 15, 21, 65, 67, 69

[80] VIGO, R. The cyber-physical attacker. In *International Conference on Computer Safety, Reliability, and Security* (2012), Springer, pp. 347–356. 16, 84

[81] XIONG, B., YANG, K., ZHAO, J., LI, W., AND LI, K. Performance evaluation of OpenFlow-based software-defined networks based on queueing model. *Computer Networks 102* (2016), 172–185. 14, 52, 56, 67

[82] YEN, T., AND SU, C. An SDN-based cloud computing architecture and its mathematical model. In *Proc. Electronics and Electrical Engineering 2014 Int. Conf. Information Science* (Apr. 2014), vol. 3, pp. 1728–1731. 14, 49, 63, 64

[83] YU, M., HE, T., MCDANIEL, P., AND BURKE, Q. K. Flow Table Security in SDN: Adversarial Reconnaissance and Intelligent Attacks. In *Proc. IEEE INFOCOM 2020 - IEEE Conference on Computer Communications* (Aug. 2020) 15

[84] ZALEWSKI, J., DRAGER, S., MCKEEVER, W., AND KORNECKI, A. J. Threat modeling for security assessment in cyberphysical systems. In *Cyber Security and Information Intelligence, CSIIRW '13, Oak Ridge, TN, USA, January 8-10, 2013* (2013), F. T. Sheldon, A. Giani, A. W. Krings, and R. K. Abercrombie, Eds., ACM, p. 10. 17, 124

[85] ZETTER, K. An unprecedented look at stuxnet, the world's first digital weapon, Nov. 2014. Available from: `https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet/`. 5, 118

Chapter 2

Resilience Analysis of Software-Defined Networks Using Queueing Networks

Resilience Analysis of Software-Defined Networks Using Queueing Networks

2020 International Conference on Computing, Networking and Communications (ICNC), Big Island, HI, USA

Livinus Obiora Nweke and Stephen D. Wolthusen

Abstract

Software-Defined Networks (SDN) are being adopted widely and are also likely to be deployed as the infrastructure of systems with critical real-time properties such as Industrial Control Systems (ICS). This raises the question of what security and performance guarantees can be given for the data plane of such critical systems and whether any control plane actions will adversely affect these guarantees, particularly for quality of service in real-time systems. In this paper we study the existing literature on the analysis of SDN using queueing networks and show ways in which models need to be extended to study attacks that are based on arrival rates and service time distributions of flows in SDN.

2.1 Introduction

Networks with hard and firm real-time constraints such as industrial control systems networks historically relied upon dedicated field bus systems with limited performance but well-defined, deterministic characteristics. In recent years, cost and performance considerations have encouraged migration to standard, if heavily over-provisioned and somewhat modified, networks such as Industrial Ethernet variants (e.g. PROFINET IRT or EtherCAT). As network infrastructures outside this domain increasingly rely on the advantages offered by software-defined networks (SDN) and network function virtualisation (NFV), it is likely that this will also need to be considered for real-time environments.

However, the more flexible and dynamic SDN architecture not only offers benefits, but also raises questions on the ability to satisfy security and quality of service (QoS) guarantees, particularly in the presence of malfunctioning or malicious entities interfering with control or data plane.

The purpose of this paper is to analyse the existing literature on the analysis of SDN using queueing networks and to show how the models can be extended to study attacks that are based on arrival rates and service time distributions of flows in SDN. We only consider attacks that can be analysed at the flow abstraction level, without considering the semantics of the flows, a limitation that is inherent in using queueing networks. Although simulations and experimentation are desirable for the study of SDN, analytical modelling permits the study of a wider range of configurations and parameters as well as the optimisation not only of performance [1], but also an understanding of the severity of attacks.

Understanding the performance of SDN through its analysis using queuing networks is of particular importance for systems with strong QoS requirements. These requirements include particularly the delay, loss, and jitter parameters and respective requirements for different types and classes of service [2]. Systems that will fail if the QoS requirements are not met are referred to as hard real-time systems [3]. Although it is easy to apply queuing networks to the general performance analysis of most SDN architectures, systems with hard real-time requirements must be more comprehensive to capture all relevant interactions. Many of the QoS requirements in hard real-time systems, moreover, are immediately usable for security considerations since the feasibility and effort required by adversaries for denial of service (DoS) attacks, both immediately and transitively, are a particular concern.

We therefore provide a review on the analysis of SDN using queueing networks in this paper, analysing the still nascent body of work currently emerging in this domain. We note that thus far the focus is on relatively straightforward models such as the M/M/1 model (using Kendall's notation) being used as the queueing model in characterising the SDN behaviour, with M/G/1 and GI/M/1/K models considered more appropriate model for SDN controllers and switches, respectively. This appears to be a lacuna in existing work as a more precise characterisation of both arrival processes and service time distributions for regular operation, configuration changes, and adversarial action is thus far not being considered. This, however, is critical to understand vulnerabilities and security requirements.

The rest of this paper is organised as follows. In section 2.2, basic concepts and terminologies use in this paper are discussed. A literature review of the works on analysis of SDN using queueing networks is presented in section 2.3. Section 2.4 presents our proposed models and metrics for security analysis of SDN using queueing networks. Section 2.5 provides an illustration using DoS attack, on how the models proposed may be applied to study attacks that are based on arrival rates and service time distributions of the packet flows in SDN. Section 2.6 offers a brief discussion on the results from the survey and the lessons learned. Finally, section 2.7 concludes the paper and presents future works.

2.2 Background

This section discusses the basic concepts and terminologies used in this survey. In particular, SDN and queueing theory are contextualised.

2.2.1 Software-Defined Networks (SDN)

SDN is a current network architecture in seeking to decouple the control plane from the data plane. Unlike in a traditional network architecture where control and data plane are embedded in the networking devices, SDN separates roles such that networking devices can become purely forwarding devices with the forwarding instructions pushed to them via the control plane and allowing the use of commodity components [4]. The main goals of the SDN architecture were to simplify the deployment of control plane functions and to enable the applications to deal with a single abstracted network device without concern for the implementation details whilst lowering dependency on dedicated components [5]. Thus, with SDN, network control functions become directly programmable enabling the automation of network functions which in turn facilitate the building of highly scalable and flexible networks that can readily adapt to the dynamic nature of today's environment [6].

The SDN architecture sketched in figure 2.1, comprises of the application layer, control layer, and infrastructure layer [4]. The SDN applications exist in the application layer, and interact with the control layer via the northbound interfaces. In the middle of the SDN architecture is the SDN controller, which translates applications' requirements and exerts low-level control over the network elements, while providing relevant information up to the SDN applications. The infrastructure layer comprises of the network elements, which expose their capabilities toward the control layer via the southbound interfaces. It can be inferred from this setup that the network intelligence is logically centralised in the SDN controllers which maintain a global view of the network [4]. Therefore, the network appears to the applications and policy engines as a single, logical switch; and the network devices no longer need to understand and process several protocols but merely accept instructions from the SDN controllers [6].

What is still not visible from the illustration above is that a reasonable large SDN architecture will have multiple controllers and several number of switches. Also, the architecture does not show the information flows and communication between these components. Further, the realisation of the concept of SDN entails that two requirements must be met: the need for a common logical architecture in the network devices to be managed by the SDN controller and the need for a standard, secure protocol between the SDN controller and the network devices (and further on to the application layer) [7]. These requirements are addressed e.g. by OpenFlow [8], which

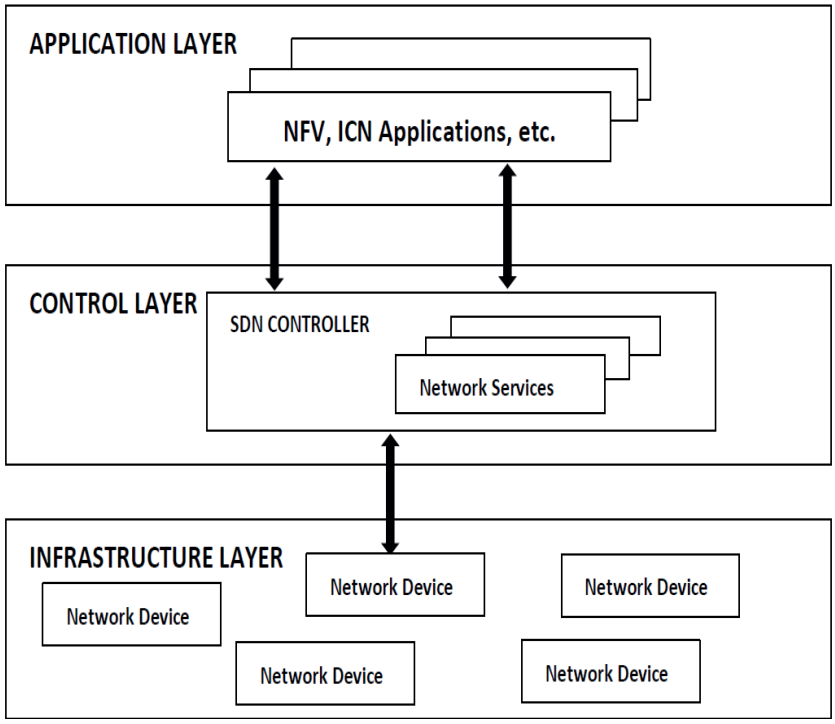


Figure 2.1: SDN Architecture

is both a protocol between SDN controllers and network devices, as well as a specification of the logical structure of the network switch function. The protocol specifies how the applications can directly access and manipulate the network devices via the SDN controller without regards to the details of how the switch are implemented, and also it uses the concept of flows to identify network traffic based on pre-defined match rules that can be statically or dynamically programmed [8].

The SDN architecture and OpenFlow standard provide an open architecture in which control functions are separated from the network devices and placed in a logically centralised controller. The centralisation of the network state in the control plane provides the flexibility to configure, manage, secure, and optimise network resources with far-reaching automation. They also enable the underlying infrastructure to be abstracted for applications and network services, enabling the network to be treated as a logical entity [6].

2.2.2 Queueing Theory

Queueing theory started by the study of queues, devising analytical mechanisms and tools for the design and evaluation of the performance of queueing systems [9]. To characterise a queueing system, there is need to identify the probabilistic properties of the arrival processes, service times and service disciplines [9]. Conventionally, the arrival process is characterised by the distribution of the inter-arrival times of the customers. These inter-arrival times are usually assumed to be independent and identically distributed random variables. They are denoted by A(t):

$$A(t) = P(\textit{interarrival time} < t).$$

The service time is used to express how long the service will take. It is usually assumed that the service time for a customer is independent and does not depend upon the arrival process. Another common assumption about service time is that it is exponentially distributed. Its distribution function is denoted by B(x):

$$B(x) = P(\textit{service time} < x).$$

The service discipline is used to decide how the next customer waiting on the queue is served. The most common methods used include: First-in, First-out (FIFO); Last-come, First-out (LIFO); Random Service (RS); Priority.

Kendall's notation is the standard notation used to describe and classify queuing systems [9]. It is given as:

$$A/B/m/K/n/D,$$

where; A is the distribution function of the inter-arrival times, B is the distribution function of the service times, m is the number of servers, K is the capacity of the system, n is the population size, and D is the service discipline.

It is common practice to denote exponentially distributed random variables by M meaning Markovain or memoryless [10]. If the population size and the capacity are infinite, the service discipline is FIFO and is usually omitted. Thus, M/M/1 denotes a system with Poisson arrivals, exponentially distributed service times and a single server [9]. Basic queueing systems are used to describe the system as a unique resource, but to describe the system as a set of interacting resources; queueing networks are usually used. A queueing network can be defined as a collection of service centres representing the system resources which are used provide service to a collection of customers that represent the users [9]. Queueing networks have been shown to be appropriate tool for system performance evaluation and have also been shown as being helpful in the modelling of attacks against distributed systems.

2.3 State-of-art in Analysis of SDN using Queueing Networks

Whilst queue network research spans decades, here we concentrate on queueing distributions for modelling software-defined networks as a method for characterisation and categorisation as this is highly pertinent to the adaptation for security modelling

2.3.1 Modelling Based on M/M/1 Distribution

The M/M/1 distribution consists of a Poission arrival, one exponential server, infinite FIFO queue and unlimited customer population. Although, these are very strong assumptions not satisfied by real systems, they provide useful insights which may be used to study how real systems will perform given certain parameters. The first known analysis of SDN using queueing model, presented by Jarschel et al. in [11] is based on M/M/1 distribution. The model assumed that packet arrivals form Poisson streams and that the queue length of the controller system is finite so as to model the possibility of dropped packets under high load condition. In addition, the model has some limitations in that it does not capture the fact that packets arriving at the switch are queued one queue per port and that it is limited to a single switch per controller. The results obtained from the analytical model were validated using a packet based simulation in OMNeT++.

Some of the limitations in [11] were subsequently addressed by Chilwan, Mahmood, Østerbø and Jarschel in [12]. It was achieved by modelling both the controller and the switch as Jackson Network, with some modifications to suit OpenFlow-based SDN. Whilst maintaining the assumptions made in the previous paper, the authors also showed that the model can be extended to handle the case where more than one switch exist in the data plane. The model was then validated using a discrete event simulation which resembles the queuing behaviour of the proposed model. In addition, the model was further extended by Mahmood, Chilwan, Østerbø and Jarschel in [13] for the performance analysis of OpenFlow-based SDN with multiple nodes in the data plane.

Yen and Su in [14] proposed SDN based cloud computing architecture which was implemented using open source Open vSwitch and POX controller packages. The queueing network model based on M/M/1 was used to model the operation of the OpenFlow architecture to show the correctness of the architecture. In this model, it was assumed that the arrival rate of cloud service request is an exponential distribution and that the service rate of both the Open vSwitch and the POX controller was exponentially distributed. Considering that the average packet queue length is an important parameter of the SDN-based cloud environment, it was used as the performance metric for the evaluation of the architecture. The numerical results

of the average queue length presented in the work show that the proposed SDN-based cloud computing architecture can provide QoS guarantees for cloud services.

A Preemption-based Packet Scheduling (P^2S) scheme to improve global fairness and reduce packet loss rate in SDN data plane was presented by Miao, Min, Wu, and Wang in [15]. The performance of the proposed scheme was analysed using queueing networks based on M/M/1 distribution. In this model, the arrival and departure of packets in the system were characterised as a birth-death process. Miao et al. assumed that the packet arrivals followed a Poisson distribution, and that the service time follows a negative exponential distribution. The performance of the P^2S scheme was compared to the traditional FIFO scheme in terms of packet loss probability and service fairness. The results presented in the work showed that FIFO cannot guarantee fairness among packets and suffers from high packet loss, while P^2S scheme offers priority for the packets from the controller and as such, achieves better performance in terms of global fairness and packet loss probability. Also, the performance of the model was validated using simulations by varying the traffic arrival rate, flow table hit probability and the service rates of the switch and controller.

Fahmin, Lai, Hossain, Lin and Saha in [16] presented the performance modeling of SDN with network virtualisation function (NFV) under or aside the controller, which are the different methods of combining SDN with NFV; using M/M/1 queueing model. Fahmin et al. used queueing theory to develop mathematical models for both scenarios resulting to queueing networks which were then analysed. In this model, they assumed that the arrival process at the switch follows Poisson process, the service time of packets in switch, controller and virtualised network function (VNF) follow exponential distribution, and the queue size of a switch, controller and VNF is infinite. The average packet delay was used as the performance metric for the analysis of both approaches. Simulation was used for the validation of the analytical process. Also, the results presented showed that the packet delay for SDN with NFV aside the controller is significantly less than that for SDN with NFV under the controller, and that the service rate at VNF does not affect the delay gap between SDN with NFV aside and under the controller.

2.3.2 Modelling Based on GI/M/1/K Distribution

The GI/M/1/K distribution is a type of queueing distribution where inter-arrival times are independent and identically distributed with general distribution, and service times are independent and exponentially distributed. Also, the system is made of a finite waiting space and the arriving customers are served on a FIFO basis. The properties of GI/M/1/K distribution is suitable for the study of packet arrivals at SDN switch. Goto, Masuyama, Ng,

Seah, and Takahashi in [1] proposed a queueing model of an OpenFlow-based SDN that takes into account classful treatment of packets arriving at a switch. They argued that the different packets arriving at the switch should be treated differently and to that end, they classified these packets as follows: external packets arriving at the switch according to a Poisson process, Class S; packet whose forwarding information is missing in the flow table and are forwarded to the controller, Class C; and packets processed by the controller and sent back to the switch, Class F. The switch was then model using GI/M/1/K distribution that can enqueue no more than K1 packets. In this model, three performance measures, namely, packet loss probabilities in Class S and F, and average packet transfer delay through the system was used in the analysis. Simulation was used in the validation of the model and the simulation results presented matches the average delay obtained by the queueing analysis and thus confirming the validity of the model.

Queueing model was used by Singh, Ng, Lai, Lin and Seah in [17] to investigate the effect of a buffer sharing in SDN switch. Using GI/M/1/K distribution for modelling the switch, they proposed two models: Shared Buffer referred to as SE Model; and Priority Queueing Buffer referred to as SPE Model. In this work, Model SE uses a single queue for the switch while Model SPE uses priority queue for the switch. They assumed that the controller had an infinite capacity queue and that the external packet arrival at the switch follows Poisson process. The relative minimum capacity and relative time to install the flow table entries were used as the performance measures to compare the performance of the models. Discrete event simulation of the SE and SPE queuing networks was used in the validation of the analytical results.

Singh, Ng, Lai, Lin and Seah in [18], presented a unified queueing model for characterizing both the performance of hardware switches and software switches in SDN. They started by first modelling SDN with software switch and hardware switch, and then used queuing models to model the software (SPE) and hardware (HPE) SDN switches. They assumed that the controller had an infinite capacity queue and that the external arrival at the switch follow Poisson process. In this model, the switches were modelled with GI/M/1/K distribution, however, M/M/1/K distribution was added in modelling the hardware processor of the hardware switch. The average packet transfer delay was used as the performance metric for comparing the models. They validated the accuracy of the analytical results for both models using simulation. The results from the study show that a hardware switch performs better than a software switch in terms of average delay and packet loss probability.

2.3.3 Modelling Based on M^X/M/1 Distribution

The M^X/M/1 distribution is a type of M/M/1 distribution with batch arrivals of random size. The arrival stream forms a Poisson process and the batch size is a random variable. Xiong, Yang, Zhao, Li, and Li in [19] investigated the packet arrival process and forwarding procedure at an OpenFlow switch. They modelled its packet forwarding performance using M^X/M/1 distribution. In this work, they argued that packet arrivals at the switch does not follow Poisson process but instead results in packet batch arrivals. They assumed that the packet arrived at the switch as Poisson stream, the number of packets in a switch conforms to Poisson distribution, and the packet processing time of the switch conforms to negative exponential distribution. The average sojourn time and average queue length were used as the performance measures for evaluating the performance of the model. Also, Bilen, Ayvaz, and Canberk in [20] used M^X/M/1 distribution to model elephant flows. They proposed a distribution flow management model in SDN ultra-dense network based on queueing theory.

2.3.4 Modelling Based on M/Geo/1 Distribution

The M/Geo/1 distribution comprises of Poisson distribution and the service times obey geometric distribution. Sood, Yu, and Xiang in [21] proposed an analytical modelling based on M/Geo/1 distribution to study the performance of SDN switches. They assumed that the packet arrival at the switch followed Poisson process and the service times obey geometry distribution. In this model, they used the Embedded Markov Chain theory and applied it to M/Geo/1 queue to obtain the number of packets in the system and the service time. The model was used for just investigating the average response time of SDN switch without considering the switch-controller interaction. Thus, the average flow response time was used as the performance metric for evaluating the performance of the switch. Simulations were used for the validation of the model and both the simulation and analytical results presented matched each other. They concluded by noting that the important factors that determine the switch's mean response time are packet arrival rate, number of flow-table entries, and the position of the targeting rule by a corresponding packet.

2.3.5 Modelling Based on M/G/1 Distribution

In M/G/1 distribution, the arrival process is Poisson and the service time for each customer is generally distributed. Also, the distribution has an infinite queueing capacity and unlimited customer population. The packet-in message process of SDN controller was modelled by Xiong, Yang, Zhao, Li, and Li in [19]. They studied the arrival process and serving process of packet-in

messages in SDN controller, and modelled the SDN controller performance with the M/G/1 distribution. In this model, they assumed that the packet-in messages at the controller from its switches constituted a Poisson stream and that the processing time of packet-in messages in the controller conforms to normal distribution. The performance of the controller was evaluated using the publicly available benchmark Cbench and the sojourn time of a packet-in message was used as the performance metric. Also, they compared the performance of their model with the most common model (M/M/1) used in characterizing the performance of SDN controllers. The results presented showed that M/G/1 provided more accurate approximation of the SDN controller performance than M/M/1. In the same way, Javed, Iqbal, Saleh, Haider and Ilyas in [22] demonstrated through experiments that M/G/1 distribution using log-normal distribution mixture as the service distribution is closer to reality in terms of SDN controller performance evaluation than M/M/1 distribution.

2.4 Models and Metrics for Security Analysis of SDN Using Queueing Networks

It is clear that it is inappropriate to use the already-strong assumptions on distributions for regular network traffic discussed in section 2.3 for adversarial traffic. We therefore propose to expand the models such that they can capture with additional traffic caused by attacks, differentiating this traffic as necessary. Within the limits of not considering the semantics of the flows themselves, what is needed to be extended are primarily arrival rates and service time distributions as the models reviewed above are too simple to capture such adversarial behaviour. We hence propose two modelling levels: one based on aggregating the distribution where we are primarily interested in breaches of QoS guarantees, and a more refined model which explicitly captures SDN behaviour by studying queue network representations of the SDN architecture, and distinguishing between baseline and adversary flows *with different arrival rates and service time distributions*.

2.4.1 Modelling Based on Aggregating the Distribution

It is possible to modify the arrival process of the existing models to capture the situation where adversarial actions may occur. The proposed model is similar to work done in [23], where the input to the queueing network model is considered as a combination of both the normal traffic and malicious traffic. The addition of the two probability distribution functions is justified by the fact that the distribution of a malicious traffic will not be the same as the normal traffic because of the way the attacker would craft the malicious packet to evade detection. Hence, it is not plausible to use Poisson distri-

bution to describe the distribution of both the normal traffic and malicious traffic. Therefore, in order to capture the behaviour of an attacker, we need to as a minimum, use a different type of distribution to describe malicious traffic and then calculate the impact using the aggregate distribution as input to the queueing network model as illustrated in figure 2.2.

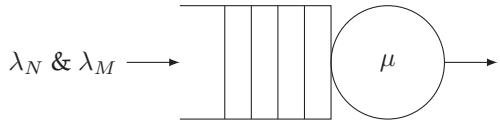


Figure 2.2: Proposed Model Based on Aggregating the Distribution

2.4.2 Modelling Based on Analysing the effects of the two Queues Separately

The effects of the two queues and distinct service times can be studied separately to understand adversarial actions. This can be achieved by distinguishing between regular flows and adversarial flows, which also cause flows internal to the SDN architecture primarily in the form of additional messages sent from controllers to switches where new or altered flows must be accommodated before actually considering the aggregate flows through the switching fabric. Once a new flow is established, the malicious packet can be treated as additional traffic as above, but aggregate service times become particularly interesting in establishing possible breaches of QoS requirements, minimizing the effort required on the part of adversaries.

2.5 Towards Security Analysis of SDN Using Queueing Networks

The previous section have identified areas of SDN security that are amenable to study using queueing networks. In this section, we use DoS attacks to illustrate how the models proposed may be applied to study attacks that are based on arrival rates and service time distributions of the packet flows in SDN. To the best of our knowledge, queueing network models and other analytical approaches have not been used in the literature to evaluate real-time processing security issues in SDN in particular. Although network calculus and queueing networks have been used in security research, the flexibility that comes with using well-established results from queueing theory justifies our choice for proposing the use of queueing networks for the study of DoS attacks in SDN, similar to the works done in other fields [24, 29, 26, 23, 27, 28].

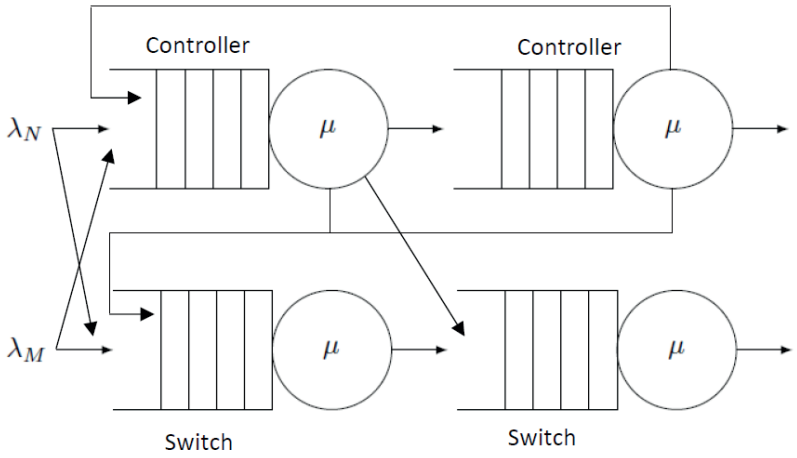


Figure 2.3: Proposed Model Based on Analysing the effects of the two Queues Separately

DoS attacks are among the security issues in SDN that can be captured easily using queueing networks. SDN can be characterized as queueing networks model to capture the interaction between its different components. Using the model based on aggregating the distribution proposed in the previous section, we can then explicitly model the two types of flows. The distribution for a typical DoS attacks is not a uniform distribution because of the way the attacker crafts the attacks. Then combining both the regular traffic distribution, which is usually assumed to be Poisson; and the adversarial distribution, we can then study the specific characteristics of the attacker.

Additional degree of flexibility would have to be introduced to the queueing networks model to capture the behaviour of SDN under DoS attacks. A typical approach that may be adopted from the queueing models presented in this paper is to use queueing distributions with finite capacity. The introduction of queues with finite capacity will facilitate the modelling of SDN under DoS attacks as was done in [29, 26, 27]. Hence, M/M/1/K or M/G/1/K may be used in characterising the behaviour of both the data plane and control plane. In using M/M/1/K or M/G/1/K for modelling the behaviour of both the data plane and control plane, blocking or loss probability (P_{loss}) may be deployed as the security metrics for studying the security properties of SDN. For example, if P_{loss} is large, SDN may be said to be under DoS attacks.

Also, the approach used in [29], which involves the use of a threshold value that is greater than 0, may be applied to provide information about the SDN security status. If P_{loss} is less than the threshold value, it can be

concluded that the SDN is not under DoS attacks. However, if P_{loss} is greater than or equal to the threshold value, implying that the SDN resources is exhausted; it can be concluded that the SDN is under DoS attacks.

Other types of security challenges that can be studied using queueing networks are described in [28, 24, 23]. In particular, the framework proposed in [24] and used in [23] could be used to study some security issues in SDN as the authors opined that the framework allows for the study of DoS and performance degradation, which can also be suffered by SDN. Therefore, queueing networks offer a promising approach to studying the security properties of SDN in order to provide the basis for it deployments in industrial environments where there are stringent security and QoS requirements.

2.6 Discussion

The goal of this study has been to evaluate the different methods in the literature employed for the analysis of SDN using queueing networks and to categorise them based on the queuing distribution deployed for the analysis as this is most pertinent for subsequent extension into studies of security properties. The categorisation showed that simple M/M/1 distribution is the most widely adopted model for characterizing the behaviour of SDN switch and controller. Although the distribution facilitates an easy analytical process, it seems to be a poor fit for evaluating the performance of SDN switch and controller. Thus, it is important that a more realistic approach for characterising the behaviour of SDN switch and controller is investigated.

Already, the authors in [13] suggested at the completion of their work with M/M/1 that M/G/1 is a more appropriate model for characterizing the behaviour of SDN controller that needs to be investigated. The authors in [19] went further to use M/G/1 to model the performance of SDN controller. In their work, they compare the performance of the M/G/1 with M/M/1 and the results from the work support the initial suggestions made by authors in [13] that M/G/1 is a more accurate approximation of the SDN controller performance than M/M/1. Also, the results from the experiments conducted in [22] further validate the claim that M/G/1 is a better fit than M/M/1 for the evaluation of the performance of SDN controller.

In the case of SDN switches, an appropriate model needs to consider both the external packet arrival rate at the switch and the packet arrival at the switch from the controller, as this will ensure that the QoS requirements of SDN environment is properly captured. Unfortunately, most of the work studied did not address this concern in their modelling of SDN switches, which is a limitation when seeking to analyse the interaction with adversarial packet arrival. However, authors in [1, 17, 18] used GI/M/1/K distributions to correctly model the packet arrival at the switch, taking into account

both the external packet arriving at the switch and the packet arriving at the switch from the controller. These works suggest that GI/M/1/K distribution is a more appropriate model for characterising the behaviour of SDN switch.

Moreover, most of the earlier work reviewed relies on heavily simplified models, which do not allow insight into the internal functioning of SDN. This is because they treat flows architecture elements as black boxes, which is interesting for security analysis as an attacker may explicitly target those flows. Although none of the works presented in this survey used queueing networks for the analysis of the security requirements of SDN, earlier work [24, 29, 26, 23, 27, 28] showed that it is possible to study DoS, DDoS, performance degradation, de-synchronization attacks, injections attacks, tails attacks, and economic denial of sustainability (EDoS) attacks using queueing networks. This can also be applied for the analysis of SDN to discover attacks and their potential solutions.

We also note that all work considered so far captured only SDN configurations with a single controller connected to one or more switches; this is not particularly realistic, particularly in critical systems requiring redundancy for resiliency. Such deployments should be seen as network of queues which could lead to analytically modelling them as queueing networks. Hence, there is need to consider such deployments as none of the existing works addressed them because such realistic modelling will provide better performance evaluation and lead to design of a more resilient SDN architecture.

2.7 Conclusions and Future Work

Research in the analysis of SDN using queueing networks is still a developing field, and clearly refinement particularly for analyses relevant for security are still called for. In this paper, we have undertaken a survey of existing work on the use of queueing networks for the analysis of SDN with a focus on the suitability of queueing disciplines to understand effective quality of service modelling and subsequent study of the security properties of SDN.

Future work will include the development of queue networks suitable for systems with hard real-time requirements and where adversarial action may occur. In such systems, there are deadlines that need to be met so as not to violate the QoS guarantees of such systems; the latter brings with it that a number of standard assumptions made in queueing theory and queue networks can no longer be fully supported, mainly on probability density functions and independence assumptions. We then seek to use queueing networks for modelling attacks against such systems. The understanding of these issues will help to address threats to reliability, resilience, and security that may arise from adopting SDN for systems with hard real-time requirements and thus, accelerate the deployment of SDN for such systems.

2.8 Bibliography

[1] Goto, Y.; Masuyama, H.; Ng, B.; Seah, W. K. G. and Takahashi, Y. Queueing Analysis of Software Defined Network with Realistic OpenFlow–Based Switch Model In *2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, IEEE, 2016 45, 51, 56

[2] Karakus, M. and Durresi, A. Quality of service (QoS) in software defined networking (SDN): A survey. *Journal of Network and Computer Applications*, Elsevier, 80 (2017), 200-218. 45

[3] Chen, Y.; Farley, T. and Ye, N. QoS requirements of network applications on the Internet. *Information Knowledge Systems Management*, IOS Press, 4 (2004), 55-76. 45

[4] Open Networking Foundation. SDN architecture. *Technical Report*, 2014. 13, 46, 63

[5] Open Networking Foundation. Software-defined networking: The new norm for networks. *ONF White Paper*, 2012, 2, 2-6. 46

[6] William, S. Software-Defined Networks and OpenFlow. *The Internet Protocol Journal*, 2013. 46, 47

[7] Amazonas, JR.; Santos-Boada, G. and Solé-Pareta, J. A critical review of OpenFlow/SDN-based networks In *2014 16th International Conference on Transparent Optical Networks (ICTON)*, 2014, 1-5. 46

[8] OpenFlow Foundation. OpenFlow Specification. *Open Networking Foundation Publication*, 2015. 46, 47, 67

[9] Smith, J. M. Introduction to Queueing Networks *Springer*, Cham, 2018 8, 9, 13, 48

[10] Pinsky, M. and Karlin, S. An introduction to stochastic modeling *Academic press*, 2010 8, 48

[11] Jarschel, M.; Oechsner, S.; Schlosser, D.; Pries, R.; Goll, S. and Tran-Gia, P. Modeling and performance evaluation of an OpenFlow architecture. In *Proceedings of the 23rd international teletraffic congress*, 2011, 1-7 14, 49, 64, 67, 72

[12] Chilwan, A.; Mahmood, K.; Østerbø, O. N. and Jarschel, M. On modeling controller-switch interaction in openflow based sdns. In *International Journal of Computer Networks and Communications, Academy and Industry Research Collaboration Center (AIRCC)*, 2014, 6, 135 49, 67

[13] MAHMOOD, K.; CHILWAN, A.; ØSTERBØ, O. AND JARSCHEL, M. Modelling of OpenFlow-based software-defined networks: the multiple node case. *IET Networks*, 2015, 4, 278-284 49, 56, 67

[14] YEN, T. AND SU, C. An SDN-based cloud computing architecture and its mathematical model. In *Proc. Electronics and Electrical Engineering 2014 Int. Conf. Information Science*, 2014, 3, 1728-1731 14, 49, 63, 64

[15] MIAO, W.; MIN, G.; WU, Y.; WANG, H. AND HU, J. Performance modelling and analysis of software-defined networking under bursty multimedia traffic. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, ACM, 2016, 12, 77 15, 50, 63, 65, 66, 69

[16] FAHMIN, A.; LAI, Y.; HOSSAIN, M. S.; LIN, Y. AND SAHA, D Performance Modeling of SDN with NFV under or aside the Controller. In *Proc. 5th Int. Conf. Future Internet of Things and Cloud Workshops (FiCloudW)*, 2016, 211-216 50, 64, 67

[17] SINGH, D.; NG, B.; LAI, Y.; LIN, Y. AND SEAH, W. K. G. Modelling Software-Defined Networking: Switch Design with Finite Buffer and Priority Queueing. In *Proc. IEEE 42nd Conf. Local Computer Networks (LCN)*, 2017, 567-570 14, 51, 56, 64

[18] SINGH, D.; NG, B.; LAI, Y.-C.; LIN, Y.-D. AND SEAH, W. K. Modelling Software-Defined Networking: Software and hardware switches. *Journal of Network and Computer Applications*, Elsevier, 2018, 122, 24-36 14, 51, 56, 64, 67

[19] XIONG, B.; YANG, K.; ZHAO, J.; LI, W. AND LI, K Performance evaluation of OpenFlow-based software-defined networks based on queueing model. *Computer Networks*, Elsevier, 2016, 102, 172-185 14, 52, 56, 67

[20] BILEN, T.; AYVAZ, K. AND CANBERK, B. QoS-based distributed flow management in software defined ultra-dense networks. *Ad Hoc Networks*, Elsevier, 2018, 78, 24-31 14, 52, 63, 64

[21] SOOD, K.; YU, S. AND XIANG, Y. Performance Analysis of Software-Defined Network Switch Using M/Geo/1 Model. *IEEE Communications Letters*, 2016, 20, 2522-2525 14, 52, 64, 67

[22] JAVED, U.; IQBAL, A.; SALEH, S.; HAIDER, S. A. AND ILYAS, M. U. A stochastic model for transit latency in OpenFlow SDNs. *Computer Networks*, Elsevier, 2017, 113, 218-229 14, 53, 56

[23] Wright, J. G. and Wolthusen, S. D Stealthy Injection Attacks Against IEC61850's GOOSE Messaging Service. In *Proc. IEEE PES Innovative Smart Grid Technologies Conf. Europe (ISGT-Europe)*, 2018, 1-6 53, 54, 56, 57, 85

[24] Wright, J. G. and Wolthusen, S. D De-Synchronisation Attack Modelling in Real-Time Protocols Using Queue Networks: Attacking the ISO/IEC 61850 Substation Automation Protocol. In *International Conference on Critical Information Infrastructures Security*, 2017, 131-143 54, 56, 57

[25] Wang, Y.; Hu, T.; Tang, G.; Xie, J. and Lu, J. SGS: Safe-Guard Scheme for Protecting Control Plane Against DDoS Attacks in Software-Defined Networking. *IEEE Access*, IEEE, 2019

[26] Hao, S.; Song, H.; Jiang, W. and Dai, Y. A queue model to detect DDos attacks . In *Proceedings of the 2005 International Symposium on Collaborative Technologies and Systems, CTS 2005, Saint Louis, Missouri, USA, May 15-20, 2005*, IEEE Computer Society, 2005, 106-112 54, 55, 57

[27] Shan, H.; Wang, Q. and Pu, C. Tail Attacks on Web Applications. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA*, ACM, 2017, 1725-1739 54, 55, 57

[28] Al-Haidari, F.; Salah, K.; Sqalli, M. and Buhari, S. M. Performance Modeling and Analysis of the EDoS-Shield Mitigation . *Arabian Journal for Science and Engineering*, Springer, 2017, 42, 793-804 54, 56, 57

[29] Wang, Y.; Lin, C.; Li, Q.-L. and Fang, Y. A queueing analysis for the denial of service (DoS) attacks in computer networks. *Computer Networks*, Elsevier, 2017, 51, 3564-3573 54, 55, 57

Chapter 3

Modelling Adversarial Flow in Software-Defined Industrial Control Networks Using a Queueing Network Model

Modelling Adversarial Flow in Software-Defined Industrial Control Networks Using a Queueing Network Model

2020 IEEE Conference on Communications and Network Security (CNS), Avignon, France

Livinus Obiora Nweke and Stephen D. Wolthusen

Abstract

In recent years, software defined networking (SDN) has been proposed for enhancing the security of industrial control networks. However, its ability to guarantee the quality of service (QoS) requirements of such networks in the presence of adversarial flow still needs to be investigated. Queueing theory and particularly queueing network models have long been employed to study the performance and QoS characteristics of networks. The latter appears to be particularly suitable to capture the behaviour of SDN owing to the dependencies between layers, planes and components in an SDN architecture. Also, several authors have used queueing network models to study the behaviour of different application of SDN architectures, but none of the existing works have considered the strong periodic network traffic in software-defined industrial control networks. In this paper, we propose a queueing network model for software-defined industrial control networks, taking into account the strong periodic patterns of the network traffic in the data plane. We derive the performance measures for the analytical model and apply the queueing network model to study the effect of adversarial flow in software-defined industrial control networks.

3.1 Introduction

Recent advances in networking technology have witnessed a gradual shift in industrial control networks from serial-based communication to Ethernet technology. This provides the opportunity of exploiting the benefits of SDN

for industrial control networks because the correctness of industrial control system protocols rely on the guaranteed QoS. SDN is a network paradigm in which "the control and data plane are decoupled, network intelligence and state are logically centralized, and the underlying network is abstracted from the applications" [1]. The separation of the control and data plane provides several advantages which include easier network management, network programmability, increased visibility into the network, efficient use of network resources and dynamic updating of network policies. However, SDN not only offers benefits but also raises questions about the ability to satisfy the guaranteed QoS in the presence of adversarial flow.

Adversarial flow is an additional flow usually malicious, that may be introduced into the SDN architecture by an attacker. The arrival rate and service time distribution of the adversarial flow can be manipulated by the attacker to achieve desired ends. We had described the idea of adversarial flow in our previous work [2] and opined that modelling based on aggregating distribution or modelling based on analysing the effect of two queues separately may be deployed to study the effect of adversarial flow in the analysis of SDN using queueing network models.

Queueing network models have been employed in the literature for performance evaluation of different application of SDN architectures [3, 4, 5, 14, 6]. The approach adopted by several authors involves the use of well-established results from queueing theory to study the interaction between the forwarding plane switches and the control plane controllers. Although simulations and experimentations could be desirable, analytical models, allow the study of wider configurations and how different parameter choices would affect the behaviour of network traffic.

The existing literature on the analysis of SDN using queueing network models has examined the different features of SDN and also its application in different domains. However, *none of the existing literature has considered software-defined industrial control networks and the strong periodic patterns of the network traffic*. Also, the effect of adversarial flow in SDN architectures has not been investigated using queueing network models. We had proposed the use of queueing network models to study the effect of adversarial flow in [2]. We exploit the idea here, to investigate the effect of adversarial flow in software-defined industrial control networks.

In this paper, we propose an analytical model for software-defined industrial control networks taking into account the strong periodic patterns of the network traffic in the data plane. Analytical modelling of software-defined industrial control networks provides useful insights for benchmarking and facilitates the identification of factors that could cause the network to breach the stringent QoS requirements. In addition, we use the analytical model to study the effect of adversarial flow in software-defined industrial control networks.

The rest of this paper is organised as follows. Section 3.2 presents related works on the analysis of SDN using queueing network models. Section 3.3 describes the queueing network model used in this paper to model the behaviour of network traffic in software-defined industrial control networks and the performance measures. Section 3.4 discusses the application of the developed queueing network model to investigate the effect of adversarial flow in software-defined industrial control networks. Section 3.5 concludes the paper and presents future works.

3.2 Related Works

The first known analytical modelling of SDN using a queueing network model is presented by Jarschel et al. in [7]. The work focus on characterizing the interaction between SDN switch and the controller without consideration for any specific application of the SDN architecture. In the same way, authors in [9, 10, 11, 8] employ queueing network models to study different features of SDN. The tradeoffs between buffer sharing mechanisms are investigated using queueing network models in [9]. The authors in [10] propose a queueing network model to characterize the performance of hardware switches and software switches in SDN. The paper in [11] presents the performance study of SDN switches using a queueing network model. And the authors in [8] use a queueing network model to examine the performance of SDN with network virtualization function under or aside the controller.

Yen and Su in [3] deploy a queueing network model to examine SDN-based cloud computing architecture. The queueing network model is based on M/M/1 (using Kendall's notation) queue, and they demonstrate it is appropriate for modelling the operation of SDN-based cloud computing architecture. Also, they show that the proposed SDN-based cloud computing architecture can provide QoS guarantees for cloud services. Chowdhary and Huang in [4] use a queueing network model to consider SDN-based network function parallelism in the cloud. They use a M/M/c queue for optimizing the service function allocation for every service function chain and show that service functions with independent action sets can be parallelized to reduce the performance overhead.

Queueing network models have also been used to study the behaviour of SDN in ultra dense network and satellite communication networks [5, 14]. The authors in [5] examine the use of SDN to ease the management of ultra dense data plane with distributed controllers. They propose a distributed flow management model based on queueing network model and characterize the distributed controllers by considering the flow characteristics and outage. M/M/1 and M^X/M/1 queues are use to model the incoming mice and elephant flows respectively, with an additional M/M/c queue for de-

tection of an outage. In [14], the authors propose an analytical model for software defined satellite networks using a queueing network model. They place the controllers on geosynchronous earth orbit (GEO) satellites and the forwarding functions on medium earth orbit (MEO) satellites and low earth orbit (LEO) satellites. M/M/1 queues are then used to model both the control plane and the data plane; finally, numerical analysis is employed to analyse the effect of different parameters on the file sojourn time.

Similar to our work is the performance modelling and analysis of SDN under bursty multimedia traffic [6]. The authors use Markov modulated Poisson process (MMPP) to investigate the performance of SDN in the presence of bursty and correlated arrivals. However, they assume that the packet departure process from the MMPP queue is MMPP to allow for tractable analytical model. Unlike them, we consider network traffic in software-defined industrial control networks. We use the results from [15] to better model the packet arrival process at the control plane. Also, we observe how adversarial flow may impact network traffic in software-defined industrial control networks.

In contrast to all the works presented above, we exploit queueing network model to analyse the behaviour of network traffic in software-defined industrial control networks. We consider the strong periodic patterns of the network traffic in industrial control networks and approximate the arrival process using MMPP. Also, we use results from [15] to obtain a realistic characterization of the interaction between the data plane and the control plane. We then apply the analytical model to study the effect of adversarial flow in software-defined industrial control networks.

3.3 System Modelling

In this section, we present a discussion on the queueing network model used in this study to model the behaviour of network traffic in software-defined industrial control networks. Also, we describe the performance measures that may be used for investigating how different parameter choices would affect the behaviour of network traffic. These performance measures are deployed in the application of the model to study adversarial flow in software-defined industrial control networks as presented in section 3.4.

3.3.1 The Software-Defined Industrial Control Network

We consider the software-defined industrial control network proposed in [16]. The data plane consists of Raspberry Pis (RPis), sensors, and actuators. RPis are used for receiving packets from sensors and instructing the relevant actuators to take actions based on the respective flow retrieved from the flow table or corresponding controller [16]. An existing flow is retrieved

from the flow table while a new flow is sent to the controller via a Packet-In message. On getting the Packet-In message, the controller instructs the RPi on how to forward the flow via a Packet-Out/Flow-MOD message. The interaction between the data plane and the control plane occurs through the southbound interface of the control plane.

Moreover, many recent measurement studies have shown that network traffic in industrial control networks exhibits strong periodic patterns [17, 18, 19]. These are usually bursty and cannot be properly modelled using Possion arrival process. Also, a peculiar characteristic of industrial control network traffic is that there are a number of components interacting with each other, which implies that the bursty and correlated nature of the traffic can be captured by superposing multiple bursty sources. And considering that the MMPP has shown to be an effective tool for capturing time-varying arrival, it would be appropriate for modelling the bursty and correlated arrival patterns of network traffic in software-defined industrial control networks [20, 6].

The MMPP is a doubly stochastic Poisson process with time-varying arrival and can be obtained by varying the arrival rate of a Poisson process according to an m-state irreducible continuous Markov chain that is independent of the arrival process [20]. Several studies have exploited a two-state MMPP to evaluate the performance of a network [21, 6]. Whilst it is possible to model the network traffic using a single distribution, we use multiple similar distributions to capture the superposition of multiple bursty sources. The superposition of MMPPs has shown to generate MMPP and if the process to be superposed is identical, the complexity is greatly reduced [20]. This allows us to derive the analytical model; as the infinitesimal generator matrix Q of the Markov chain and arrival matrix Λ can be parametrized as follows:

$$Q = \begin{bmatrix} -\varphi_1 & \varphi_1 \\ \varphi_2 & -\varphi_2 \end{bmatrix} and\, \Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

where φ_1 is the transition rate from state 1 to 2, and φ_2 is the rate out of state 2 to 1. Also, λ_1 and λ_2 represent the traffic rates when the Markov chain is in states 1 and 2, respectively. We can then obtain the average rate of arrival λ_s of network traffic in industrial control networks as:

$$\lambda_s = \frac{\varphi_1\lambda_2 + \varphi_2\lambda_1}{\varphi_1 + \varphi_2} \tag{3.1}$$

3.3.2 Queueing Network Model for Software-defined Industrial Control Networks

We employ queueing network model here, to study the behaviour of network traffic in software-defined industrial control networks. The arrival

process of packet at the data plane of SDN deployed in industrial control networks consists mainly of the network traffic modelled using MMPP. This makes up the existing flow in the software-defined industrial control networks. However, there could be a new flow that needs the attention of the control plane. This type of flow does not have an entry in the flow table of the RPi (switch) and it would have to be processed by the control plane.

For the control plane, we consider the case of multiple controllers. Multiple controllers are usually deployed to improve the resilience of SDN architecture. This is to ensure that the network continues to function even if one of the controllers fails. The handover between the controllers in the case of failure is initiated by the controllers themselves, which facilitates fast recovery from failure and also controllers load balancing [22]. The mechanism used by the controllers to manage the data plane devices among themselves is beyond the scope of this paper. Also, according to the OpenFlow specification [22], the data plane devices must connect to all the controllers they are configured with and would try to maintain connectivity with all of them concurrently. Hence, the packets departing from the data plane are simultaneously fed into the controllers in the control plane of the SDN architecture. A more complex architecture for the controllers have been considered in [23, 24] but for the purpose of the discussion in this paper, we are concentrating on this simple model.

The arrival process of packets at the controllers in the control plane is dependent on the departure process of packets from the data plane. Tian [15] have shown that the output process of the Markovial arrival process is not a Markovial process, but rather, it becomes a Poisson process. Thus, like most existing works on the analysis of SDN using queueing network models, we employ a Poisson arrival process with exponential service time distribution (M/M/1) to model the behaviour of the controllers in the control plane [7, 12, 13, 11, 25, 8, 10]. Also, to obtain a more realistic characterization of the controllers' behaviour, we use a finite capacity queue of M/M/1 (M/M/1/K) for the controllers at the control plane. The queueing network model used for this study is shown in figure 3.1, such that $n < m$.

From the queueing network model of the software-defined industrial control network shown in figure 3.1, an arriving packet at the data plane is checked to see if it belongs either an existing flow or if it is a new flow. The probability that the arriving packet is a new flow is given as p. If the arriving packet is a new flow, the packet is sent to the controllers in order to obtain information on how the packet should be forwarded. In the case where the arriving packet belongs to an existing flow, the packet is forwarded with a probability of $1 - p$ without consulting the controllers.

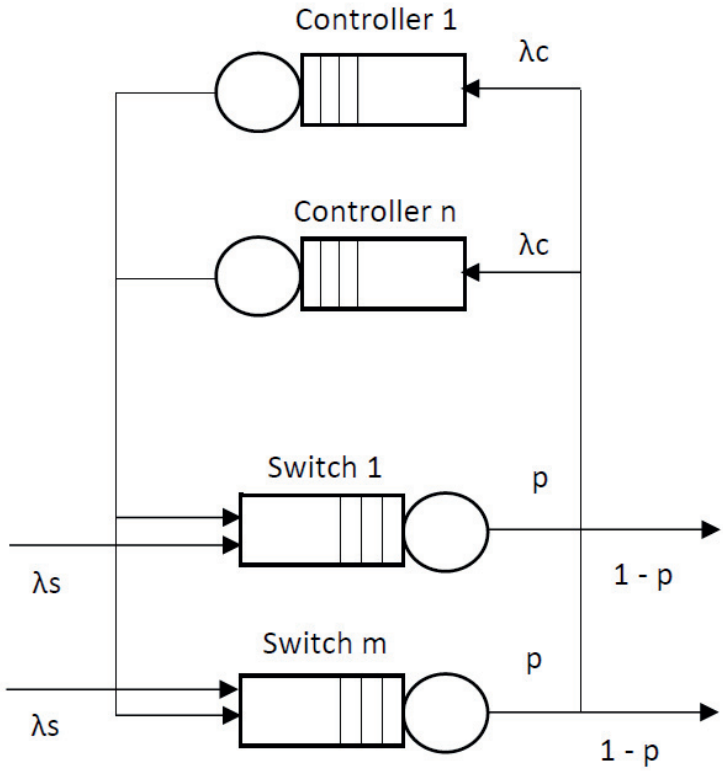


Figure 3.1: Queueing Network Model for Software-defined Industrial Control Networks

3.3.3 Performance Measures

Let T_s be the forwarding time of a packet through the data plane, and T_c be the forwarding time of a packet through the control plane, then we can obtain the total forwarding time of a packet through the queueing network model, T by dividing the packet forwarding into two cases: forwarding without the intervention of control plane and forwarding with the involvement of the control plane. The latter case consists of the sojourn time of the packet (the expected time spent by the packet) in the data plane T_s, and the sojourn time of the packet in the control plane T_c, while the former is just the sojourn time of the packet in the data plane. Thus, we have:

$$T = \begin{cases} T_s & \text{with probability } 1-p \\ T_s + T_c & \text{with probability } p \end{cases}$$

Also, we can obtain the total forwarding time of packet through the queueing network model as:

$$\begin{aligned} T &= (1-p)T_s + p(T_s + T_c) \\ &= T_s + pT_c \end{aligned} \tag{3.2}$$

The value of T_s, which is the average sojourn time in the data plane can be calculated using the method proposed by Fischer and Meier-Hellstern [20]:

$$\begin{aligned} T_s = \frac{1}{\rho}\Big(\frac{1}{2(1-\rho)}(2\rho + \lambda_s m_1 - 2m_1((1-\rho)g + m_2\pi\Lambda) \\ (Q + e\pi)^{-1}\lambda_e) - \frac{1}{2}\lambda_s m_2\Big) \end{aligned} \tag{3.3}$$

where ρ is the traffic intensity at the data plane, given by $\rho = \lambda_s/\mu_s$; m_1 and m_2 are the mean and the second moment of the service time, given by $m_1 = 1/\mu_s$ and $m_2 = 2/{\mu_s}^2$, respectively; $ge = 1$ and $g\mu_s = (1-\rho)^{-1}$ [26]; $\pi = (\varphi_1, \varphi_2)/(\varphi_1 + \varphi_2)$; and $\lambda_e = (\lambda_1, \lambda_2)$.

In addition, the blocking probability P_{b_s} that an arriving packet finds the buffer full can be obtained by calculating the probability P'_n, $0 \leq n \leq K_s$ that there are n packets in a MMPP/M/1/K queue [27] and it is given by [28]:

$$P'_n = \left(\sum_{n=0}^{K_s} P_n \times \Lambda \times e\right)^{-1} P_n \times \Lambda \times e \tag{3.4}$$

Thus, the blocking probability P_{b_s} can be written as [27]:

$$P_{b_s} = P'_{K_s} \tag{3.5}$$

For the value of T_c, the packet arriving at the control plane is obtained from the packet departing from the data plane. Unlike the assumption made by Maio et al. [6] that the packet departure process from the MMPP queue is MMPP to allow for a tractable analytical model, we use results from Tian [15] to model the packet arrival process at the control plane. The results show that the departure process of the Markovial arrival process is not Markovial but rather, it becomes a Poisson process [15]. Hence, the packet departure process of the Markovial arrival at the data plane, which is the same as the packet arrival at the control plane λ_c can be obtained using the Laplace transform matching method and it is given as [15]:

$$\lambda_c = \mu_s(\lambda_1\lambda_2 + \lambda_1\varphi_2 + \lambda_2\varphi_1) \tag{3.6}$$

Given that N_c is the number of packets in the M/M/1/K queue at the control plane, the average number of packets at the control plane E[N_c] can be calculated using the standard formula given as:

$$E[N_c] = \frac{\rho(1-(K+1)\rho^K + K\rho^{K+1})}{(1-\rho)(1-\rho^{K+1})} \tag{3.7}$$

where ρ is the traffic intensity at the control plane (λ_c/μ_c) and K is the buffer size .

We can then derive the value of T_c, which is the average sojourn time in the control plane using Little's Law as follows:

$$T_c = \frac{\rho^{K+1}(K\rho - K - 1) + \rho}{\lambda_c(1-\rho)(1-\rho^K)} \tag{3.8}$$

The blocking probability, which is the probability that an arriving packet finds the system full is given as:

$$P_{b_c} = \rho^K \frac{1-\rho}{1-\rho^{K+1}} \tag{3.9}$$

3.4 Modelling the Adversarial Flow Using the Queueing Network Model

In this section, we investigate the effect of adversarial flow in software-defined industrial control networks using the developed queueing network model. We model the adversarial flow by aggregating the adversarial traffic and the regular traffic. We assume that the attacker has probabilistic knowledge of the regular traffic and using this assumption, we observe how an attacker may vary the arrival rate of the adversarial traffic to increase the average sojourn time of traffic in the network leading to a breach of QoS requirement. The main objective of the attacker is to cause a denial of service (DoS) attack.

3.4.1 The Adversarial Flow Model

In order to model the adversarial flow, there are different hierarchies of models that may be considered. These hierarchies of models would be based on how much knowledge the adversary is using to formulate the adversarial flow model. The first level of the model involves the superposition of the adversarial traffic and the regular traffic. In the second level of the model, a stochastic model of the legitimate traffic can be used for the modelling the adversarial flow. Furthermore, the third level of the model involves the use of actual observation of the regular traffic for modelling the adversarial flow.

Although the hierarchy of models describes a spectrum of how much information we allow the attacker to have when formulating the model, we present the first level of the model in this subsection. Along that spectrum, what we then describe is the simplified version of the adversarial flow model to show the utility of our model. We assume that the attacker has some knowledge of the existing flow and is able to create the model of the regular flow. This will allow the attacker to adapt future adversarial flow by looking at the historical flows that have been seen. The attacker does not need to see the actual traffic, but rather the model of the traffic can be employed to adapt the adversarial flow. It is then possible for the attacker to use the knowledge of the model to modify the adversarial flow in such a way that it could increase the average sojourn time of network traffic which may cause a DoS attack.

The first level of the modelling, which is the superposition of the adversarial traffic and the regular traffic is shown in figure 3.2. The arrival rate of traffic λ_s to the queueing network model is then the ratio of the arrival rate of regular traffic λ_R as in (3.1), and that of the adversarial traffic λ_A (which is under the control of the attacker). Already, we have assumed that the adversary knows the model of the regular traffic, which implies that the adversary is aware that the regular traffic follows MMPP arrival process. We can then evaluate how an adversary can adapt the arrival rate of the adversarial traffic by looking at the regular traffic such that the average sojourn time of network traffic is increased in the next subsection.

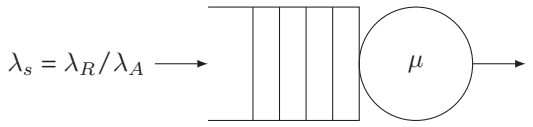


Figure 3.2: Superimposing Regular Flow and Adversarial Flow

3.4.2 Evaluation of the Model

The purpose of this evaluation is to study the effectiveness of adding an adversarial flow in breaching QoS parameters. To do this, we parametrize the queueing network model by first assuming that the traffic rates of the regular traffic are same as the traffic rates of the adversarial traffic. This is to allow us to examine how an intelligent attacker may manipulate the arrival rate of the adversarial flow against the regular traffic in such a way that the probability of the network traffic breaching QoS requirement is high. An explicit estimate of the combined traffic can then be obtained such that we take the regular traffic and express the adversarial traffic as an addition with some probability of extra traffic.

The remaining variables of the queueing network model are parametrized as follows. We assume that the service rate of the data plane is same as the control plane, the infinitesimal generator of the MMPP arrivals is set as $\varphi_1 = 0.06$ and $\varphi_2 = 0.03$, and we use the same buffer sizes for both the data plane and the control plane. Also, we use the result from [7] that show the probability that an arriving packet is a new flow, $p = 0.04$ as the worst case estimate. By varying the arrival rate of the adversarial traffic against the regular traffic, we observe the condition under which the adversarial traffic would increase the average sojourn time of traffic in the network.

In the scenario that we described, an attacker with some knowledge of the regular traffic can increase the arrival rate of the adversarial traffic in order to increase the average sojourn time of traffic in the network. The attacker may continue to increase the arrival rate of the adversarial traffic in a stealthy way to avoid detection until it causes the network to breach the QoS requirement. This implies that given a regular traffic with parameters in the preceding paragraphs, if the attacker is able to create adversarial flow with parameters described in figure 3.3 then there would be a breach of QoS requirement.

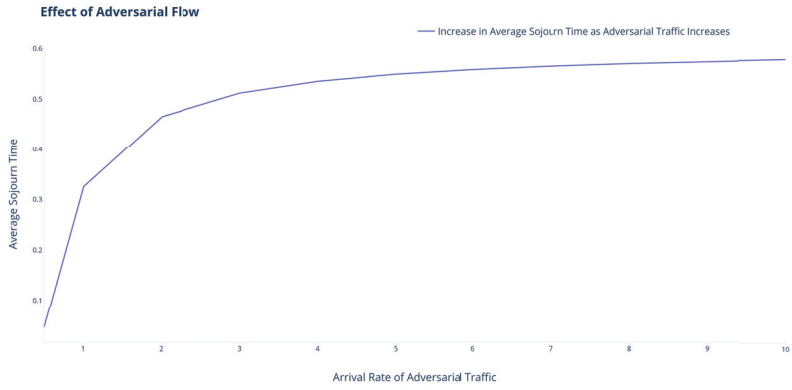


Figure 3.3: Average Sojourn Time vs Arrival Rate of Adversarial Traffic

We can infer from our study that the probability of breaching the QoS requirements of network traffic in software-defined industrial control networks would increase with a knowledgeable attacker. An attacker with some knowledge of the regular traffic is able to manipulate the arrival rate of the adversarial flow according to the nature of the regular traffic. By so doing, the attacker can launch an attack at the right time so as to trigger an adverse effect or event. Therefore, the arrival rate of the adversarial flow would have to be at least higher than the arrival rate of the regular traffic, so that the probability of this breaching the boundary is high.

Considering the attacker's objective, this can effectively create an optimization problem with a number of constraints. The optimization problem is not a static optimization, but has constraints that can be expressed in the form of functions. This is because the attacker is modulating based on the existing bursty traffic. In addition, the attacker does not just create a DoS attack, but is constrained by the number of adversarial flows that can be used to maximise disruption and stealthiness at the same time. We are formulating this as standard optimization problem, but we are not dealing with the optimization part here, because it is a well-known and well-solved problem.

We can then say that given the constraints faced by the attacker which can be expressed in the form of functions, these would be the criteria for getting the sojourn time over the threshold that could result in a DoS attack. These constraints have to be solvable by the attacker to achieve the desired goal (i.e., DoS attack). Under these constraints, the attacker can adapt some parameters like the arrival rate to increase the average sojourn time of network traffic. Although there are more clever ways of modulating the attacker traffic than the superposition with some knowledge of the regular traffic, we use the idea here to show the capability of an intelligent attacker.

3.5 Conclusions and Future Work

Research in the applicability of SDN for industrial control networks is still on-going and consideration for how adversarial flow may impact network traffic in software-defined industrial control networks is very pertinent. In this paper, we have proposed an analytical model using a queueing network model to study the behaviour of network traffic in software-defined industrial control networks. We have observed that the network traffic in industrial control networks exhibit strong periodic patterns and we approximated the arrival process at the data plane using MMPP. Also, we derived the performance measures for the analytical model and then applied the model to study the effect of adversarial flow in software-defined industrial control networks.

Our study indicates that an adversary with some knowledge of the regular traffic is able to increase the arrival rate of the adversarial traffic in such a way that it is higher than that of the regular traffic to increase the average sojourn time of traffic in the network. It is also possible for the attacker to continue to increase the arrival rate of the adversarial traffic in a covert manner. This ensures that the intrusion detection system will not notice the additional malicious traffic until the QoS requirement has been breached.

There are several directions for future research within the analysis of software-defined industrial control networks using queueing networks. One direction is to layer a particular application domain on top of the queueing network model. For example, it is possible to consider the condition under

which the adversarial flow in SDN deployed for IEC 61850 substation could breach the performance bounds as specified by the standard. Another direction is to investigate refinements of the probability density function of the regular traffic and to derive its performance metrics. This could then be referenced when the regular traffic is combined with adversarial traffic to study the effect of adversarial flow.

3.6 Bibliography

[1] OPEN NETWORKING FOUNDATION. SDN architecture. *Technical Report*, 2014. 13, 46, 63

[2] NWEKE, L. O. AND WOLTHUSEN, S. D. Resilience Analysis of Software-Defined Networks Using Queueing Networks In *2020 International Conference on Computing, Networking and Communications (ICNC)*, Feb 2020, IEEE. 14, 63

[3] YEN, T. AND SU, C. An SDN-based cloud computing architecture and its mathematical model. In *Proc. Electronics and Electrical Engineering 2014 Int. Conf. Information Science*, 2014, 3, 1728-1731 14, 49, 63, 64

[4] CHOWDHARY, A. AND HUANG, D. SDN based Network Function Parallelism in Cloud. In *Proc. Networking and Communications (ICNC) 2019 Int. Conf. Computing*, 2019, 486-490 14, 63, 64

[5] BILEN, T.; AYVAZ, K. AND CANBERK, B. QoS-based distributed flow management in software defined ultra-dense networks. *Ad Hoc Networks*, Elsevier, 2018, 78, 24-31 14, 52, 63, 64

[6] MIAO, W.; MIN, G.; WU, Y.; WANG, H. AND HU, J. Performance modelling and analysis of software-defined networking under bursty multimedia traffic. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, ACM, 2016, 12, 77 15, 50, 63, 65, 66, 69

[7] JARSCHEL, M.; OECHSNER, S.; SCHLOSSER, D.; PRIES, R.; GOLL, S. AND TRAN-GIA, P. Modeling and performance evaluation of an OpenFlow architecture. In *Proceedings of the 23rd international teletraffic congress*, 2011, 1-7 14, 49, 64, 67, 72

[8] FAHMIN, A.; LAI, Y.; HOSSAIN, M. S.; LIN, Y. AND SAHA, D Performance Modeling of SDN with NFV under or aside the Controller. In *Proc. 5th Int. Conf. Future Internet of Things and Cloud Workshops (FiCloudW)*, 2016, 211-216 50, 64, 67

[9] SINGH, D.; NG, B.; LAI, Y.; LIN, Y. AND SEAH, W. K. G. Modelling Software-Defined Networking: Switch Design with Finite Buffer and Priority Queueing. In *Proc. IEEE 42nd Conf. Local Computer Networks (LCN)*, 2017, 567-570 14, 51, 56, 64

[10] SINGH, D.; NG, B.; LAI, Y.-C.; LIN, Y.-D. AND SEAH, W. K. Modelling Software-Defined Networking: Software and hardware switches. *Journal of Network and Computer Applications*, Elsevier, 2018, 122, 24-36 14, 51, 56, 64, 67

[11] SOOD, K.; YU, S. AND XIANG, Y. Performance Analysis of Software-Defined Network Switch Using M/Geo/1 Model. *IEEE Communications Letters*, 2016, 20, 2522-2525 14, 52, 64, 67

[12] CHILWAN, A.; MAHMOOD, K.; ØSTERBØ, O. N. AND JARSCHEL, M. On modeling controller-switch interaction in openflow based sdns. In *International Journal of Computer Networks and Communications, Academy and Industry Research Collaboration Center (AIRCC)*, 2014, 6, 135 49, 67

[13] MAHMOOD, K.; CHILWAN, A.; ØSTERBØ, O. AND JARSCHEL, M. Modelling of OpenFlow-based software-defined networks: the multiple node case. *IET Networks*, 2015, 4, 278-284 49, 56, 67

[14] LI, T.; ZHOU, H.; LUO, H.; QUAN, W. AND YU, S. Modeling software defined satellite networks using queueing theory. In *Proc. IEEE Int. Conf. Communications (ICC)*, 2019, 1-6 14, 63, 64, 65

[15] TIAN, W. Analysis and efficient provisioning of access networks with correlated and bursty arrivals. *Int. J. Communication Systems*, 2014, 27, 551-570 15, 21, 65, 67, 69

[16] AHMED, K.; BLECH, J. O.; GREGORY, M. A. AND SCHMIDT, H. W. Software defined networks in industrial automation. *Journal of Sensor and Actuator Networks*, Multidisciplinary Digital Publishing Institute, 2018, 7, 33 65

[17] BARBOSA, R. R. R.; SADRE, R. AND PRAS, A. Exploiting traffic periodicity in industrial control networks. *IJCIP*, 2016, 13, 52-62 66

[18] JIEXIN ZHANG; SHAODUO GAN; LIU, X. AND ZHU, P. Intrusion detection in SCADA systems by traffic periodicity and telemetry analysis. In *Proc. IEEE Symp. Computers and Communication (ISCC)*, 2016, 318-325 66

[19] BARBOSA, R. R. R.; SADRE, R. AND PRAS, A. Towards periodicity based anomaly detection in SCADA networks . In *Proc. IEEE 17th Int. Conf. Emerging Technologies Factory Automation (ETFA 2012)*, 2012, 1-4 66

[20] Fischer, W. and Meier-Hellstern, K. S. The Markov-Modulated Poisson Process (MMPP) Cookbook. *Perform. Eval*, 1993, 18, 149-171 66, 69

[21] Mark, B. L. and Ephraim, Y. Explicit Causal Recursive Estimators for Continuous-Time Bivariate Markov Chains. *IEEE Transactions on Signal Processing*, 2014, 62, 2709-2718 66

[22] OpenFlow Foundation. OpenFlow Specification. *Open Networking Foundation Publication*, 2015. 46, 47, 67

[23] Wang, G.; Li, J. and Chang, X. Modeling and performance analysis of the multiple controllers' approach in software defined networking. In *Proc. IEEE 23rd Int. Symp. Quality of Service (IWQoS)*, 2015, 73-74 67

[24] Bozakov, Z. and Rizk, A. Taming SDN Controllers in Heterogeneous Hardware Environments. In *Proc. Second European Workshop Software Defined Networks*, 2013, 50-55 67

[25] Xiong, B.; Yang, K.; Zhao, J.; Li, W. and Li, K Performance evaluation of OpenFlow-based software-defined networks based on queueing model. *Computer Networks*, Elsevier, 2016, 102, 172-185 14, 52, 56, 67

[26] Lucantoni, D. M. New results on the single server queue with a batch markovian arrival process. *Stochastic Models*, 1991, 7, 1-46 69

[27] Wu, Y.; Min, G. and Yang, L. T. Performance Analysis of Hybrid Wireless Networks Under Bursty and Correlated Traffic. *IEEE Transactions on Vehicular Technology*, 2013, 62, 449-454 69

[28] Meier-Hellstern, K. S. The analysis of a queue arising in overflow models . *IEEE Transactions on Communications*, 1989, 37, 367-372 69

Chapter 4

Adversary Model for Attacks Against IEC 61850 Real-Time Communication Protocols

Adversary Model for Attacks Against IEC 61850 Real-Time Communication Protocols

2020 16th International Conference on the Design of Reliable Communication Networks DRCN 2020, Milan, Italy

Livinus Obiora Nweke, Goitom Kahsay Weldehawaryat and Stephen D. Wolthusen

Abstract

Adversarial models are well-established for cryptographic protocols, but distributed real-time protocols have requirements that these abstractions are not intended to cover. The IEEE/IEC 61850 standard for communication networks and systems for power utility automation in particular not only requires distributed processing, but in case of the generic object oriented substation events and sampled value (GOOSE/SV) protocols also hard real-time characteristics. This motivates the desire to include both quality of service (QoS) and explicit network topology in an adversary model based on a π-calculus process algebraic formalism based on earlier work. This allows reasoning over process states, placement of adversarial entities and communication behaviour. We demonstrate the use of our model for the simple case of a replay attack against the publish/subscribe GOOSE/SV subprotocol, showing bounds for non-detectability of such an attack.

4.1 Introduction

Real-time communication protocols are among the most prominent communication protocols used in networked critical infrastructures. They are used to monitor and control industrial automation processes deployed in critical infrastructures including power stations, power and water distribution, and traffic systems. The resilience of networked critical infrastructures is depended on the ability of the communication protocols used in such environments to adapt well in the face adversarial actions.

Adversary model describes the capabilities of an attacker [1] and facilitates reasoning about how a system may be compromised. The conventional

adversary models are not suitable for capturing the capabilities of an attacker in IEC 61850 environment due to the stringent QoS requirements and the network topology [2]. Also, the conventional adversary models do not consider the network topology of IEC 61850 because they assume that there is a point-to-point communication between all parties. Thus, it is important to develop an adversary model which takes into account the constraints imposed on an attacker with the intention of attacking the IEC 61850 real-time communication protocols.

We therefore propose an adversary model for IEC 61850 real-time communication protocols in this paper. First, we use IEC 61850 GOOSE messaging service as an example, and derive its formalization using π-calculus variant. We then show that the relative positions of the adversary in relation to the publisher, event notification service, and subscriber determine the type of attacks that can be launched by an attacker. Lastly, we use our model to describe a reply attack that can result in a denial of service (DoS) attack.

The rest of this paper is organized as follows. Section 4.2 presents a general discussion on real-time communication protocols and introduces the π-calculus syntax. Section 4.3 discusses the related works. Section 4.4 describes the adversary model and its formalization using π-calculus variant. Section 4.5 presents the application of our model. Section 4.6 concludes the paper and presents future work.

4.2 Background

This section begins with a general discussion on real-time communication protocols and presents IEC 61850 real-time communication protocols as examples of real-time communication protocols. A detailed discussion on IEC 61850, IEC 62351 and publish-subscribe communication model is presented. The section concludes with a brief discussion on π-calculus which will be used for the formalization of the IEC 61850 GOOSE Messaging Service and the adversarial model.

4.2.1 Real-Time Communication Protocols

Real-time communication protocols can be referred to as the communication protocols used in real-time systems. In real-time systems, "the correctness of the system depends not only on the logical results of the computation, but also on the time at which the results are produced" [3]. These types of systems usually have stringent QoS requirements, and industrial applications constitute the major application area. Examples of industrial applications of real-time systems include but not limited to the following: industrial

automation systems, process control systems, and supervisory control and data acquisition (SCADA) applications.

There are several features inherent in a real-time system which must be considered by communication protocols to be deployed in such an environment. Among these features, the time constraint requirement is of interest in this study because it relates to the QoS parameters that need to be fulfilled by these real-time communication protocols. Every task in real-time systems is time bond and it is expected that a task must be completed within the specified time. For example, if a message transmission time is 3ms, it must be delivered within this time or be considered as lost. In this paper, we present IEC 61850 real-time protocols as examples of real-time communication protocols and using them as a reference model for other protocols that share similar characteristics, to discuss adversary model and to study attacks against real-time communication protocols.

4.2.2 IEC 61850/IEC 62351

IEC 61850 provides a framework for substation integration which defines the communication requirements of substations; the functional characteristics, the structure of data in devices, naming conventions for the data, how applications interact and control devices, and how conformity to the standard should be tested [4]. An important goal of the standard is to enable interoperability among the components in and between substation automation systems. IEC 61850 also aims to support defined processes and procedures of utilities around the world and to provide future-proof standard which may adopt to the dynamic nature of today's environment [5].

In IEC 61850, all application functions, with the data interfaces to the primary equipment are reduced to the smallest possible pieces, which may interact with each other and could be implemented separately in intelligent electronic devices (IEDs) [5]. The IEDs are divided into logical devices that are implemented in servers residing in IEDs. These IEDs contain group of logical nodes or functions, which include all data objects they need for the function. Common classes are defined by the IEC 61850 standard, and vendors of IED may implement the actual data objects based on the class in the IED. The data objects have at least three attributes (value, quality, and time stamp) and they may include other data objects as attributes. Also, IEC 61850 describes how the data objects may be accessed. These are services that may be provided by abstract communication service interface (ACSI). Some of the common services include querying object set, getting/setting data values, controlling system objects, reporting, logging, GOOSE, and SV [5]. All these services are initiated by applications and responded by servers.

Another important observation about IEC 61850 standard is that the defined data objects and the set of abstract communication services (ACSI) are mapped into specific protocols. The ISO/OSI communication stack consist-

ing of Ethernet (layers 1 and 2) and TCP/IP (layers 3 and 4) and manufacturing messaging specification, MMS, (layers 5 to 7) was chosen for the mapping [5]. While the data object model and its services are mapped to the application layer, only time-critical services, such as SV and GOOSE are mapped directly to the Ethernet link layer. In addition, the MMS protocol uses a client/server communication mode that runs over TCP/IP, while SV and GOOSE protocol deploy the publisher/subscriber methodology. Although security is not defined by IEC 61850, a separate standard (IEC 62351) may be used for implementing security measures.

IEC 62351 is the standard that provides security measures for a number of TC57 protocols and parts 3, 4 and 6 of the standard relates to IEC 61850 security [6]. IEC 62351-3 discusses the security for profiles including TCP/IP, IEC 62351-4 has to do with the security rules for MMS, while the security of GOOSE message is the focus of IEC 62351-6 [6]. The aim of the standard is to provide authentication and encryption for the IEC 61850 protocols to prevent attacks against the protocols. Given that digital signatures and encryption methods require a lot of time to generate and verify, the IEC 62351 standard observed that for applications using GOOSE messages with multicast configurations and low CPU overhead, encryption is not recommended [7]. In this paper, we are interested in understanding the adversarial model for time-critical services (SV and GOOSE) and to investigate attacks against such services.

4.2.3 Publish-Subscribe Communication Model

Publish-subscribe communication model as shown in figure 4.1 is a type of communication model which involves two participants, where one acts as a publisher and generates events; and the other as a subscriber and express interest in an event or pattern of events, so as to be notified when the event or pattern of events it indicated interest in, is/are generated [8]. The communication between the publisher and subscriber is anonymous in that both are not aware of the existence of each other. The publisher just produces events which are multi-cast to all the subscribers, and only the subscriber(s) that have expressed interest in the published events would receive them. In addition, the communication between the publisher and the subscriber is achieved asynchronously because the subscriber does not have to be in a blocked waiting state for an event to arrive, but rather, it is able to carry-out concurrent operations.

In the actual implementation of the publisher-subscriber communication model in IEC 61850 environment, taking GOOSE message communication as an instance; the publishing IED writes the values into a local buffer at the sending side, which is then multi-cast; while the subscribing IED(s) reads the values from a local buffer at the receiving side [9]. The communication channel is saddled with the task of updating the local buffers of the

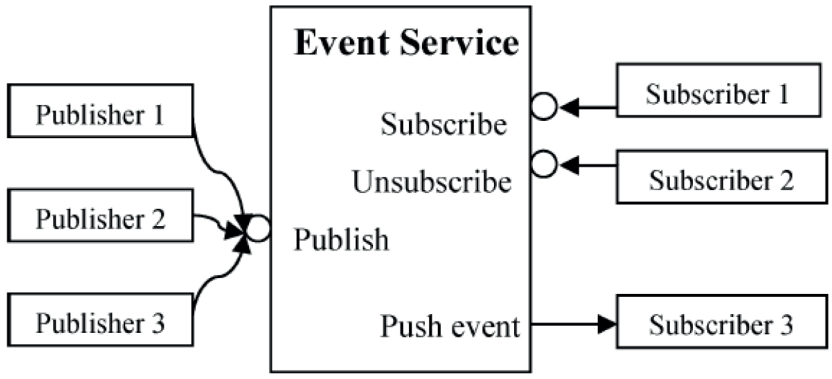


Figure 4.1: Publish-Subscribe Communication Model

[8]

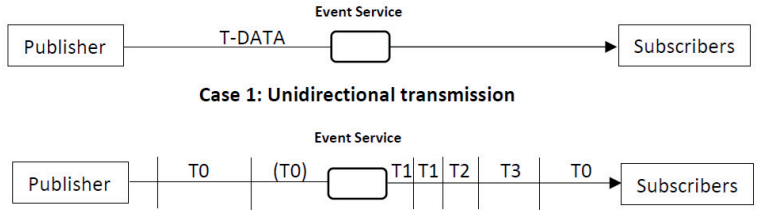
subscribers. And in the case of the publisher, GOOSE-Control-Block is responsible for controlling the overall communication mechanism.

According to the IEC 61850 standard, the transmission of GOOSE message from the publisher to the subscriber is unidirectional and does not require an acknowledgement from the subscriber. The GOOSE message is transmitted as T-DATA (transmitted data) on the multi-cast association. The reliability of the communication is ensured by retransmitting the same message with gradually increasing sequence number and retransmission time. The retransmission interval is not specified by the standard. In addition, the GOOSE message in the retransmission sequence carries a timeAllowedTo-Live parameter, which is used to indicate the maximum time the subscriber would have to wait for the next retransmission. Therefore, if a new GOOSE message is not received within that time interval, the subscriber infers that the message is lost. The semantics of the GOOSE message transmission mechanism is shown in figure 4.2.

4.2.4 The π-Calculus

The π-calculus provides a formal mechanism to model communication among processes over dynamic links [10]. A channel is an abstraction of the communication link between processes, and processes interact by sending information through these channels. An infinite set of names are used for communication channels, and an infinite set of variables (x, y, z, etc) are used to define the terms. The set of processes can be defined by the syntax given in table 4.1.

A composition $P|Q$ behaves as if processes P and Q are running in parallel. Processes operate on channels to communicate with each other and



T-DATA: transmitted GOOSE message on the multicast association

T0: retransmission in stable conditions (no event for a long time)

(T0): retransmission in stable conditions may be shortened by an event

T1: shortest retransmission time after the event

T2, T3: retransmission times until achieving the stable conditions time

Figure 4.2: Transmission of GOOSE Message

Table 4.1: Syntax of π-calculus

Term	Semantics
$P ::=$	Processes
0	empty process
$\bar{x}\langle z\rangle.P$	output
$x(y).P$	input
$P + Q$	choice
$P\|Q$	parallel composition
$!P$	replication
$\nu x.P$	restriction
τ	silent function/action

with the outside the network. The basic interaction is defined using $\bar{x}\langle z\rangle.P$ that defines an output process that is ready to output on channel x, or $x(y).P$ that defines an input process that is ready to receive a value over channel x. The replication $!P$ behaves as an infinite number of copies of P running in parallel. The name restriction operator ($\nu x.P$) is a process that makes a new, private name x, and then behaves as P. τ represents the internal (silent) action of a process that is not observable outside the scope of the process. 0 is the empty process.

A variant of π-calculus is a widely used to model interacting systems rep-

resenting concurrent computations whose configuration may change during computation [11, 2]. Section 4.4 presents a model of the GOOSE messaging service interaction using π-calculus that helps to analyse and understand how the communication paradigm can be exploited by an adversary to manipulate substation system operations.

4.3 Related Works

Adversary model and attacks against real-time systems have been studied over the past years. In this section, we present a discussion on the formalization of attacks in real-time systems and a review of attacks against IEC 61850 in the literature.

4.3.1 Formalization of Attacks

One of the earliest works done on adversary model is presented by Dolev and Yao in [1]. The Dolev-Yao model assumes that the attacker have complete control over the network. Although the paper presented a formal model with limited assumptions on the capabilities of the adversary, it is the foundation on which subsequent adversary models were developed. Efforts have been made in the past by several authors to formalize attacks for real-time systems given the stringent QoS requirements and the fact that the assumptions of the Dolev-Yao model can no longer hold in such constrained environments. The authors in [2] described a formal adversary capability model for SCADA environments and used π-calculus variant to reason about adversarial actions. They argued that the Dolev-Yao model and variants are not suitable for capturing the capabilities of an adversary in a SCADA environment because of the segmented network architecture and real-time processing.

Another interesting work on formalization of attacks was presented in [12] where the authors proposed an adversary model which could be used to study the security promises of real-time systems. In this work, the attacker is assumed to be able to compromise both physical and cyber weaknesses of the systems and the adversary model was able to capture the capabilities and spatial distribution of the adversary. In [13] the authors used a state-based stochastic model to formalize the security properties of real-time systems. They assumed that the system had Markovian property and considering that general probability distributions are assigned to its transitions, the resulting model is a semi-Markov chain. Further, the proposed model is then parametrized based on a time distribution describing the attacker and the system behaviours over time.

A generalized attacker and attack models for real-time systems were presented in [14]. The authors described an attacker model for real-time sys-

tems and used the attack models that were obtained from the attacker model to generate parametrized attack methods for real-time systems. The authors in [15] used the formalism of discrete event systems modelled as finite state automata to reason about the problem of synthesizing an attack strategy for real-time systems. The model presented in the work was able to capture a class of deception attacks, where the attacker is capable of modifying a subset of sensor reading in order to mislead the supervisor and forcing the system into an undesirable state.

In addition, a formal approach for characterizing attacks in real-time systems was presented in [16]. The authors deployed formal methods to capture interactions in a real-time system and to reason about how the system may be attacked. They used a hybrid process calculus to characterize both the system and the attacks against the system. The adversary model used in this work assumed that the adversary is not able to compromise the communication, but may compromise physical devices. Different from the works presented so far, we present an adversary model specifically for IEC 61850 environment. Like most of the works, we argue that the Dolev-Yao model is not suitable for modelling attacks against real-time systems. Thus, we use π-calculus variant to first capture the multicast, publish-subscribe model and then reason about how adversarial actions can compromise the system.

4.3.2 Attacks Against IEC 61850

Attacks against IEC 61850 have been studied over the past years. The authors in [17] presented one of the earliest works on how IEC 61850 can be attacked and the type of attack presented in this paper is referred to as spoofing attack. In this work, the attacker is able to falsify GOOSE message in order to trick the subscribers into accepting the falsified message as legitimate GOOSE message from the publisher. Spoofing attack has also be investigated by authors in [19, 20], and the authors [18] presented an approach for real-time detection of attacks in IEC 61850, which is able to spot spoofing attack by comparing changes in the values in the fields of GOOSE messages.

Injection attack is another type of attack that may be targeted against IEC 61850 real-time protocols. The attack exploits the lack of authentication of the IEC 61850 real-time protocols to insert false data or malicious fault. This type of attack has generated interest in recent years. A stealthy injection attack against IEC 61850 GOOSE messaging service was described in [21]. The authors argued that lack of acknowledgement of received messages and limited security protection makes the GOOSE service vulnerable to injection attack. In the same way, authors in [22, 23] discussed false data injection attacks. Also, a fault injection attack was presented in [24]. This type of attack can be achieved by injecting computation errors in the target either using invasive or non-invasive techniques [24].

Furthermore, it is possible for an attacker to throngs false messages to compete with legitimate messages for the shared network and computing resources which in turn affects the delivery delay of legitimate messages. This type of attack is referred to as flooding attack and it can result to not meeting the timing constraint for message delivery of IEC 61850 real-time protocols. The effects of flooding attacks on time-critical communications in IEC 61850 substation were studied in [25, 26]. Both papers concluded that the effects of flooding attacks are more severe in the wireless network than in the wired network. In addition, the authors in [25] proposed the use of bait message detection-based technique to combat the effects of flooding attacks on time-critical communications in IEC 61850 substation.

An interesting attack peculiar to IEC 61850 real-time protocols is replay attack, where an attacker is able to capture GOOSE or SV messages and then send them without modification to the subscriber at a different time. The goal of the attacker is to trick the subscriber to executing valid commands at the wrong time, which may lead to compromising the normal functioning of the substation. Replay attacks that can be targeted at IEC 61850 real-time protocols were presented in [7] and they include: replay after stNum Reset in the GOOSE protocol and cross receiver replay in the SV protocol. The replay attack against GOOSE protocol exploits the stNum reset features to launch an attack, while the replay attack targeted at the SV protocol exploits the lack of control block reference to craft an attack [7]. Also, in [27] replay attack was simulated on a cost-efficient software test-bed for cyber-physical security in IEC-61850 substation and a network-based cyber intrusion system which is able to detect replay attacks was described in [28].

Also, IEC 61850 standard assumes that the source of the timestamp mechanism is trustworthy but recent studies have shown that an attacker is able to trick the timestamp mechanism to de-synchronize the time base of the station [29, 30, 31, 32, 33]. For example, the authors in [30] demonstrated that a delay box, which can be acquired easily in any fibre shop; is able to introduce time delay by tricking a packet-based time synchronization protocol and injecting an undetectable malicious offset. In addition, lack of message authentication between the master and slave clocks makes the timestamp mechanism vulnerable to attacks as shown in [29, 33]. The attacker is able to exploit lack of message authentication to flood large number of spoofed ANNOUNCE and SYNC packets against a precision time protocol (PTP) network, forcing the slave's clock out of sync with the master clock and the rest of the network [33]. In contrast from these works, we develop an adversary model specifically for the IEC 61850 environment. The developed adversary model is then used to describe a replay attack that can results in a DoS attack to show the application of our model.

4.4 Formal Model

This section presents a formalization of the IEC 61850 GOOSE messaging service using π-calculus variant and description of the adversarial model.

4.4.1 Model of the IEC 61850 GOOSE Messaging Service

We define a model of the IEC 61850 GOOSE messaging service using the π-calculus that captures the publish-subscribe communication model. The basic publish-subscribe interaction relies on an event notification service that provides storage and management for subscriptions and efficient delivery of events. GOOSE messaging service allows the exchange of data between two or more IEDs, where one IED (the publisher) publishes a message that is delivered to a group destinations IEDs (the subscribes). Two instances can trigger the sending of GOOSE messages, and figure 4.2 shows a sequence diagram for message interactions between the *publisher, Event Notification Service, and Subscribers*. We consider three processes, P, N and S corresponding to the publisher, event notification service and subscribe, respectively. The processes are considered to start with their parallel composition $(P|N|S)$, where P and N are connected by a channel c_{PN}, and N and S by a channel c_{NS}. The publisher uses c_{PN} channel for sending a message to the event notification service, and the event notification service use c_{NS} channel for sending a message to the subscriber(s). In informal notation, we may write this communication as follows:

$$\begin{aligned} P \rightarrow N: &\quad M \quad on \quad channel \quad c_{PN} \\ N \rightarrow S: &\quad M \quad on \quad channel \quad c_{NS} \end{aligned}$$

A π-calculus description of this message interaction is:

$$\begin{aligned} P(M) &= \overline{c_{PN}}\langle M\rangle \\ N &= c_{PN}(x).\overline{c_{NS}}\langle x\rangle \\ S &= c_{NS}(x).x(y) \\ Inst(M) &= (\nu c_{PN})(\nu c_{NS})(P(M)|N|S) \end{aligned}$$

$Inst(M)$ describes one instance of the GOOSE protocol, and a publisher sends a *publish (M)* message in a specific event to the event notification service, and it will forward the message to any subscribers interested in that event. However, if an event trigger occurs, the publisher keeps retransmitting the *publish* message until it reaches the stable retransmission time. The

message interaction when an event (E) occurs can be described as follows:

$$\begin{aligned} P \rightarrow N: &\quad M \quad on \quad channel \quad c_{PN} \\ N \rightarrow S: &\quad M \quad on \quad channel \quad c_{NS} \\ N \rightarrow S: &\quad E \quad on \quad channel \quad c_{NS} \\ &\qquad\qquad\qquad\qquad\qquad\quad ..... \\ N \rightarrow S: &\quad M \quad on \quad channel \quad c_{NS} \end{aligned}$$

Case 2 can be represented using the π-calculus as follows.

$$\begin{aligned} &if(t = T_0) \\ &P(M) = \overline{c_{PN}}\langle M\rangle \\ &N = c_{PN}(x).\overline{c_{NS}}\langle x\rangle \\ &S = c_{NS}(x).x(y) \\ &Inst(M) = (\nu c_{PN})(\nu c_{NS})(P(M)|N|S) \\ &else \\ &for(t = T_i(i = 1)); t \leq T_{stableCondition}; t++) \\ &N(E) = \overline{c_{PN}}\langle E\rangle \\ &S = c_{NS}(x).x(z) \\ &endfor \\ &Inst(M) = (\nu c_{PN})(\nu c_{NS})(P(M)|N|S)|(\nu c_{PN}) \\ &(\nu c_{NS})(N(E)|S) \end{aligned}$$

The Publisher IED starts by sending a *Publish* message to the event notification service. The event notification service publishes the message on the c_{PN} channel to the subscriber IEDs. When an event occurs the publisher retransmits the message with a new $Publish$ message.

4.4.2 Adversary Model

An adversary refers to an attacker, often with malicious intent, undertaking an attack on a system or protocol [34]. The goal of the adversary is to disrupt or prevent proper operation of a secure system (e.g., by violating the confidentiality, data integrity or availability of the system). An adversary model is a formalization of an attacker in a computer or networked system. We describe an adversarial model for the IEC 61850 GOOSE publisher-subscriber communication model. In Dolev-Yao adversary model [1], an adversary can control network operations. The assumption on the capabilities of an adversary is very strong in Dolev-Yao model, but it is customized to the IEC 61850 application and communication requirements. Our model allows us

characterize IEC 61850 network topology explicitly in the form of processes and messages.

The features of publish/subscribe services are the causes of the vulnerabilities that an adversary can use to perform an attack and violate the security goals of the service [35, 36, 37, 38]. The adversarial model can be used to describe how different attackers may attack different entities of the publish/subscribe service (*publisher, Event Notification Service, and subscribes)*. Thus, we consider malicious adversaries who can have access to the network communication of the GOOSE publish-subscribe messaging service and can *observe, insert, and modify* events and subscriptions, In other words, we consider adversaries who will attempt to violate confidentiality of events by observing them, and violate integrity and authentication by inserting / injecting fake events and subscriptions. An adversary can be modelled as an arbitrary process running in parallel with the protocol, which can interact with the protocol in order to gain information.

In the following subsections we describe the *adversarial capabilities* to perform security attacks using π-calculus with respect to the entities of the GOOSE publish-subscribe messaging service.

4.4.2.1 Publisher(s)

An adversary may attempt to spoof the identity of a legitimate publisher and send incorrect or fake application data to the pub-sub network nodes. Example of attacks include spoofing and flooding attacks. For instance, malicious publisher(s) can flood the network with a large number of bogus messages from the publisher(s) to the Event Notification Service using channel c_{PN}. A π-calculus description of *case 1* message interaction with a malicious publisher P_{adv} can be given as follows:

$$\begin{aligned} P(M)_{adv} &= \overline{c_{PN}}\langle M\rangle \\ N &= c_{PN}(x).\overline{c_{NS}}\langle x\rangle \\ S &= c_{NS}(x).x(y) \\ Inst_{adv}(M) &= (\nu c_{PN})(\nu c_{NS})(P(M)_{adv}|N|S) \end{aligned}$$

$Inst_{adv}(M)$ describes one instance of the GOOSE protocol, and the compromised publisher sends a *publish (M)* message in a specific event to the event notification service, and the notification service will forward the modified message to any subscribers interested to that event.

4.4.2.2 Event Notification Service

An adversary can target an Event Notification Service to intercept messages, mis-forward messages, or modify messages. For example, an adversary can

intercept and modify the message forwarded by the Event Notification Service to the Subscribers. A π-calculus description of *case 2* message interaction with a compromised notification service N_{adv} can be given as follows:

$$
\begin{aligned}
&if(t = T_0)\\
&P(M) = \overline{c_{PN}}\langle M\rangle\\
&N_{adv} = c_{PN}(x).\overline{c_{NS}}\langle x\rangle\\
&S = c_{NS}(x).x(y)\\
&Inst(M)_{adv} = (\nu c_{PN})(\nu c_{NS})(P(M)|N_{adv}|S)\\
&else\\
&for(t = T_i(i = 1)); t \leq T_{stableCondition}; t++)\\
&N(E_{adv}) = \overline{c_{PN}}\langle E\rangle\\
&S = c_{NS}(x).x(z)\\
&endfor\\
&Inst(M)_{adv} = (\nu c_{PN})(\nu c_{NS})(P(M)|N_{adv}|S)|\\
&(\nu c_{PN})(\nu c_{NS})(N(E)_{adv}|S)
\end{aligned}
$$

When an event occurs the publisher retransmits a new *publish message*, the ability to modify a message is dependent on the message channel containing the compromised notification service N_{adv}. This is possible because the compromise of N provides the adversary a message channel to/from each subscriber node.

4.4.2.3 Subscriber(s)

An adversary may use the subscriber(s) as a potential point of vulnerability in the system if that subscriber does not provide adequate controls on the information received. The adversary may also attempt to spam or flood the pub-sub network with duplicate or fake subscriptions and un-subscriptions. Example of attacks include eavesdropping and replay attacks. For instance, an adversary may be only interested in eavesdropping messages between the Event Notification Service and the subscriber(s). Thus, its definition is to listen or observe messages continuously on the channel c_{NS} over which the Event Notification Service and the subscriber(s) are communicating.

4.5 Application of Our Model

To show the utility of our model, we describe a replay attack that can result in DoS attack in this subsection. The attack involves capturing GOOSE or SV messages and then sending them back to the subscribers at a different time

without modification. It exploits the lack of integrity checks in the IEC 61850 real-time communication protocols. This is because the use of encryption is not recommended so as not to breach the deadlines of time critical services [7]. However, the attacker would try to avoid detection and as such would be constrained by the number of messages which can be injected to achieve the desired end. Also, we assume that the attacker cannot sit anywhere inside the network but would have to choose limited number of places inside the network from which to launch the attack. A scenario of publishing IED's data to the subscribers is shown in the figure 4.3.

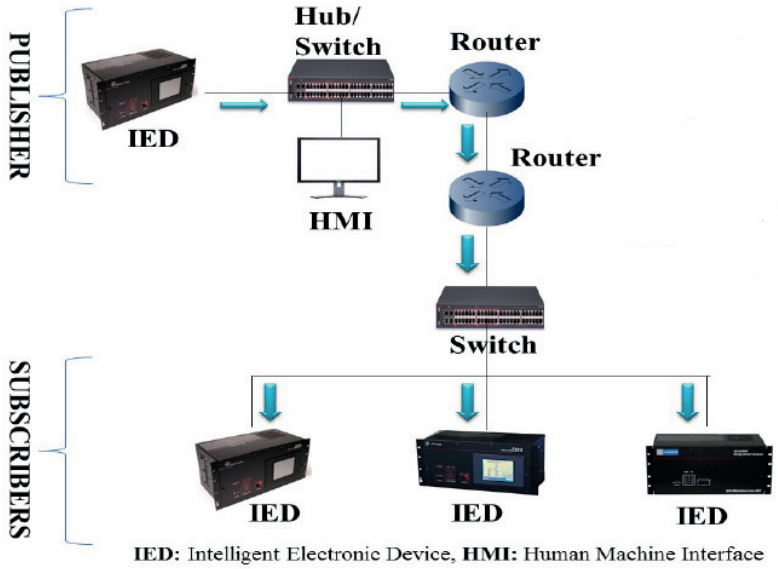


Figure 4.3: Publishing IED's data to the subscribers

[39]

The success of a replay attack in GOOSE/SV scenario above will depend on the timing relationship between the replay and the retransmission. We define the ordering condition and formally derive constraints on the attacker's success. The constraints are based on the semantics of the process as it is described i.e. ordering of messages.

The ordering of the messages would depend on whether the replay occurs before the state change of the publisher. As long as the sender is retransmitting the same measurement value, it does not have an impact on the replay. However, if the retransmission gets to the point where the publisher transition from one value to the other, it is at this point that the attacker

may exploit to insert previously captured message. There is a relatively narrow window when retransmission matters for a reply attack and that is the constraint an attacker would have to deal with, for the attack to succeed.

We provide a π-calculus description of replay attack that can result in DoS attack, given the constraints imposed on the attacker is given as follows:

$$
\begin{aligned}
&if(t = T_0)\\
&P(M) = \overline{c_{PN}}\langle M\rangle\\
&N_{adv} = c_{PN}(x).\overline{c_{NS}}\langle x\rangle\\
&S = c_{NS}(x).x(y)\\
&Inst(M)_{adv} = (\nu c_{PN})(\nu c_{NS})(P(M)|N_{adv}|S)\\
&else\\
&for(t = T_i(i = 1)); t \leq T_{stableCondition}; t++)\\
&N(E)_{adv} = \overline{c_{PN}}\langle E\rangle\\
&S = c_{NS}(x).x(z)\\
&endfor\\
&Inst(M)_{adv} = (\nu c_{PN})(\nu c_{NS})(P(M)|N_{adv}|S)|\\
&(\nu c_{PN})(\nu c_{NS})(N(E)_{adv}|S)
\end{aligned}
$$

The compromised notification service N_{adv} replay or delays messages from the Publisher to the Subscribers using the message channel from N to S (c_{NS}), and the subscriber IEDs receive the message to perform certain actions or not. When an event occurs ($N(E)_{adv}$), the replay attack can be performed continuously during message retransmissions to cause denial of service for the subscriber IEDs.

The attacker may also reorder the messages to cause disruption by swapping the order of messages that the publisher sends to the subscribers. For example, if the publisher sends messages ($M_i = m_1, m_2, m_3$), an attacker at the notification service can reorder this message and send to the subscribers in different orders (e.g., $M_j = m2, m_1, m_3$) while continuing to maintain stealthiness. A π-calculus description of message reordering attack can be

given as follows:

$$
\begin{aligned}
&if(t = T_0)\\
&P(M_i) = \overline{c_{PN}}\langle M_i\rangle\\
&N(M_j)_{adv} = c_{PN}(x).\overline{c_{NS}}\langle x\rangle\\
&S = c_{NS}(x).x(y)\\
&Inst(M)_{adv} = (\nu c_{PN})(\nu c_{NS})(P(M_i)|N(M_j)adv|S)\\
&else\\
&for(t = T_i(i = 1)); t \leq T_{stableCondition}; t++)\\
&N(E)_{adv} = \overline{c_{PN}}\langle E\rangle\\
&S = c_{NS}(x).x(z)\\
&endfor\\
&Inst(M)_{adv} = (\nu c_{PN})(\nu c_{NS})(P(M_i)|N(M_j)_{adv}|S)|\\
&(\nu c_{PN})(\nu c_{NS})(N(E)_{adv}|S)
\end{aligned}
$$

The *undetectability* property can be formalised as *observational equivalence* [10]. It is a notion that allows expressing flexible notions of security properties by requiring observational equivalence between *a protocol* and *an idealized version of it*, that realizes the desired properties. In the message reordering attack, the publisher P is as usual, but the notification service N is replaced with a variant N_{adv} that intercept and reorder the messages to send to the subscribers. A simplified π-calculus description of the protocol instance and its modified variant with the message reordering attack is given as follows:

$$
\begin{aligned}
P(M_i) &= \overline{c_{PN}}\langle M_i\rangle\\
N(M_i) &= c_{PN}(x).\overline{c_{NS}}\langle x\rangle\\
S &= c_{NS}(x).x(y)\\
Inst(M_i) &= (\nu c_{PN})(\nu c_{NS})(P(M_i)|N(M_i)|S)
\end{aligned}
$$

The modified protocol instance with the message reordering

$$
\begin{aligned}
P(M_i) &= \overline{c_{PN}}\langle M_i\rangle\\
N(M_j)_{adv} &= c_{PN}(x).\overline{c_{NS}}\langle x\rangle\\
S &= c_{NS}(x).x(y)\\
Inst(M_i)_{adv} &= (\nu c_{PN})(\nu c_{NS})(P(M_i)|N(M_j)adv|S)
\end{aligned}
$$

The undetectability property can be stated in terms of equivalences: if $N(M_i) \simeq N(M_j)_{adv}$, for any M_i, M_j, then $Inst(M_i) \simeq Inst(M_i)_{adv}$. This means that if

$N(M_i)$ is indistinguishable from $N(M_j)_{adv}$, then the protocol instance with message M_i is indistinguishable from the protocol instance with message M_j (the protocol instance with the reordering message attack). In this attack, it is not necessary to add attack vectors as it does not require modification or bad data injection into the intermediate nodes. However, we assume that all data is subjected to outlier removal or detection which is usually a *residue test* to filter bad data. Thus, the reordering attack should be stealthy to circumvent the detection test. This is to maximize the impact I of swapping while keeping the message reordering to the minimum, and an optimization problem can be formulated as $\min_{M_j} I \quad s.t. \quad ||M_j|| \leq \mu$, where $\mu > 0$ is the desired bound on the size of attack that the bad data detection test is not triggered.

4.6 Conclusion and Future Work

Networked critical infrastructures should be designed with resilience in mind and an understanding of how adversarial actions may affect the communication protocols deploy in such systems is an essential step in that direction. We have noted that the conventional adversary models are not suitable for IEC 61850 environment and thus, there is a need for adversary model that captures the stringent QoS constraints and the network topology. Also, we presented using π-calculus variant the limitations placed on the attacker and using this understanding, we described a replay attack that can result in a DoS attack, to show the application of our model.

Future work will include modelling of timing attacks against IEC 61850 real-time communication protocols using our model, so as to provide the basis for a resilient mechanism to mitigating such attacks. The attack models would not only help in understanding and mitigating attacks against IEC 61850 real-time communication protocols but also may be used for other protocols that share similar characteristics.

4.7 Bibliography

[1] DOLEV, D. AND YAO, A. On the security of public key protocols *IEEE Transactions on Information Theory*, 1983, 29, 198-208 16, 78, 84, 88, 137

[2] MCEVOY, T. R. AND WOLTHUSEN, S. D. A formal adversary capability model for SCADA environments. *International Workshop on Critical Information Infrastructures Security*, 2010, 93-103 2, 8, 16, 79, 84

[3] BERNAT, G.; BURNS, A. AND LIAMOSI, A. Weakly hard real-time systems . *IEEE Transactions on Computers*, 2001, 50, 308-321 79

[4] TEBEKAEMI, E. AND WIJESEKERA, D. Designing an IEC 61850 based power distribution substation simulation/emulation testbed for cyber-physical security studies. In *Proceedings of the First International Conference on Cyber-Technologies and Cyber-Systems*, 2016, 41-49 80

[5] BRAND, K.-P. AND WIMMER, W. Special Report IEC 61850. *ABB*, 2010 80, 81

[6] YOUSSEF, T. A.; HARIRI, M. E.; BUGAY, N. AND MOHAMMED, O. A. IEC 61850: Technology standards and cyber-threats. In *Proc. IEEE 16th Int. Conf. Environment and Electrical Engineering (EEEIC)*, 2016, 1-6 81

[7] STROBEL, M.; WIEDERMANN, N. AND ECKERT, C. Novel weaknesses in IEC 62351 protected Smart Grid control systems. *Proc. IEEE Int. Conf. Smart Grid Communications (SmartGridComm)*, 2016, 266-270 81, 86, 91

[8] OZANSOY, C. R.; ZAYEGH, A. AND KALAM, A. The Real-Time Publisher/Subscriber Communication Model for Distributed Substation Systems. *IEEE Transactions on Power Delivery*, 2007, 22, 1411-1423 81, 82

[9] CONG, W.; PAN, Z.; GAO, Z.; ZENG, Y. AND ZHAI, Y. Communication service model for wide area protection system based on IEC 61850. *Transactions of Tianjin University*, Springer, 2008, 14, 226-230 81

[10] SANGIORGI, D. AND WALKER, D. The pi-calculus: a Theory of Mobile Processes. *Cambridge university press*, 2003 9, 82, 93, 122, 140, 159

[11] AKELLA, R. AND MCMILLIN, B. M. Modeling and Verification of Security Properties for Critical Infrastructure Protection. In *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, ACM, 2013, 6:1-6:5 84

[12] VIGO, R. The cyber-physical attacker. In *International Conference on Computer Safety, Reliability, and Security*, 2012, 347-356 16, 84

[13] OROJLOO, H. AND AZGOMI, M. A. A method for modeling and evaluation of the security of cyber-physical systems. In *2014 11th International ISC Conference on Information Security and Cryptology*, 2014, 131-136 16, 84

[14] ADEPU, S. AND MATHUR, A. Generalized attacker and attack models for cyber physical systems. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, 2016, 1, 283-292 16, 18, 84, 125

[15] GÓES, R. M.; KANG, E.; KWONG, R. AND LAFORTUNE, S. Stealthy deception attacks for cyber-physical systems. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, 4224-4230 16, 85

[16] LANOTTE, R.; MERRO, M.; MURADORE, R. AND VIGANÒ, L. A formal approach to cyber-physical attacks . In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, 2017, 436-450 15, 16, 17, 18, 19, 85, 122, 126, 129, 148, 149, 150, 155, 157

[17] HOYOS, J.; DEHUS, M. AND BROWN, T. X. Exploiting the GOOSE protocol: A practical attack on cyber-infrastructure. In *Proc. IEEE Globecom Workshops*, 2012, 1508-1513 85

[18] SILVA, L. E. D. AND COURY, D. V. A new methodology for real-time detection of attacks in IEC 61850-based systems. *Electric Power Systems Research*, Elsevier, 2017, 143, 825-833 85

[19] KABIR-QUERREC, M.; MOCANU, S.; BELLEMAIN, P.; THIRIET, J.-M. AND SAVARY, E. Corrupted GOOSE Detectors: Anomaly Detection in Power Utility Real-Time Ethernet Communications. *GreHack 2015*, 2015 85

[20] NOCE, J.; LOPES, Y.; FERNANDES, N. C.; ALBUQUERQUE, C. V. N. AND MUCHALUAT-SAADE, D. C. Identifying vulnerabilities in smart gric communication networks of electrical substations using GEESE 2.0. In *Proc. IEEE 26th Int. Symp. Industrial Electronics (ISIE)*, 2017, 111-116 85

[21] WRIGHT, J. G. AND WOLTHUSEN, S. D Stealthy Injection Attacks Against IEC61850's GOOSE Messaging Service. In *Proc. IEEE PES Innovative Smart Grid Technologies Conf. Europe (ISGT-Europe)*, 2018, 1-6 53, 54, 56, 57, 85

[22] CHLELA, M.; JOOS, G.; KASSOUF, M. AND BRISSETTE, Y. Real-time testing platform for microgrid controllers against false data injection cybersecurity attacks. In *Proc. IEEE Power and Energy Society General Meeting (PESGM)*, 2016, 1-5 85

[23] KABIR-QUERREC, M.; MOCANU, S.; THIRIET, J. AND SAVARY, E. A Test bed dedicated to the Study of Vulnerabilities in IEC 61850 Power Utility Automation Networks. In *21st IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2016*, IEEE, 2016, 1-4 85

[24] CHATTOPADHYAY, A.; UKIL, A.; JAP, D. AND BHASIN, S. Toward Threat of Implementation Attacks on Substation Security: Case Study on Fault Detection and Isolation. *IEEE Transactions on Industrial Informatics*, 2018, 14, 2442-2451 85

[25] ZHANG, F.; MAHLER, M. AND LI, Q. Flooding attacks against secure time-critical communications in the power grid. In *Proc. IEEE Int. Conf. Smart Grid Communications (SmartGridComm)*, 2017, 449-454 86

[26] LI, Q.; ROSS, C.; YANG, J.; DI, J.; BALDA, J. C. AND MANTOOTH, H. A. The effects of flooding attacks on time-critical communications in the smart grid. In *Proc. IEEE Power Energy Society Innovative Smart Grid Technologies Conf. (ISGT)*, 2015, 1-5 86

[27] ELBEZ, G.; KELLER, H. B. AND HAGENMEYER, V. A Cost-efficient Software Testbed for Cyber-Physical Security in IEC 61850-based Substations. In *Proc. and Computing Technologies for Smart Grids (SmartGridComm) 2018 IEEE Int. Conf. Communications, Control*, 2018, 1-6 86

[28] HONG, J.; LIU, C. AND GOVINDARASU, M. Detection of cyber intrusions using network-based multicast messages for substation automation. In *Proc. ISGT 2014*, 2014, 1-5 86

[29] ZHANG, Z.; GONG, S.; DIMITROVSKI, A. D. AND LI, H. Time synchronization attack in smart grid: Impact and analysis. *IEEE Transactions on Smart Grid*, IEEE, 2013, 4, 87-98 86

[30] BARRETO, S.; SURESH, A. AND LE BOUDEC, J. Cyber-attack on packet-based time synchronization protocols: The undetectable Delay Box. In *Proc. IEEE Int Instrumentation and Measurement Technology*, 2016, 1-6 86

[31] MOUSSA, B.; DEBBABI, M. AND ASSI, C. A Detection and Mitigation Model for PTP Delay Attack in an IEC 61850 Substation. *IEEE Transactions on Smart Grid*, 2018, 9, 3954-3965 86

[32] MOUSSA, B.; ROBILLARD, C.; ZUGENMAIER, A.; KASSOUF, M.; DEBBABI, M. AND ASSI, C. Securing the Precision Time Protocol (PTP) Against Fake Timestamps. *IEEE Communications Letters*, 2019, 23, 278-281 86

[33] DECUSATIS, C.; LYNCH, R. M.; KLUGE, W.; HOUSTON, J.; WOJCIAK, P. AND GUENDERT, S. Impact of Cyberattacks on Precision Time Protocol. *IEEE Transactions on Instrumentation and Measurement*, 2019, 1 86

[34] DO, Q.; MARTINI, B. AND CHOO, K.-K. R. The role of the adversary model in applied security research. *Computers and Security*, Elsevier, 2018 88

[35] UZUNOV, A. V. A Survey of Security Solutions for Distributed Publish/Subscribe Systems. *Comput. Secur.*, Elsevier Advanced Technology Publications, 2016, 61, 94-129 89

[36] SRIVATSA, M.; LIU, L. AND IYENGAR, A. EventGuard: A System Architecture for Securing Publish-Subscribe Networks. *ACM Trans. Comput. Syst.*, ACM, 2011, 29, 10:1-10:40 89

[37] Esposito, C. and Ciampi, M. On Security in Publish/Subscribe Services: A Survey. *IEEE Communications Surveys Tutorials*, 2015, 17, 966-997 89

[38] Wun, A.; Cheung, A. and Jacobsen, H.-A. A Taxonomy for Denial of Service Attacks in Content-based Publish/Subscribe Systems. In *Proceedings of the 2007 Inaugural International Conference on Distributed Event-based Systems*, ACM, 2007, 116-127 89

[39] Saxena, N.; Grijalva, S. and Choi, B. J. Securing restricted publisher-subscriber communications in smart grid substations. In *Proc. 10th Int. Conf. Communication Systems Networks (COMSNETS)*, 2018, 364-371 91

Chapter 5

A Review of Asset-Centric Threat Modelling Approaches

A Review of Asset-Centric Threat Modelling Approaches

International Journal of Advanced Computer Science and Applications (IJACSA), 11(2), 2020

Livinus Obiora Nweke and Stephen D. Wolthusen

Abstract

The threat landscape is constantly evolving. As attackers continue to evolve and seek better methods of compromising a system; in the same way, defenders continue to evolve and seek better methods of protecting a system. Threats are events that could cause harm to the confidentiality, integrity, or availability of information systems, through unauthorized disclosure, misuse, alteration, or destruction of information or information system. The process of developing and applying a representation of those threats, to understand the possibility of the threats being realized is referred to as threat modelling. Threat modelling approaches provide defenders with a tool to characterize potential threats systematically. They include the prioritization of threats and mitigation based on probabilities of the threats being realized, the business impacts and the cost of countermeasures. In this paper, we provide a review of asset-centric threat modelling approaches. These are threat modelling techniques that focus on the assets of the system being threat modelled. First, we discuss the most widely used asset-centric threat modelling approaches. Then, we present a gap analysis of these methods. Finally, we examine the features of asset-centric threat modelling approaches with a discussion on their similarities and differences.

5.1 Introduction

Threats are events that could cause harm to the confidentiality, integrity, or availability (CIA model [1]) of information systems, through unauthorized disclosure, misuse, alteration, or destruction of information or information system [5]. The process of developing and applying a representation of those threats, to understand the possibility of the threats being realized

is referred to as threat modelling. It includes selecting a threat modelling framework and populating that framework with specific values (e.g. adversary expertise, attack patterns and attack events) as relevant to the intended scope (e.g. architectural layers or stakeholder concerns). The populated framework can then be used to construct threat scenarios; characterize controls, technologies, or research efforts; and/to share threat information and responses [2].

Threat modelling methodologies are very few; although there are several frameworks or threat classification models that are usually combined and leveraged by threat modelling methodologies [4]. The choice of threat modelling approach to adopt for a particular situation is dependent on the business objectives. It follows that the first step towards choosing the threat modelling technique to use for a system is to have a clear understanding of what the system being threat modelled is supposed to do. Basically, there are no good or bad threat modelling methods but rather, there are good and bad threat modelling approaches for a particular system.

There are three main approaches that are usually deployed for threat modelling activities and they include: the approaches that focus on the assets of the system being threat modelled, which are referred to as asset-centric threat modelling approaches; the approaches that focus on the attackers, also called the attack-centric threat modelling approaches; and the approaches that focus on the software or the system, which are referred to as software-centric or system-centric threat modelling approaches [6]. We are mainly concern with the asset-centric threat modelling approaches in this paper.

In this paper, we provide a review of asset-centric threat modelling approaches. First, we examine the general objectives and benefits of threat modelling. Also, we present a discussion on the existing threat modelling approaches and justification for reviewing asset-centric threat modelling approaches. We observe that DREAD (damage, reproducibility, exploitability, affected users, discoverability), Trike, OCTAVE (operationally threat asset, and vulnerability evaluation) and PASTA (process for attack simulation and threat analysis) are the most widely used asset-centric threat modelling approaches. The limitation of these approaches is presented. We also examine the features of the asset-centric threat modelling approaches. And using these features, we present a discussion on their similarities and differences. The overall goal of this review is to serve as a foundation for selecting asset-centric threat modelling approaches and to further advance the use of asset-centric methodologies in threat modelling activities.

The rest of this paper is organised as follows. Section 5.2 examines the general objectives and benefits of threat modelling. Also, a discussion on the existing threat modelling approaches is presented with the justification for reviewing asset-centric threat modelling approaches. Section 5.3 presents

state-of-the-art of the most widely used asset-centric threat modelling approaches. Section 5.4 presents gap analysis of the asset-centric threat modelling approaches reviewed. Section 5.5 discusses the similarities and differences of the asset-centric threat modelling approaches based on their features. Section 5.6 concludes the paper and present future work.

5.2 Background

In this section, we examine the general objectives and benefits of threat modelling. We also present a discussion on the existing threat modelling approaches and justification for reviewing the state-of-the art in the asset-centric threat modelling approaches in this paper.

5.2.1 Threat Modelling

Threat modelling is a systematic approach for characterizing potential threats to a system. It ensures completeness by including the prioritization of threats and mitigation based on probabilities, business impacts and cost of countermeasures. Threat modelling provides a means of evaluating all possible risks throughout the system and not just concentrating on where flaws are expected to be discovered [7]. It is also useful in ranking the likelihood of a threat being realized. An essential step for threat modelling is having an understanding of assets and threats [4].

Assets are usually discrete data entities, but they can be physical objects, which feature in the business rules of a system [7]. Assets are artefacts which are important to a specific problem domain of a system, and not just in the actual implementation of a system. Identifying assets can be a very challenging endeavour, but it is the initial step that needs to be carried out in order to understand the amount of resource which can be allocated for threat modelling activities. Also, the amount of threats increases geometrically as the number of assets increases [7].

UcedaVelez and Morana [4] observe that most organizations, businesses, and governments depend on sources such as threat intelligence for the acquisition of threat knowledge. It is obvious that threats would mean different things to different types of organizations. For instance, in the case of private organization, potential threats are those targeting their business assets. For government organizations, potential threats are those relating to national security. Analysing the potential threat scenarios that target an organization's assets is important in determining the likelihood of the threats being realized.

Once the analysis of the potential threat scenarios has been concluded and it shows that the system being threat modelled is at risk, the next step of the risk mitigation strategy is to determine if similar assets are also ex-

posed and can be affected [4]. Also, it is important to consider whether the mitigation measures suggested are able to eliminate the risk to the system without creating additional security threats. This ensures a wholistic mitigation measures are adopted to reduce the business impact of the threat being realized.

Another important factor to consider during threat modelling is the business impact of a threat being realized. A business impact is different from information security risk in that it measures the economic impact caused by either the loss or the compromise of an asset while information security risk affects the confidentiality, integrity and availability of data [4]. Determining the business impact requires a consideration for the business context in which the system operates. This can be achieved by examining at a high level, the assets of the system and the functionality the system provides based on these assets.

In general, threat modelling involves a great amount of effort and resources of so many individuals beyond those of information security [4]. It encourages collaboration and as such, the threat modelling methodology that should be deployed for a particular system may have to consider how collaboration can be fostered. The next subsection presents the different threat modelling approaches. We agree with the authors in [4] that none of these approaches are flawed but rather the way in which they are selected may be flawed.

5.2.2 Threat Modelling Approaches

Threat modelling approaches can be categorized according to the focus of the approaches. These approaches include those that focus on the assets of the system being threat modelled, which are referred to as asset-centric threat modelling approaches; the approaches that focus on the attackers, also called attack-centric threat modelling approaches; and the approaches that focus on the software or the system, which are referred to as software-centric or system-centric threat modelling approaches [6]. Deciding which of the method to deploy depends on the system being threat modelled and the tools available.

Asset-centric threat modelling approaches focus on the assets of the system being threat modelled. It involves analysing the information loss or business impact of targeted assets. Asset-centric threat modelling can be extended beyond identifying the motives and intentions of the attacker to incorporating the discovery of security gaps for the system environment [4]. Although, asset-centric threat modelling is not concerned about flaws or insecure coding/design practices, it could be used to uncover possible threats scenarios.

Attack-centric threat modelling approaches include those approaches that focus on the attacker. The idea here is to examine the threats against a sys-

tem from the perspective of an attacker. Attack-centric threat modelling approach aims to identify which threats can be successfully executed against a system given a number of identified misuse cases, vulnerabilities, and more [4]. Also, the approach attempts to examine the motive, sources and relative identity of the attacker or group associated with the attacker as these can help to uncover the approach and resources of the attacker [4].

System-centric threat modelling approaches focus on the system being threat modelled. They first consider the design model of the system under consideration. The objective of these approaches is to ensure that the complexity of the system being threat modelled is well understood before considering threats the system may be exposed to. System-centric threat modelling approaches expects those involved in threat modelling of a system, to have a good grasp of the system they are developing [6].

In this paper, we interested in understanding the state-of-the-art in asset-centric threat modelling approaches. It is usually the case that most businesses have a clear understanding of their business objectives and assets to be protected. Also, the system to be threat modelled and the business impacts of threats being realized are likely to be known. Thus, the obvious threat modelling approaches that can be employed for the protection of assets, understanding and managing business risks for most businesses are the asset-centric threat modelling approaches. Therefore, we present this review to serve as a basis for selecting or combining the appropriate asset-centric threat modelling approaches and to further advance the use of asset-centric threat modelling techniques.

5.3 The State-of-the-Art in Asset-Centric Threat Modelling Approaches

In this section, we present a review of asset-centric threat modelling approaches. We observe that the most widely used asset-centric threat modelling approaches are DREAD, Trike, OCTAVE, and PASTA. We use this understanding to present a discussion on the state-of-the-art in asset-centric threat modelling approaches.

5.3.1 DREAD

DREAD is an acronym for Damage potential, Reproducibility, Exploitability, Affected users, and Discoverability. It is an asset-centric threat modelling approach developed by Microsoft. DREAD uses the traditional qualitative risk rating (HIGH, MEDIUM, LOW) with a qualitative risk rating 3,2,1 applied respectively. In general, DREAD threat modelling approach uses a scoring system to calculate the probability of occurrence for each of the identified areas of the asset being threat modelled. By combining the risk rating

values obtained, DREAD threat modelling approach is able to predict the probability of occurrence of each threat identified during the threat modelling process [4].

The Damage potential refers to the level of havoc that could be done to users and the organization if an attack were to succeed. Damage could be concrete, such as financial liability or abstract, such as damage to organization's reputation. Also, it depends on the nature of the attack and the assets being targeted. Reproducibility measures the easy with which the attack can be replicated. The goal is to measure the effort that would be expended by an attacker for the realization of an attack and use such measure, in the scoring system. If an attack can be reproduced with much ease, the attack would be rated high in the scoring system as against an attack that cannot be reproduced with much ease.

The remaining letters of DREAD are described as follows. Exploitability describes the possibility of an attacker taking advantage of a vulnerability. Several exploits exist and they can be classified as those that are easily understood and could be accomplished by anyone and those that are difficult that required specialized skills to achieve. This understanding is used to rate threat that have high level of exploitability as high risk in the scoring system and those with low level of exploitability as low risk. Affected users refers to the number of users that will be affected by the realization of a particular threat. A threat that is likely to affect a great number of users when realized would have a higher risk factor rating compared to a threat that is likely to affect a limited number of users. Discoverability describes the ease with which the vulnerability is uncovered. There are threat that are very difficult to learn and those that can be learn with ease. Hence, a threat that is very difficult to learn would be rated lower than those that has been released in the public domain. The DREAD approach is summarised in figure 5.1.

Although DREAD is an asset-centric threat modelling approach, several of its application in the literature is in combination with STRIDE (spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege) model [10, 11, 20, 17, 12]. In this type of approach, the DREAD scoring scheme is used to identify the likelihood that an attack is able to exploit a particular threat.

5.3.2 Trike

Trike offers a threat modelling approach which is asset-centric and it achieves that through the generation of threat models in a reliable, and repeatable manner [7]. It facilitates constructive interaction among relevant stakeholders by providing standardized framework for reasoning about threats that a system would have to overcome. The achievement of Trike objectives entail the following [7]:

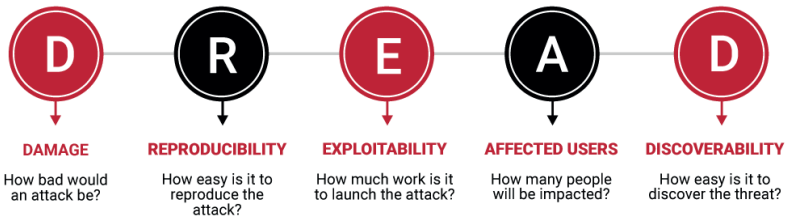


Figure 5.1: DREAD Summary

[8]

- With assistance from the system stakeholders, ensure that the risk the system presents to each asset is acceptable to all stakeholders.
- Be able to tell whether that have been done.
- Communicate what have been done and it effects to the stakeholders.
- Empower stakeholders to understand and reduce the risks to themselves and other stakeholders implied by their actions within their domains.

Another important observation about Trike is that it follows a defensive approach. Understanding the system itself and the environment in which the system is going to be used is more important when using Trike threat modelling approach than understanding the capability of an attacker. This is because without a complete knowledge of the system, it is difficult to appropriately characterize the threat that a system would have to face [7].

5.3.3 OCTAVE

OCTAVE is another asset-centric threat modelling approach. It is an acronym for Operationally Threat Asset, and Vulnerability Evaluation. The OCTAVE methodology takes the advantage of people's understanding of their organization's security-related practices and process to model the state-of-the-art of security practice within the organization. Threat to the most critical assets are used to prioritize areas of improvement and to set security strategy for the organization [14].

The two aspects that are the foundation of OCTAVE approach include: operational risk and security practices. The security practices encompasses efforts by an organization to refine its existing security practices. Technologies deployed by an organization in meeting its business objectives are evaluated in relation to security practices. For the operational risks, an organization considers all aspects of risk (asset, threats, vulnerabilities, and organization impact) in its decision making enabling the organization to match a practice-based protection strategy to its security risks [14]. The OCTAVE process is depicted in figure 5.2.

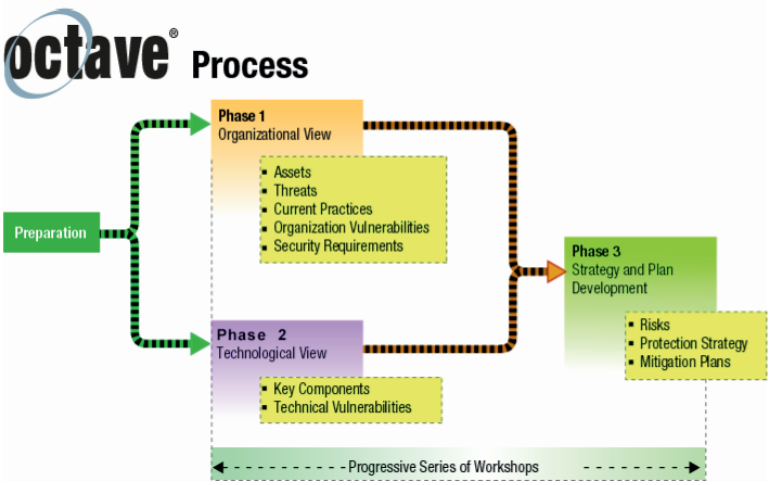


Figure 5.2: OCTAVE Process

[14]

The evaluation process of OCTAVE approach involves the following [14]:

- Identify information-related assets that are important to the organization;
- focus risk analysis activities on those assets judged to be most critical to the organization;
- consider the relationships among critical assets, the threat to those assets, and vulnerabilities (both organization and technological) that can expose assets to threats;
- evaluate risks in an operational context, i.e. how they are used to conduct an organization's business and how those assets are at risk due to security threats;

- create a practice-base protection strategy for organizational improvement as well as risk mitigation plans to reduce the risk to the organization's critical assets.

In addition, the evaluation process for the organizational, technological, and analysis aspects are complemented by a three-phased approach, namely build asset-based threat profiles; identify infrastructure vulnerabilities, and develop security strategy and plans [14].

It is also imperative to note that the essential elements or requirements of the OCTAVE approach are captures in a set of criteria [14]. As of now, there are three methods consistent with the criteria and they are: the OCTAVE Method, that is designed for large organization; the OCTAVE-S, which is well-suited for small organizations; and the most recent version called the OCTAVE Allegro. The OCTAVE Allegro has been applied in [3] to evaluate the security risks of IoT (Internet of things) based smart homes. The authors in [15] developed a university information security risk management framework using OCTAVE Method based on ISO/EIC 27001:2013. Also, the OCTAVE-S has been combined with ISO 27001:2005 in [18] for risk management.

5.3.4 PASTA

PASTA is an acronym for Process for Attack Simulation and Threat Analysis which is an asset-centric threat modelling approach. It combines topicality, substantiation, and probabilistic analysis as the key three attributes as part of its methodology [4]. According to UcedaVelez and Morana [4], PASTA approach can be deployed in almost any scenario except for those scenarios where executive sponsorship of its process and produced artefacts is not available. This is because the deliverables produced by the PASTA approach are supposed to be familiarized with the organization's executives too.

When adopting and executing PASTA threat modelling approach, it is essential to review the following: sponsorship and support (without executive support the process will not succeed); maturity, as the maturity of the processes and controls employed will affect the outcome of PASTA; awareness, efficient and effective communication is required for the entire activities; input and outputs, people are the main input to consider for the threat modelling activity and outputs are to be defined for each process involved in the threat modelling; and lastly participants are recruited and retrain [4].

For the actual deployment, the PASTA threat modelling methodology include the following stages. The first stage involves defining objectives, where the business objectives of the system to be threat modelled is clearly defined. The technological scope is defined in the second stage and it involves identifying all the assets of the system. Next, the system is decomposed to facilitate an understanding of the system's operations. In the fourth

stage, threat analysis is carried out to identify threats to the system. Then, weakness and vulnerability analysis which allows vulnerable areas across the system to be identified and mapped to the attack tree introduced in the threat analysis stage. Attack modelling and simulation is followed and the focus is to study the possibility that the identified vulnerabilities can be exploited. Lastly, residual risk analysis and management is done to mitigate threat that are major concerns to the system. All these stages are shown in figure 5.3.

Figure 5.3: PASTA Stages

[4]

5.4 Limitation of the Asset-Centric Threat Modelling Approaches

In this section, we present a gap analysis of the asset-centric threat modelling approaches discussed in section 5.3.

5.4.0.1 DREAD

has been shown to be fairly subjective and leads to inconsistent results [2]. In fact, as of 2010, Microsoft discontinued the use of DREAD for their software development life-cycle [2]. This further underscores the limitation of DREAD as a threat modelling approach. However, DREAD is still widely used and recommended for threat and risk modelling endeavours. Hence, useful suggestions have been made in [9] on modifications to the scoring scheme in order to improve its reproducibility.

5.4.0.2 Trike

requires an analyst undertaking a threat modelling exercise to have full a grasp of the whole system while assessing the risk of attacks. This can be very challenging if the system to be threat modelled is very large. Also, the authors in [13] observed that the Trike scoring system is too vague to represent a formal. In addition, Trike does not have sufficient documentation even though its website is still available.

5.4.0.3 OCTAVE

is a robust, asset-centric threat modelling approach but it is highly complex. It takes considerable time to learn and the processes involved can be time consuming. Also, OCTAVE documentation can become voluminous, which is likely to discourage policy makers from adopting it as a threat modelling approach for their organization.

Another limitation of OCTAVE threat modelling approach is the way in which the identification and classification of threat is achieved. The capturing of risks and threats using the threat tree when OCTAVE is employed can become undesirable for complex environment. As the number of paths increases in the case of a very large computing environment, it may become unclear which of the paths represent the threats being modelled.

5.4.0.4 PASTA

is design for organizations that desire to position threat modelling with their strategic objectives. This is because PASTA incorporates business impact analysis as an important part of the PASTA process, which extends security responsibilities to the entire organization. This positioning can become a

drawback for using PASTA because it may require several hours of training and education of the key stakeholders.

5.5 Discussion

This section presents a discussion on the similarities and differences of the asset-centric threat modelling approaches we have presented so far. First, the features of the asset-centric threat modelling approaches are given in table 5.1. We then provide a discussion on their similarities and differences.

A feature that is common to all the asset-centric threat modelling approaches as can be observed from table 5.1, is the fact that they all contribute to risk management process. In fact, asset-centric threat modelling approaches are sometimes referred to as risk-based threat modelling approaches [4]. They employ a risk-based approach in analysing the business impact of possible threat scenarios. This can then be used to prioritize threat mitigation strategies, which is also a feature that all the asset-centric threat modelling approaches we have presented in this paper possesses.

Apart from DREAD, the remaining asset-centric threat modelling approaches encourage collaboration among the stakeholders and can be used to identify relevant mitigation techniques. Collaboration is an essential part of any threat modelling activities. Considering that majority of the asset-centric threat modelling approaches presented in this review encourage collaboration among relevant stakeholders further buttress the importance of collaboration during threat modelling process. Mitigation techniques ensures that actionable steps which can help to avoid the threats identified during the threat modelling process are recommended.

Another important desirable characteristics of any threat modelling approach are reproducibility and automation. Reproducibility refers to the ability of the threat modelling approach to have consistent results when repeated. Unfortunately, the only asset-centric threat modelling approach that seems to have such property is OCTAVE. Other approaches are usually subjective and depend on those carrying out the threat modelling activities. Automation ensures that the threat modelling process can be undertaken without human intervention. As of now, only Trike has automated components and given the insufficient documentation there is still a lot of work to be done in automating asset-centric threat modelling approaches.

5.6 Conclusion

Asset-centric threat modelling approaches have shown to be effective for the protection of assets, understanding and managing business risks. In this paper, we have reviewed the state-of-the-art in asset-centric threat modelling approaches. We have observed that DREAD, Trike, OCTAVE, and PASTA

Table 5.1: Features of Asset-Centric Threat Modelling Approaches

Asset-centric Threat Modelling Approach	Features
DREAD	• Helps to assess risk associated with a threat exploit • Can predict the probability of an exploit being realized • Contributes to risk management • Has built-in prioritization of threat mitigation • Offers flexibility and can be applied and adopted to any situation
Trike	• Encourages collaboration among stakeholders • Has built-in prioritization of threat mitigation • Has automated components • Contributes to risk management • Can identify mitigation techniques
OCTAVE	• Encourages collaboration among stakeholders • Has built-in prioritization of threat mitigation • Has consistent results when repeated • It is designed to be scalable • Contributes to risk management • Can identify mitigation techniques
PASTA	• Encourages collaboration among stakeholders • Has built-in prioritization of threat mitigation • Contributes to risk management • Can identify mitigation techniques.

are the most widely used asset-centric threat modelling approaches. Then, we present a discussion on the state-of-the-art of these approaches. Also, a gap analysis of these approaches is discussed. Finally, we describe the features of the asset-centric threat modelling approaches we have reviewed, with a discussion on their similarities and differences.

In the future, we hope to explore formal methods that can exploit asset-centric threat modelling approach to reason about the potential threats to a cyber-physical system. This is because the asset-centric threat modelling approaches we have reviewed in this paper are not suitable for capturing the potential threats to a cyber-physical system due to the timing, uncertainty, and dependencies that exist between its entities. Although, several attempts have been made in the literature to threat model cyber-physical systems [16, 19, 21], we intend to use the formal method for expressing the requirements that are unique to a cyber-physical system in order to facilitate the identification of potential threats to the system.

5.7 Bibliography

[1] NWEKE, L. O. Using the CIA and AAA Models to Explain Cybersecurity Activies. In *PM World Journal*, 2017, 6 100

[2] BODEAU, D. J.; MCCOLLUM, C. D. AND FOX, D. B. Cyber Threat Modeling: Survey, Assessment, and Representative Framework. *The Homeland Security Systems Engineering and Development Institute*, 2018 101, 110

[3] ALI, B. AND AWAD, A. I. Cyber and Physical Security Vulnerability Assessment for IoT-Based Smart Homes. *Sensors*, MDPI AG, 2018, 18, 817 108

[4] UCEDAVELEZ, T. AND MORANA, M. M. Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis. *John Wiley and Sons*, 2015 101, 102, 103, 104, 105, 108, 109, 111, 119

[5] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST). Information Security: Guide for Conducting Risk Assessments *NIST Special Publication 800-30*, 2012 100, 119

[6] SHOSTACK, A. Threat Modeling: Designing for Security. *John Wiley and Sons*, 2014 101, 103, 104, 119

[7] EDDINGTON, M.; LARCOM, B. AND SAITTA, E. Trike v1 Methodology Document . 2005 102, 105, 106

[8] WILDCARD CORP Threat Modeling. 2019. Available from: `https://wildcardcorp.com/security/threat-modeling` 106

[9] Leblanc, D. DREADful. 2007. Available from: https://blogs.msdn.microsoft.com/david-leblanc/2007/08/14/dreadful/ 110

[10] Cagnazzo, M.; Hertlein, M.; Holz, T. and Pohlmann, N. Threat modeling for mobile health systems. In *Proc. IEEE Wireless Communications and Networking Conf. Workshops (WCNCW)*, IEEE, 2018, 314-319 105

[11] Omotosho, A.; Haruna, B. A. and Olaniyi, O. M. Threat Modeling of Internet of Things Health Devices. *Journal of Applied Security Research*, 2019, 14, 106-121 105

[12] Hagan, M.; Siddiqui, F. and Sezer, S. Policy-Based Security Modelling and Enforcement Approach for Emerging Embedded Architectures. In *Proc. 31st IEEE Int. System-on-Chip Conf. (SOCC)*, IEEE, 2018, 84-89 105

[13] Shevchenko, N.; Chick, T. A.; O'Riordan, P.; Scanlon, T. P. and Woody, C. Threat Modeling: a Summary of Available Methods. *Software Engineering Institute*, 2018 110

[14] Alberts, C.; Dorofee, A.; Stevens, J. and Woody, C. Introduction to the OCTAVE Approach. *Software Engineering Institute*, 2003 106, 107, 108, 120

[15] Sulistyowati, I. and Ginardi, R. H. Information Security Risk Management with Octave Method and ISO/EIC 27001: 2013 (Case Study: Airlangga University). *IPTEK Journal of Proceedings Series*, 2019, 32-38 108

[16] Fernandez, E. B. Threat Modeling in Cyber-Physical Systems. In *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, 2016 18, 113, 121, 125

[17] Abomhara, M.; Gerdes, M. and Køien, G. M. A STRIDE-Based Threat Model for Telehealth Systems. *Norsk informasjonssikkerhetskonferanse (NISK)*, 2015, 8, 82-96 105

[18] Stephanus, S. Implementation Octave-S and Iso 27001controls in Risk Management Information Systems. *ComTech: Computer, Mathematics and Engineering Applications*, 2014, 5, 685 108

[19] Rekik, M.; Gransart, C. and Berbineau, M. Cyber-physical Threats and Vulnerabilities Analysis for Train Control and Monitoring

Systems. In *Proc. Computers and Communications (ISNCC) 2018 Int. Symp. Networks*, 2018, 1-6 18, 113, 125

[20] AMINI, A.; JAMIL, N.; AHMAD, A. R. AND Z'ABA, M. R. Threat Modeling Approaches for Securing Cloud Computing. *Journal of Applied Sciences*, 2015, 15, 953-967 105

[21] ATIF, Y.; JIANG, Y.; JIANGUO, D.; JEUSFELD, M.; LINDSTRÖM, B.; ANDLER, S.; BRAX, C.; HAGLUND, D. AND LINDSTRÖM, B. Cyber-threat analysis for Cyber-Physical Systems . *Technical Report*, University of Skövde, 2018 18, 113, 125

Chapter 6

Threat Modelling of Cyber-Physical Systems Using an Applied π-Calculus

Threat Modelling of Cyber-Physical Systems Using an Applied π-Calculus

International Journal of Critical Infrastructure Protection
Volume 35, December 2021, 100466

Livinus Obiora Nweke, Goitom Kahsay Weldehawaryat and
Stephen D. Wolthusen

Abstract

Cyber-Physical Systems (CPS) are distributed systems in which the state of the physical system is generally not observable in non-trivial cases, and where state transitions of this physical system can also occur without resulting in immediate changes to observable variables. This poses challenges for the bidirectional synchronisation of the discrete cyber models and the partially continuous physical systems. Threats to CPS from cyber attacks are, however, often instantiable only where conditions on the CPS state during the attack meet certain conditions such that they drive the system state outside a desirable or safe space.

In this paper we propose an extension to an applied π-calculus in which we can capture both the behaviour of the CPS as well as modelling possible adversary behaviour. This is achieved by embedding an algebraic representation of Attack-Defence Trees (ADT) in the applied π-calculus and augmenting this by the addition of a partial ordering over the constituents of the ADT within the embedding, offering an elegant mechanism to extend ADT to ordering and time-related attacks. We illustrate the modelling approach for the case of an electrical substation fragment in which components communicate via the IEC 61850 protocol.

6.1 Introduction

Attacks against CPS are no longer theoretical concepts but are real and here before us. The attack against Natanz uranium enrichment plant in Iran, where the infamous malware known as Stuxnet escaped the digital realm and wreaked physical damage to a CPS [1] attest to that fact. Most recently,

researchers have discovered Ekans ransomware, which they observed was specifically designed to target CPS [2]. These trends call for rethinking about how we threat model a CPS considering timing, uncertainty, and dependencies that exist between its entities.

Threat modelling approaches provide a systematic way of reasoning about the potential threats to a system. Threats are events that could compromise the confidentiality, integrity, or availability of a system, through unauthorized disclosure, misuse, alteration or destruction [3]. There are three main approaches that may be employed to threat model a system and they include: approaches that are concerned with the assets of the system being threat modelled (asset-centric threat modelling); approaches that aim to understand the nature of the attackers (attack-centric threat modelling); and approaches that focus on the software or the system (software-centric or system-centric threat modelling) [4, 5]. We are interested in using an applied π-calculus to capture both the behaviour of the CPS as well as modelling possible adversary behaviour.

CPS integrate both computation and communication capabilities in order to control physical components. The computational elements are used for processing measurement values received from the physical components through the communication channels. This entails that there are several assets that make up the CPS and the use of threat modelling approach would be effective for understanding the potential threats to CPS. However, the existing threat modelling approaches are not able to capture the potential threats to CPS due to the timing, uncertainty, and dependencies that exist between the entities of CPS.

In a CPS, the measurement values obtained from the physical components are used to ascertain the state of the system. Unless these parameters and how they interact with one another are threat modelled, it will be difficult to know how threats to these assets may be exploited. For example, CPS may have a handful of critical states and if an attack is launched when the system is not in a critical state, the impact may not be adverse. However, an attack that is launched when the system is in a critical state would have a devastating effect. The question that is important for a threat modelling approach to consider is then; what is the likelihood of an attacker finding the system in a critical state in order to launch an attack to obtain an adverse impact.

The assertion we are making in this paper is that the likelihood of an attacker finding a CPS in a critical state cannot be expressed using the existing threat modelling approaches. The big risk of a more abstract approach like the existing threat modelling approaches is that there are chances of making a mistake by either being too confident that the approach is going to find the threats even though an understanding of the critical processes or states is missed or wasting a lot of resources by defending an attack that is very

unlikely. For instance, the processes in OCTAVE (operationally threat asset and vulnerability evaluation) [6] are narrative driven. They consider the assets of a system and the interaction between these assets, then employ verbal reasoning to evaluate the threats to the systems. We argue that this type of intuitive reasoning no longer suffices for CPS where timing, uncertainty, and dependencies between the entities exist.

Hence, we extend the attack-defence trees (ADT) with partial ordering to represent causality relationship and use an applied π-calculus to describe the formal model for CPS, taking into account its unique properties. We then define the semantics of the applied π-calculus using a labelled transition system to highlight the CPS interactions with the environment and to facilitate the definition of observational equivalences such as *bisimilarity*. This is to allow us to capture the potential threats to the CPS and to deduce some reasoning about the behaviour of the system. Also, to show the utility of our model, we present a use case scenario where the applied π-calculus is employed to reason about false measurement injection attack against IEC 61850 protocol. The main contributions of this paper are as follows.

- We propose an extension to an applied π-calculus in which we can capture both the behaviour of the CPS as well as modelling possible adversary behaviour.
- We use the ADT and extend it with partial ordering of events to show causality relationship. This allows us to represent not only the necessary conditions for an attack to be successful but also the sequencing of events or the way events have to be order for the attack to be successful.
- We translate the ADT with partial ordering into the applied π-calculus using the message synchronization primitives for partial ordering, which enables us to make an argument for equivalence.
- We illustrate our modelling approach for the case of an electrical substation fragment in which components communicate via the IEC 61850 protocol.

The rest of this paper is organised as follows. Section 6.2 presents a discussion on CPS, where we use smart grid system as an example of CPS. Also, we briefly describe the π-calculus that will be employed to threat model the CPS. Section 6.3 discusses the different approaches that have been utilized in the literature to threat model CPS. Section 6.4 provides justification on why the existing threat modelling approaches are not able to capture the unique properties of a CPS. Section 6.5 presents the formal model for a CPS using an applied π-calculus and threat modelling of the CPS based on the formal model. In addition, the section describes a use case scenario to show

the utility of our model. Section 6.6 concludes the paper and presents future works.

6.2 Background

In this section, we present a discussion on CPS. We use smart grid system as an example of CPS to describe a high-level architecture, where we extract some properties that are specific to CPS. Also, we briefly discuss π-calculus that will be employed to threat model the CPS.

6.2.1 Cyber-Physical Systems

CPS are systems that consist of computation, communication, and physical components. They combine computing and communication capabilities with the monitoring and control of assets in the physical domain [7]. Some of these systems are usually referred to as real-time systems with stringent quality of service (QoS) requirements. Also, the coupling of physical and cyber components entails that any malicious activity in the cyber components would have devastating effects on the physical components, which in turn may endanger the lives of humans operating the physical components. For this reason, CPS are sometimes called safety-critical systems. The application of CPS span through several domains including; power stations, power and water distribution, traffic systems, oil and gas sector, etc.

In CPS, there are several assets that makeup the system and they can be classified as follows: the cyber and control part assets, the physical assets, and the communication channel between the cyber and physical assets. The cyber assets consist of hardware, software, and data that connects to the Internet infrastructure. For the physical assets, they include sensors and actuators that monitor the physical environment. And the communication channels are assets used to send data from the physical environment to the cyber and control parts, and commands from the cyber and control parts to the sensors and actuators. A high-level description of these assets and the communication between them is shown in figure 6.1.

Figure 6.1 depicts a smart grid system which is an example of a complex CPS. In a smart grid system, the conventional electrical grid has been integrated with information communication technology (ICT). Smart grid system consists of several assets, and they can be classified as we have already done in the preceding paragraph. The cyber and control part assets include the supervisory control and data acquisition (SCADA) which facilitates the interconnection of the field devices like the sensors, actuator, etc. Also, the communication asset (Communication Network) that ensures bidirectional communication of data and signals in the smart grid, in addition to enabling interaction between the cyber and physical assets. Lastly, the physical assets

which include the transformers, power transmission networks, distribution networks, etc.

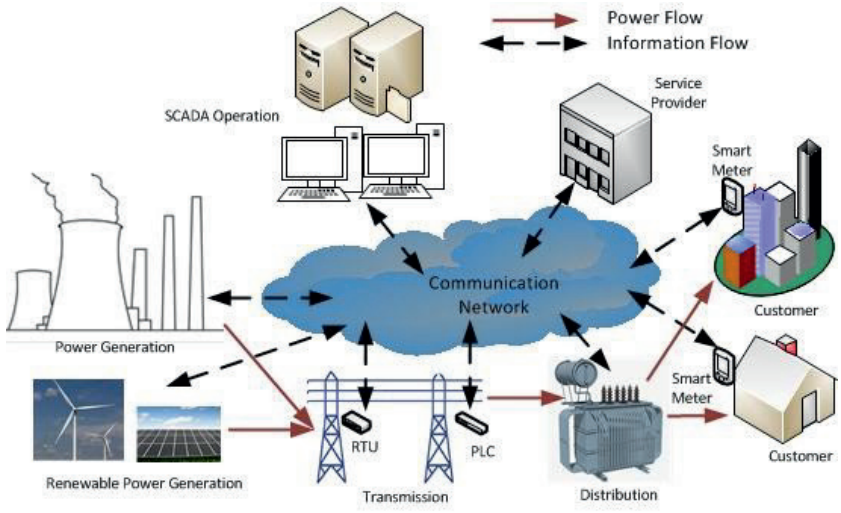


Figure 6.1: Smart Grid System

[8]

6.2.2 π-Calculus

The π-calculus is a process algebra proposed by Robin Milner [9] for describing and analysing concurrent systems with evolving communication structure. It provides a formal mechanism for modelling communication among processes over dynamic links [10] and has since been extended and applied in several studies including for modelling different types of security processes [11, 12, 13, 14]. A system in the π-calculus is made up of independent processes that communicate via channels. A channel is an abstraction of the communication link and is referred to by name. Names are the simplest entities of the π-calculus and there are infinite number of names, represented by lowercase letters (x, y, z, etc).

Processes in the π-calculus evolve by performing actions. These capabilities for action are expressed via the prefixes, of which there are four kinds:

$$\pi := \overline{x}y \mid x(z) \mid \tau \mid [x = y]\pi$$

The first capability is to send the name y via name x, and the second to receive any name via x. The third capability refers to internal action or un-

observable action. And lastly, the fourth is a conditional capability where the capability π is executed if x and y are the same. The set of processes can also be defined by the syntax given in Table 6.1.

Table 6.1: Syntax of π-calculus

Term	Semantics
$P ::=$	Processes
0	empty process
$\overline{x}z.P$	output
$x(y).P$	input
$P + Q$	choice
$P\|Q$	parallel composition
$!P$	replication
$\nu x.P$	restriction
τ	silent function/action

- A composition $P|Q$ behaves as if processes P and Q are running in parallel. This implies that the two processes can evolve separately at the same time and can operate on the channels to communicate with each other and with the outside the network.
- The basic interaction is defined using $\overline{x}z.P$ that defines an output process that is ready to output on channel x, or $x(y).P$ that defines an input process that is ready to receive a value over channel x.
- The replication $!P$ behaves as an infinite number of copies of P running in parallel.
- The name restriction operator ($\nu x.P$) is a process that makes a new, private name x, and then behaves as P.
- τ represents the internal (silent) action of a process that is not observable outside the scope of the process.
- 0 is the empty process.

To briefly describe the use of the π-calculus for modelling systems, let consider the following example which is similar to the illustration provided by Parrow in [15]. Suppose we have a system which consists of three processes, namely: a controller, a resource and an agent. The controller controls access to the resource and the agent needs access to it. We can represent the original state of the controller using a communication link x. The agent interacts with the controller via another link y to have access to the resource. After this interaction, access to the resource will be transferred to

the agent. We can express the communication among these processes using the π-calculus as follows: the controller that sends x along y is $\overline{y}x.C$; the agent that receives some link along y and then uses it to send data along it is $y(a).\overline{a}z.A$. The interaction we have described so far can be formulated in the π-calculus as follows:

$$\overline{y}x.C \,|\, y(a).\overline{a}z.A \xrightarrow{\tau} C \,|\, \overline{x}z.A$$

However, we use the extended π-calculus in this work to model CPS. As we have observed already, CPS consist of a physical component that embodies all physical aspects of a system (state variables, physical devices, etc.) and a cyber component that interacts with the physical devices (sensors and actuators) of the system and can communicate via channels with other processes of the same CPS or other CPS. The overall behaviour of the CPS is structured by the combination of the behaviours of its subsystems. Thus, in section 6.5 we use the capability of the applied π-calculus to model the message exchanges/interactions that captures the specific behaviour associated with a CPS.

6.3 Related Works

CPS security has generated a lot of attention in recent years. A significant amount of research effort has been dedicated towards the analysis, detection and identification of security issues in CPS. For example, Mo et al [16] develop a model-based techniques capable of detecting integrity attacks on the sensors of a control system. Also, Pasqualetti et al in [17] present attack detection and identification in CPS and analyse the core monitoring limitations for CPS under attack modelled by linear time-invariant descriptor systems with exogenous inputs. Several other works that have considered security issues in CPS, such as denial-of-service attacks [18, 19], replay attacks [20, 21, 22], and false data injection attacks [23, 24, 25]. However, the threat model used in most of these works employs custom construct which makes them difficult to use in different environments. Our proposed model offers a set of constructs that can be used to decompose threats in CPS.

Threat modelling of CPS has been attempted in several works in the literature. One of the earliest attempts to threat model CPS came from Zalewski et al [26]. They propose the use of a discrete time Markov chain (DTMC) model to characterize the transitions between the secure and insecure states of CPS. The authors argue that quantifying the probabilities of transitions between secure and insecure states will allow for the derivation of important inferences about the security related features of CPS. Then, the conventional threat modelling techniques (STRIDE, DREAD, CVSS) are applied in the work, to assign the probabilities of transitions between the states. These

techniques capture threats at certain level abstraction which does not allow for reasoning over the communication between assets and their timing property.

Martins et al in [27] present a tool to perform a systematic threat modelling for CPS using a real-world temperature monitoring system as a case study. The authors use the Generic Modelling Environment for the creation of domain-specific modelling for threat analysis CPS. Also, they extended and deployed Microsoft SDL Threat Modelling Tools to model, identify, and mitigate threats in a systematic way for the proposed CPS. A model to represent CPS threats using patterns that are related to architectural aspects of the CPS is described in [7]. The author shows how to extend the misuse pattern to characterise cyber-physical threats and how to enumerate and unify cyber-physical threats.

A threat modelling framework for CPS using STRIDE is presented in [28]. The authors demonstrate the applicability of the proposed framework using a real synchrophasor-based synchronous islanding test-bed in the laboratory. They show that an adversary can achieve a specific malicious goal by exploiting threats at different locations in the system. Also, they illustrate that by identifying component level vulnerabilities and their potential physical consequence, STRIDE can be applied to address such challenge. Almohri et al in [29] present threat modelling of medical CPS. The authors consider the roles of stakeholders and system components. They use this understanding to sketch an abstract architecture of medical CPS and then show the various threat modelling options.

CPS threats and vulnerabilities analysis for train control and monitoring systems is presented in [30]. The authors evaluate vulnerabilities and characteristics of railway threat landscape including potential threats, threat actors and motivations. Also, they examine the direct impacts and cascading consequences of threats on the whole system as well as risk produced. Atif et al in [31] describe cyber threat analysis for CPS. They employ data-driven approach to threat model CPS. A machine learning algorithm based on K-Nearest-Neighbour (K-NN) is used in this work, to ascertain the threat category faced by the CPS considered.

Attacker models for CPS have been discussed in [32]. The authors present a literature review of the attacker models for CPS and define a taxonomy of ten different features that they applied to the literature. Also, a generalized attacker and attack models for CPS has been proposed in [33] and have been employed to investigate the impact of single-point cyber attacks on a Secure Water Treatment (SWaT) system in [34]. Unlike these attacker and attack models presented in [32, 33, 34] where the authors utilise descriptive threat modelling techniques, our approach allows us to be analytical. We are able to describe the adversary behaviour at a level of details that allows us to effectively explore the range of parameters or the behaviour of a system. This

enables us to infer not only the necessary conditions for an attack to be successful but also the sequencing of events or the way events have to be order for the attack to be successful.

In contrast to all the works described in this section, we present threat modelling of CPS using an applied π-calculus in this paper. The applied calculus has also been used for attacking modelling in [14]. The main differentiator of our approach to the existing literature on threat modelling and attack modelling in CPS is two fold. First, our method has the potential to be automated. This is because it is possible to take an applied π-calculus model and translate it into a theorem prover or prover assistance and then perform the reasoning automatically. However, it requires that the specification is sufficiently precise that it can be used to reason over the semantics. Second, our approach allows us to analyse the threats to CPS in a more precise way. It enables us to capture the pre-conditions that are applicable to certain types of threats. This is because a CPS will not always be vulnerable: there will be some states where manipulating a variable will have an effect and there will be other states where manipulating the same variable will not have an effect. Our method allows us to represent these states in the form of processes and the interaction between these processes and to reason about the likelihood of an attacker finding the system in a critical state to launch an attack for adverse effect.

6.4 Towards Threat Modelling of Cyber-Physical Systems Using an Applied π-Calculus

We conducted a review of the existing asset-centric threat modelling approaches in [35] and observed that the intuitive reasoning approach employed in those threat modelling approaches is not sufficient to threat model CPS where uncertainty, timing and dependencies between the entities exist. In threat modelling of CPS, we have to take into account that the processes we are trying to capture are not all predictable and deterministic [36]. Also, Murphy's law holds in CPS as it has been noted that a system with vulnerabilities will be exploited given a suitable operational environment [37]. A possible corollary for Murphy's law in CPS is that because of some variations in the process, one would obtain abnormal behaviour in the CPS at the same time as someone probing or attacking the system.

The existing threat modelling approaches assume that the systems being threat modelled are in a normal state when the attacker might strike. This type of assumption cannot capture all the possible threat scenarios in CPS because there are likelihoods that something may be going wrong with the CPS and at the same time a threat may be stressing the system even more. Situations like this are unlikely to occur in conventional systems, where the existing threat modelling approaches are usually applied. Hence, we need

to find a way of representing the interaction that are occurring between entities in CPS taking into account the unpredictable and nondeterministic behaviour of CPS.

Moreover, the existing threat modelling approaches are not good at expressing timing property between assets and operations. CPS as we have already observed, may have a handful of critical states. The threat modelling process for CPS should have a way of depicting such critical states because if an attack is launched when CPS is not in a critical state, the impact may be negligible. Unfortunately, the existing threat modelling approaches do not have a way of expressing the likelihood that an attacker may find the system in a critical state. Thus, it is important to consider an appropriate threat modelling approach for CPS, which takes into consideration the timing property between assets and operations.

Also, the inherent nature of CPS implies that there are dependencies between the assets at different levels and the operations of the system. It is no longer the case that the behaviour of the system can be understood by looking at the assets at the different components in isolation, but rather in combination with other assets. This is because an asset in CPS may be critical not in its own right, but instead as a provider for services in another asset. Also, these dependencies can be annotated with additional requirements to reason about how threats to these assets may be realized.

In additional to the dependencies between the assets at different levels and the operations of CPS, threats to availability are important requirements to consider. In a typical CPS, we are dealing with availability problem for example, redundancy. We are interested in expressing risks and threats to assets and services which can be provided in different ways. It is possible to examine a situation where we have an asset with a vulnerability, and to know if we can replace the output of that asset with some other substitute asset to give the same input to another asset that depends on the asset with a vulnerability.

By explicitly exposing the communication between the assets of a CPS using an applied π-calculus, we can deduce some reasoning about the behaviour of the system. One of the things that could be interesting for such a model is what it tells us about, for example, what an adversary can and cannot know about the state of CPS. There are some assets of the system that are only going to expose some information through messages or other interactions; and if an attacker is place at a particular place in that topology, the attacker would not be able to see any interaction unless there is a way of getting the message across. This is a useful insight because it might mean that any adversary would not be able to make use of this information and may have to rely on some sort of model.

Generally, we need a formal way of expressing these requirements that are peculiar to CPS environment. This will facilitate the deployment of an

appropriate threat modelling approach for the identification of threats to assets in CPS. So far, different methodologies have been proposed for formal modelling of CPS for the purpose of identifying threats to the system. However, we employ an applied π-calculus in this paper as described in following section to formally model the CPS environment. Then, we deploy the threat modelling approach using an applied π-calculus to evaluate threats that are applicable to CPS.

6.5 Threat Modelling of Cyber-Physical Systems Using an Applied π-Calculus

In this section, we present a formal model for CPS using an applied π-calculus. The formal model takes into account the requirements identified in the preceding subsection to ensure that the threats to CPS are captured. We then use the formal model to threat modelled the CPS.

6.5.1 Formal Model for Cyber-Physical System

An essential step in threat modelling a system is to develop a model of the system to be threat modelled. This would allow for the identification assets of the system and to reason about the likelihood of those assets being compromised. The process of identifying assets of a system is an important approach in threat analysis. It provides the security practitioner with insights into the most critical assets of the system and ensures efficient deployment of resources to protect those critical assets.

A CPS has a physical process under its control, a set of sensors that report the state of the process to a controller, which in turn sends control signals to actuators to maintain the system in a desired state. The controller also communicates with a supervisory and configuration device (e.g., a SCADA system in the power grid) which can monitor the system or change the settings of the controller. Figure 6.1 illustrates an example of CPS architecture. CPS consist of two components: a *physical plant/environment* that encloses all physical aspects of a system (state variables, physical devices, etc.) and a *cyber component* represented as a concurrent process that interacts with the physical devices (sensors and actuators) of the system and can communicate via channels with other processes of the same CPS or with processes of other CPS.

The cyber-physical system is widely modelled as a linear discrete-time stochastic system in state-space form as follows:

$$\begin{cases} x_{t+1} = Ax_t + Bu_t + w_t, \\ y_t = Cx_t + e_t, \end{cases} \tag{6.3}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ denote the plant's state and input vectors, respectively, while $y \in \mathbb{R}^p$ is the plant's output vector obtained from measurements of p sensors from the set $S = \{1, 2, ..., p\}$. The process noise $w_t \in \mathbb{R}^n$ and the measurement noise $e_t \in \mathbb{R}^m$ obey some zero-mean stochastic distributions. Moreover, $A \in \mathbb{R}^{nxn}$ is the system matrix, $B \in \mathbb{R}^{nxp}$ is the actuator matrix and $C \in \mathbb{R}^{mxn}$ is the measurement matrix. The next state x_{t+1} depends on the current state x_t and the corresponding control actions u_t, at the sampling instant $t \in N$.

As shown in figure 6.2, the physical plant is supported by a communication network through which the sensor measurements and actuator data are exchanged with the controller. The main interactions between cyber and physical components can be described as follows:

- The interactions between the *physical plant* and *sensors*
- The interaction between the *sensors* and the *controller*
- The interactions between the *controller* and the *actuators*
- The interactions between the *actuators* and the *physical plant*

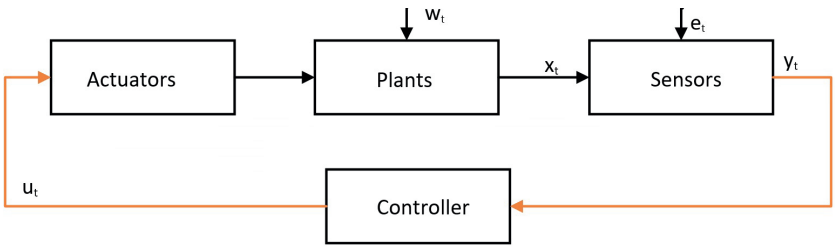


Figure 6.2: Interaction of a simplified CPS components

An Applied π-Calculus Representation

The combination of the *physical systems* (G) and the *cyber components* (P) which represents a typical example of CPS, improves the operations of the physical systems but introduces challenges to the bidirectional synchronisation between the components. Based on Lanotte *et al* [14] work, a variant of applied π-calculus is used to formalise and model the interactions of the CPS.

Physical Component: let $\bar{\mathcal{X}} \subseteq \mathcal{X}$ be a set of state variables, $\bar{\mathcal{A}} \subseteq \mathcal{A}$ be a set of actuators, and $\bar{\mathcal{S}} \subseteq \mathcal{S}$ be a set of sensors. The physical environment G is represented as $\{\xi_x, \xi_u, \xi_w, evol, \xi_e, meas, inv, safe, secure\}$, where:

- $\xi_{\mathcal{X}} \in \mathbb{R}^{\bar{\mathcal{X}}}$ is the *state function* that returns the current value associated to each variable in $\mathcal{X}$
- $\xi_u \in \mathbb{R}^{\bar{\mathcal{A}}}$ is the *actuator function* that returns the current value associated to actuators in $\mathcal{A}$
- $\xi_w \in \mathbb{R}^{\bar{\mathcal{X}}}$ is the *uncertainty function* that returns the uncertainty/accuracy associated to each state variable
- $evol$: $\mathbb{R}^{\bar{\mathcal{X}}} x \mathbb{R}^{\bar{\mathcal{A}}} x \mathbb{R}^{\bar{\mathcal{X}}} \rightarrow 2^{\mathbb{R}^{\bar{\mathcal{X}}}}$ is the *evolution map* that models the evolution law of the physical system, where changes made on the actuators may reflect on state variables
- $\xi_e \in \mathbb{R}^{\bar{\mathcal{S}}}$ is the *sensor-error function* that returns maximum error associated to sensors in $\bar{\mathcal{S}}$
- $meas$: $\mathbb{R}^{\bar{\mathcal{X}}} x \mathbb{R}^{\bar{\mathcal{S}}} \rightarrow 2^{\mathbb{R}^{\bar{\mathcal{S}}}}$ is the *measurement map* that returns the set of next admissible sensor measurements based on the current state function
- inv: $\mathbb{R}^{\bar{\mathcal{X}}} \rightarrow \{true, false\}$ is the *invariant set* that returns the set of state functions that satisfy the invariant of the system
- $safe$: $\mathbb{R}^{\bar{\mathcal{X}}} \rightarrow \{true, false\}$ is the *safety function* that represents the set of state functions that satisfy the safety conditions of the system
- $secure$: $\mathbb{R}^{\bar{\mathcal{X}}} \rightarrow \{true, false\}$ is the *security function* that represents the set of state functions that satisfy the security properties of the system. Specifically, if a CPS gets into an insecure state, then its security property may get compromised

The *cyber components* of a CPS are defined using an applied π-calculus with constructs to read values detected at the sensors and write values on actuators. Processes are defined as follows:

$$
\begin{aligned}
&P, Q ::= nil \quad | \quad \tau.P \quad | \quad P|Q \quad | \quad \lfloor \pi.P \rfloor Q \quad | \quad if \\
&(b)\,\{P\} \quad else\,\{Q\} \\
&\pi ::= snd\,\bar{c}\langle v\rangle \,|\, rcv\,c(x) \,|\, read\,s(x) \,|\, write\,\bar{a}\langle v\rangle \\
&\mu ::= forge\,p\langle v\rangle \quad | \quad drop\,a(x)
\end{aligned}
$$

The *nil* represents a *terminated process*. The process $\tau.P$ represents a silent action and then continues as P. $P|Q$ denotes the parallel composition of concurrent threads P and Q. Thus, $\lfloor snd\,\bar{c}\langle v\rangle.P \rfloor Q$ sends the value v on channel c, and it continues as P; otherwise, it evolves into Q. The process $[rcv\,c(x).P]Q$ represents the reception case. The process $[read\,s(x).P]Q$ reads the value detected by the sensor S, whereas $[write\,\bar{a}\langle v\rangle.P]Q$ writes on the actuator a. The process $if(b)\{P\}else\{Q\}$ is the standard conditional, where b is a decidable guard. For $\{\mu \in forge\,p\langle v\rangle, \quad drop\,a(x)\}$, the process

$\lfloor \mu.P \rfloor Q$ denotes the threats targeting a CPS system. Specifically, the attacks represent integrity attacks on data coming from sensors to the controller and dropping of actuator commands.

Labelled Transition Semantics

We define the semantics of the applied π-calculus using a *labelled transition system* to highlight the CPS interactions with the environment and enable the definition of observational equivalences such as *bisimilarity*. The operational semantics is given in tables 6.2 and 6.3. The rules of table 6.2 describe the behaviour of processes whereas the rules in table 6.3 describe the behaviour of a CPS. A transition of P has the form $P \xrightarrow{\alpha} P'$, specifying that P can perform action α to evolve into P' where α can represent different actions. The meta-variable α ranges over labels in the set $\{nil, \tau, \bar{c}v, cv, a!v, s?v, p!v, p?v, \tau : p\}$. Rules(*outp*), (*Inpp*) and (*Com*) serve to model channel communication on some channel c. Rules (*write*) and (*read*) denote the writing/reading of some data on the physical device p. Rule *(SensWrite)* models an integrity attack on sensor s. Rule (*Par*) propagates untimed actions over parallel components.

Table 6.2: LTS for Processes

$$(Outp)\ \frac{-}{\lfloor sndc\langle v\rangle.P\rfloor Q \xrightarrow{\bar{c}v} P} \qquad (Inpp)\ \frac{-}{\lfloor rcvc(x).P\rfloor Q \xrightarrow{\text{cv}} P\{v/x\}}$$

$$(Com)\ \frac{P \xrightarrow{\bar{c}v} P' \quad Q \xrightarrow{\text{cv}} Q'}{P|Q \xrightarrow{\tau} P'|Q'} \qquad (Par)\ \frac{P \xrightarrow{\lambda} P' \quad \lambda \neq nil}{P|Q \xrightarrow{\lambda} P'|Q}$$

$$(Write)\ \frac{-}{\lfloor writea\langle v\rangle.P\rfloor Q \xrightarrow{\text{a!v}} P} \qquad (Read)\ \frac{-}{\lfloor reads(x).P\rfloor Q \xrightarrow{\text{s?v}} P\{v/x\}}$$

$$(forge)\ \frac{P \xrightarrow{!s!v} P' \quad Q \xrightarrow{!s?v} Q'}{P|Q \xrightarrow{\tau:s} P'|Q'} \qquad (ActRead)\ \frac{P \xrightarrow{a!v} P' \quad Q \xrightarrow{!a?v} Q'}{P|Q \xrightarrow{\tau:a} P'|Q'}$$

The transition rules for the physical and cyber components are given in Table 6.3. A CPS can evolve if the invariant property is satisfied, otherwise the system will be in undesirable state or deadlocked. Actions ranged over by α are in the set $\{\tau, \bar{c}v, cv, nil\}$. These actions denote non-observable activities (τ), observable activities such as channel transmissions ($\bar{c}v$ and cv). Rules (*out*) and (*Inp*) model transmission and reception with an external system on a channel c. Rule (*SensRead*) models the reading of the current data detected at sensor s. Rule(*ActWrite*) models the writing of a value v on an actuator a.

Table 6.3: LTS for CPS

$$(Out)\frac{P \xrightarrow{\bar{c}v} P' \quad inv(G)}{G \bowtie P \xrightarrow{\bar{c}v} G \bowtie P'} \qquad (Inp)\frac{P \xrightarrow{cv} P' \quad inv(G)}{G \bowtie P \xrightarrow{cv} G \bowtie P'}$$

$$(SensRead)\frac{P \xrightarrow{s?v} P' \quad s \in S \quad inv(G) \quad v \in read\_sensor(G,s)}{G \bowtie P \xrightarrow{\tau} G \bowtie P'} \qquad (Tau)\frac{P \xrightarrow{\tau} P' \quad inv(G)}{G \bowtie P \xrightarrow{\tau} G \bowtie P'}$$

$$(ActWrite)\frac{P \xrightarrow{a!v} P' \quad a \in S \quad G' = update\_act(G,a,v)}{G \bowtie P \xrightarrow{\tau} G' \bowtie P'} \qquad (Deadlock)\frac{'inv(G)}{G \bowtie P \xrightarrow{\text{deadlock}} G \bowtie P}$$

$$(Safety)\frac{'safe(G) \quad inv(G)}{G \bowtie P \xrightarrow{\text{unsafe}} G \bowtie P} \qquad (Security)\frac{'secure(G) \quad inv(G)}{G \bowtie P \xrightarrow{\text{insecure}} G \bowtie P}$$

Bisimulation

Bisimulation is a binary relation between state transition systems in which the systems behave in the same way in the sense that one system simulates the other and vice versa. The operational semantics of the CPSs is described in subsection 6.5.1 in terms of a Labelled Transition Semantics similar to the SOS style of Plotkin [38]. This subsection defines a *weak bisimulation*-based behavioural equivalence for CPSs. The capability to observe physical events depends on the capability of the cyber components to recognise these events by acting on sensors and actuators, and the transmission of messages (over unrestricted channels) can be observed.

Consider the labelled transition system $A = (S, Act, \rightarrow, s, T)$. A relation R over CPSs $R \subseteq SxS$ is defined as a *weak bisimulation relation* iff for all $s, t \in S$ such that $s \; R \; t$, the following conditions hold [39]:

1. If $s \xrightarrow{\alpha} s'$, then
   - either $\alpha = \tau$ and $s' \; R \; t$, or
   - there is a sequence $t \xrightarrow{\tau} ... \xrightarrow{\tau} \xrightarrow{a} \xrightarrow{\tau} ... \xrightarrow{\tau} t'$ such that $s' \; R \; t'$
2. Symmetrically, if $t \xrightarrow{\alpha} t'$, then
   - either $\alpha = \tau$ and $s \; R \; t'$, or
   - there is a sequence $s \xrightarrow{\tau} ... \xrightarrow{\tau} \xrightarrow{\alpha} \xrightarrow{\tau} ... \xrightarrow{\tau} s'$ such that $s' \; R \; t'$
3. If $s \in T$, then there is a sequence $t \xrightarrow{\tau} ... \xrightarrow{\tau} t'$ such that $t' \in T$
4. Again, symmetrically, if if $t \in T$, then there is a sequence $s \xrightarrow{\tau} ... \xrightarrow{\tau} s'$ such that $s' \in T$

Two states s, t are weakly bisimilar $(s \approx t)$ if there exists a weak bisimulation R such that $\langle s, t \rangle \in R$. In order to consider two states equivalent, it is

necessary that for each visible action performed by one of them, the other has to have the possibility of performing the same visible action possibly preceded and followed by any number of invisible actions.

We consider two states (systems) are equivalent if they behave indistinguishably in the presence/absence of any adversary, where the adversary can compromise the security property of the system.

6.5.2 Threat Modelling of Cyber-Physical Systems Using an Applied π-Calculus

This section formalises a threat model using an applied π-calculus to specify denial of service and man-in-the-middle (MITM) attacks that can manipulate sensor readings or control commands in order to drive a CPS into an undesired state. In figure 6.2, the output of the process (y_t) is transmitted over a communication network, and the received output is used to compute control actions (u_t) which are sent back to the physical process. In between the transmission and reception of sensor data and control commands, an attacker may replace the signals coming from the sensors to the controller and from the controller to the actuators. Thus, after each transmission and reception, the attacked output $\bar{y}$ and attacked input $\bar{u}$ take the form

$$\begin{cases} \bar{y}_t := y_t + \delta_t^y, \\ \bar{u}_t := u_t + \delta_t^u, \end{cases} \tag{6.4}$$

where $\delta_t^y \in \mathbb{R}^m$ and $\delta_t^u \in \mathbb{R}^l$ denote additive sensor and actuator attacks, respectively.
Then, a system under attack from equations (1) is modelled by

$$\begin{cases} x_{t+1} = Ax_t + B(u_t + \delta_t^u) + w_t, \\ \bar{y}_t = Cx_t + e_t + \delta_t^y. \end{cases} \tag{6.5}$$

A residue vector (Δz_t) represents the difference between the system in the presence of attacks $\bar{z}_t$ and the system without attacks z_t , and it determines if an attack can be detected or not. An attack is hardly detectable if Δz_t is small enough, i.e., there exist $\delta > 0$ such that if $||\Delta z_t|| \leq \delta, \forall t \in \mathcal{N}$ [40].

Different attack scenarios can also be considered in the architecture illustrated in figure 6.2:

1. An attacker can inject false measurement into the system by faking sensor data and cause the controller to act on malicious data. This can be formalised in the applied calculus as:

   $$S(forge\ M)_{adv} = snd\, c_{SC}\langle M \rangle$$

   where c_{SC} is the channel used by the sensor for sending measurement value to the controller;

2. An attacker may be able to compromise the controller and send incorrect control signals to the actuators. This can be formalised in the applied calculus as:

$$C = rcv\ c_{SC}(x).\ snd\ c_{CA}\langle comprom.\ x\rangle_{adv}$$

where c_{SC} is the channel used by the controller for receiving measurement value from the sensor and c_{CA} is the channel used by the controller for sending a control signal to the actuators;

3. An attacker can compromise the actuators and execute a control action that is different to what the controller intended. This can be formalised in the applied calculus as:

$$A = rcv\ c_{CA}(comprom.\ x)_{adv}$$

where c_{CA} is the channel used by the actuators for receiving measurement value from the controller.

Depending on the motive and level of access, an adversary may block and/or modify messages from a compromised device or link. For example, if an attacker controls a sensor that outputs a measurement, then the controller may receive a corrupted version of the measurement. And because we are explicitly modelling distributed systems, we are not able to obtain a global view of the dynamical state of the system. However, we can approximate the dynamics of the system using the local state of the processes. These dynamics will be reflected in the message passing in the applied π-calculus.

Moreover, we are interested in the threats posed by cyber-physical attacks and are concerned with events that leaves a reflection in the cyber domain. Since the cyber domain is a distributed system, we consider the synchronization that can be observed and modified by the attacker. And for the physical system, the internal processes will evolve according to certain dynamics but what can be observed are only the reflection of the internal dynamics whenever there is an interaction with another process. The zero dynamics or the internal behaviour of the processes is beyond the scope of our model. Also, labelled bisimilarity has been showed to be the same as observation equivalence [41, 12]. This implies that as long as we have bisimilarity and can prove it, the internal dynamics of the physical system process can be inferred.

To formally reason about the necessary conditions for the attacks described above to be successful, we use attack-defence trees (ADT) which are employed to analyse an attack-defence scenario [42] and extend it with partial ordering of events to show causality relationship. We then translate the ADT with partial ordering into an applied π-calculus using the message synchronization primitives for partial ordering. This is because an ADT shows

that a particular attack will succeed if some conditions are met but the approach only works if there is no timing or sequencing of events. The abstract representation of ADT with partial ordering is depicted in figure 6.3.

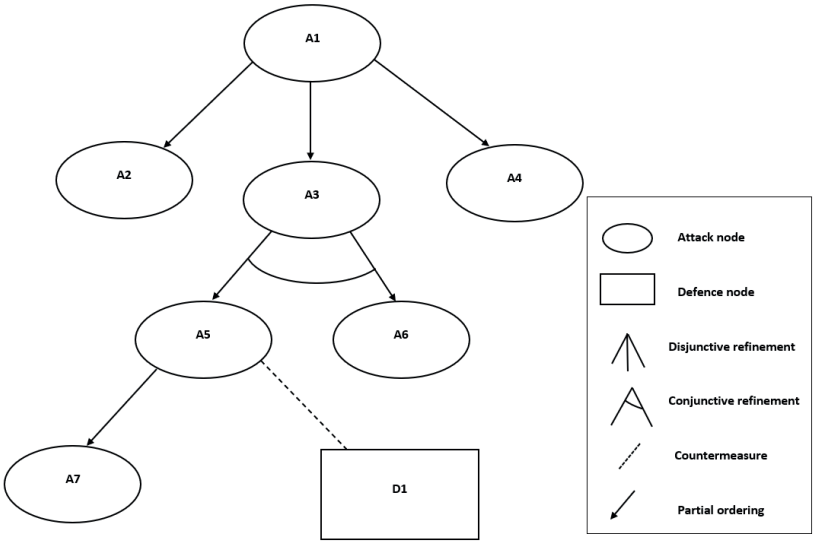


Figure 6.3: An Abstract Attack-Defence Tree with Partial Ordering

One of the main goals of an attacker in the CPS environment is to cause an undesirable state change in the physical system. We consider two forms of attacks that can be deployed to drive the physical system into an undesirable state: attacking the sensor or attacking the actuator. In order to drive the physical system into an undesirable state using the sensor, the attacker needs both, to compromise the communication channel and to inject false sensor measurement value. We ignore how an attacker might inject the false sensor measurement value and focus on how the communication channel is compromised. The communication channel could be compromised using MITM attack. However, the defender could counter the attacker's action by securing the communication channel. This defence mechanism is subject to the requirements of the specific CPS environment.

For actuator attack, the attacker needs to compromise the communication channel and to inject false control command. We ignore how an attacker might inject the false control command and focus on the communication channel is compromised. Similar to the sensor attack, the communication channel can be compromised through MITM attack. Also, the defender can protect against this attack by securing the communication channel. This protection mechanism would have to be designed so as to meet the require-

ments of the specific CPS environment that such mechanism would be deployed.

The ADT with partial ordering representing the above described state is shown in figure 6.4.

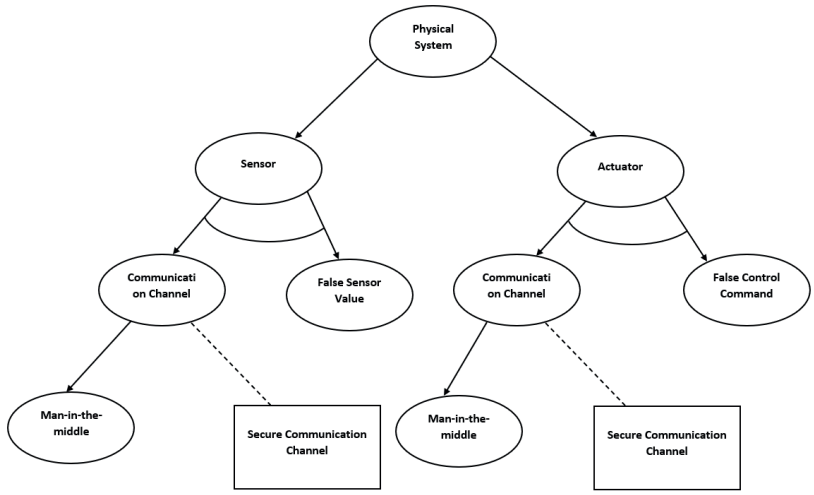


Figure 6.4: The Attack-Defence Tree with Partial Ordering for CPS Attack

And the ADTerm representing the ADT using the semantics that can be found in [42] with an extension to show partial ordering of events is given as follows:

$$\bigvee^{p} \Big[\wedge^{p} \Big(\text{MITM}, \leq c^{p}\big(\wedge^{p} (\text{Comm. Channel}, \text{False Sensor Value}), c^{o}(\text{Secure Comm. Channel})\big), \leq \text{Sensor}\Big), \wedge^{p}\Big(\text{MITM}, \leq c^{p}\big(\wedge^{p} (\text{Comm. Channel}, \text{False Control Command}), c^{o}(\text{Secure Comm. Channel})\big), \leq \text{Actuator}\Big)\Big]$$

This formalism can be translated into the applied π-calculus using the message synchronization primitives for partial ordering to represent not only the necessary conditions for an attack to be successful but also the sequencing of events or the way events have to be order for the attack to be successful. This allows us to reason about the timing property of the CPS because the concept of time is derived from the order in which events occur [43]. Although we do not consider explicit timing, we use partial ordering of events to represent dependencies of internal states and the function over

these states and how they are linked together with the message - basically the semantics of the applied π-calculus. Thus, we have internal states that represent the current states and then we have a message and that message is implicitly creating a partial ordering over the events which we might consider the equivalent of ADT but with sequencing of operation.

To translate the ADT with partial ordering in figure 6.4 into an applied π-calculus using the message synchronization primitives for partial ordering and taking the sensor attack into consideration, the sensor (S), controller (C) and actuator (A) activities can be represented as parallel composition ($S|C|A$). This composition where S and C are connected by a channel c_{SC}, and C and A by a channel c_{CA} shows the partial ordering of the events within the CPS. The sensor uses c_{SC} channel for sending a measurement to the controller, and the controller uses c_{CA} channel for sending a control signal to the actuators. We can represent these partial orders as follows:

$$\begin{aligned} S \rightarrow C: &\quad M \quad on \quad channel \quad c_{SC} \\ C \rightarrow A: &\quad M \quad on \quad channel \quad c_{CA} \end{aligned}$$

The actual applied π-calculus description of this message interaction (M) is:

$$\begin{aligned} S(M) &= snd\, c_{SC}\langle M\rangle \\ C &= rcv\, c_{SC}(x).\, snd\, c_{CA}\langle x\rangle \\ A &= rcv\, c_{CA}(x)) \\ Ctrl(M) &= (\nu c_{SC})(\nu c_{CA})(S(M)|C|A) \end{aligned}$$

Thus, the whole CPS is defined as: $CPS = G \bowtie Ctrl(M)$,
where G is the physical environment defined in subsection 6.5.1.

Similarly, the adversary's actions must coincide with message transactions for an attack to be successful. We consider Dolev-Yao threat model [44] where an adversary can compromise the communication channel to inject false measurement into the system by faking sensor data and causing the controller to act on malicious data. An applied π-calculus description of the false measurement injection using the compromised communication channel and false sensor measurement value S_{adv} can be given as follows:

$$\begin{aligned} S(forge\, M)_{adv} &= snd\, c_{SC}\langle M\rangle \\ C &= rcv\, c_{SC}(x).\, snd\, c_{CA}\langle x\rangle \\ A &= rcv\, c_{CA}(x)) \\ Ctrl(M)_{adv} &= (\nu c_{SC})(\nu c_{CA})(S(M)_{adv}|C|A) \end{aligned}$$

Thus, the CPS under attack is defined as:

$$C\bar{P}S = G \bowtie Ctrl(M)_{adv}$$

It is then trivial to derive the necessary conditions under which the attack will be successful and the way events have to be order for the attack to be successful. Therefore, our threat model allows us to know not only the conditions the attacker would have to meet to be successful but also the partial ordering of the events needed for the successful execution of the attack in the CPS environment.

6.5.3 Use Case Scenario

In this subsection, we demonstrate the utility of our proposed model using the IEC 61850 based substation. The IEC 61850 is a standard which defines the communication requirements for substation automation [45]. The communication architecture for substation automation as defined by IEC 61850 standard include three different levels of communication, namely: the station level, the bay level and the process level [45]. The IEC 61850 communication architecture is shown in Figure 6.5.

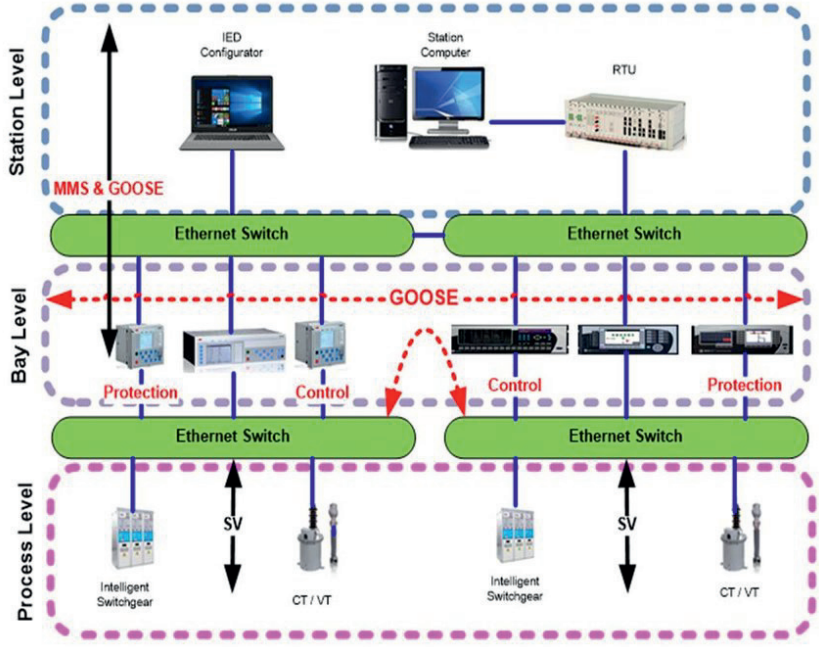


Figure 6.5: IEC 61850 Communication Architecture

[46]

However, for the purpose of illustrating the use of our model, we use an applied π-calculus to reason about the potential threats to the interaction

within/between the process level and the bay level. The process level as can be seen from Figure 6.5 consists of devices such intelligent switchgear, current transformer (CT), and voltage transformer (VT); and the bay level consists of protection and control devices which include relays, intelligent electronic devices (IEDs), and phasor measurement unit (PMU). IEC 61850 protocols such as the generic object-oriented substation event (GOOSE) and sampled value (SV) are then employed for communication within/between the process level and the bay level. Whilst GOOSE is used to send tripping signals from the IEDs to a circuit breaker (CB), SV is used to send sampled voltage and current values from MU to IEDs.

The bay level consists of different IEDs that collect sample measured values from the process level devices (currents and voltages using sampled value messages from merging units) via the communication channel (local area network). The IEDs can make local control decisions, protect irregularities from the process level, transmit data to other IEDs, or send the data to the substation level for further processing and monitoring. Also, the CB will receive a trip signal from the IEDs using a GOOSE message travelling in the process bus.

Moreover, there are situations where the CB may fail. The protective mechanisms for such scenarios are initiated via the communication channel. The IED relay would broadcast a GOOSE message to the adjacent break control relays. On receiving the GOOSE message, the breaker control relays would trip and block close their respective breakers. The breaker control relays also communicate with each other using the communication channel.

However, an adversary may inject false streaming measurement data (currents and voltages) with the intent of causing the protective relays to issue false tripping commands (inducing the IED to trip CBs). For instance, a malicious MU (sensor) can issue a false SV to an IED (controller) that indicates a fault current when there is no fault, and it leads to a needless CB trip action by the relay subscribing to the SV stream.

The interaction between the adversary and the defender can be represented using the ADT with partial ordering. As we have observed in the description of our use case scenario, the main goal of the attacker is to cause physical state change i.e., needless CB trip action by the relay subscribing to the SV stream. To achieve this goal, the attacker would have to cause a sensor change. This sensor change can be triggered by compromising the communication channel and injecting false SV value. In this type of attack, MITM approach can be adopted by in order to compromise the communication channel. The defender, on the other hand, can counter the actions of the attacker by securing the communication channel between the sensor and the controller. The ADT with partial ordering showing this interaction between the attacker and the defender for the use case scenario we have described is equivalent to the sensor attack path of the ADT with partial ordering for

CPS attack in Section 6.5.2.

The ADT translation into an applied π-calculus using the message synchronization primitives for partial ordering, representing the MU, IED and CB regular measurement (RM) interaction instance, and its variant with the false measurement injection (FMI) attack is given as follows:

$$\begin{aligned} S(M) &= snd\, c_{SC}\langle M\rangle \\ C &= rcv\, c_{SC}(x).\, snd\, c_{CA}\langle x\rangle \\ A &= rcv\, c_{CA}(x)) \\ RM(M) &= (\nu c_{SC})(\nu c_{CA})(S(M)|C|A) \end{aligned}$$

The MU measurement instance with the false measurement injection attacks is given as follows:

$$\begin{aligned} S(forge\ M)_{adv} &= snd\, c_{SC}\langle M\rangle \\ C &= rcv\, c_{SC}(x).\, snd\, c_{CA}\langle x\rangle \\ A &= rcv\, c_{CA}(x)) \\ FMI(M)_{adv} &= (\nu c_{SC})(\nu c_{CA})(S(forge\ M)_{adv}|C|A) \end{aligned}$$

We can then make an argument for equivalence where we can say whether that this attack can be successful. For example, the outputs of these two instances (i.e., $RM(M)$ and $FMI(M)_{adv}$) can be considered indistinguishable by an external observer such as the defence mechanism. This *undetectability* property can be formalised as *observational equivalence* [10]. Observational equivalence is used to express the notion that two system instances are observationally equivalent if they behave indistinguishably *from the defender's or the attacker's perspective*. It can be used to specify security properties such as the inability of the defender or the attacker to distinguish between two instances of a system. To show the undetectability property of the two output instances, we use weak bisimulation defined in subsection 6.5.1 as follows.

Given the two instances $RM(M)$ and $FMI(M)_{adv}$, lets assume that

$$P_i \approx RM(M)$$

and

$$Q_i \approx FMI(M)_{adv}$$

for all i. Our goal is to show that $P_i \approx Q_i$ for each i. To achieve this, it is sufficient to demonstrate that

$$S = \{\, (P, Q) \mid P \approx P_i \text{ and } Q \approx Q_i \text{ for some } i \,\}$$

is a weak bisimulation.

Proof: So, lets consider an arbitrary pair $(P, Q) \in S$. First, suppose $P \Rightarrow P'$. Then

$$P \approx P_i \approx RM(M) \Rightarrow P'' \approx P', \text{ for some } P''.$$

But since $\alpha_{ij} \neq \tau$ for all i, j; it follows that $P \approx P'$. By selecting $Q' = Q$, we actually have a Q' such that $Q \Rightarrow Q'$ and $(P', Q') \in S$.

Also, suppose $P \stackrel{\lambda}{\Rightarrow} P'$. Then

$$P \approx P_i \approx RM(M) \stackrel{\lambda}{\rightarrow} P_k \Rightarrow P'' \approx P',$$

where $\lambda = \alpha_{ij}$ and $P_k = P_{k(ij)}$ for some j. Using the same reasoning as above, $P_k \approx P'$. Further, we have

$$FMI(M)_{adv} \stackrel{\lambda}{\rightarrow} Q_k;$$

but

$$Q_i \approx FMI(M)_{adv},$$

so

$$Q \stackrel{\lambda}{\Rightarrow} Q' \approx Q_k$$

for some Q', which is what we require to complete the proof.

Consequently, for the false measurement injection attack we have described, the necessary conditions for the attack to be successful and the sequencing of events or the way events have to be order for the attack to be successful are given by the sensor attack path of the ADT with partial ordering for CPS attack in Figure 6.4. We also translated the ADT with partial ordering into an applied π-calculus using the message synchronization primitives for partial ordering so as to make an argument for equivalence; where we have showed that the false measurement injection attack can be successful if the output of the two instances - $RM(M)$ and $FMI(M)_{adv}$ - are observationally equivalent.

6.6 Conclusion and Future Work

Indeed, an understanding of assets that make up a CPS and the use of an applied π-calculus can provide useful insights about threats to a CPS and help in reasoning about its behaviour. We have presented threat modelling of CPS using an applied π-calculus in this paper. We argued that the regular threat modelling approaches are not suitable for capturing the threats to CPS considering the uncertainty, timing and dependencies that exist between its

entities. We then proposed an extension to an applied π-calculus which allows us to capture both the behaviour of the CPS as well as modelling possible adversary behaviour. Lastly, the utility of our model was demonstrated for the case of an electrical substation fragment in which components communicate via the IEC 61850 protocol.

In the future, we hope to consider the possibility of automating the threat modelling process of CPS using the theoretical foundations presented in this work. This is because the use of an automated support tool can facilitate the threat modelling of a much more complex system. In addition, we intend to expand our approach to investigate threats in other sectors within the critical infrastructure. To that end, an adequate applied π-calculus model for the system under consideration would have to be developed.

6.7 Bibliography

[1] ZETTER, K. An Unprecedented Look at Stuxnet, the World's First Digital Weapon. 2014. Available from: `https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet/` 5, 118

[2] PALMER, D. Ransomware attacks are now targeting industrial control systems. *ZDNet*, 2020 119

[3] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST). Information Security: Guide for Conducting Risk Assessments *NIST Special Publication 800-30*, 2012 100, 119

[4] SHOSTACK, A. Threat Modeling: Designing for Security. *John Wiley and Sons*, 2014 101, 103, 104, 119

[5] UCEDAVELEZ, T. AND MORANA, M. M. Risk Centric Threat Modeling: Process for Attack Simulation and Threat Analysis. *John Wiley and Sons*, 2015 101, 102, 103, 104, 105, 108, 109, 111, 119

[6] ALBERTS, C.; DOROFEE, A.; STEVENS, J. AND WOODY, C. Introduction to the OCTAVE Approach. *Software Engineering Institute*, 2003 106, 107, 108, 120

[7] FERNANDEZ, E. B. Threat Modeling in Cyber-Physical Systems. In *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, 2016 18, 113, 121, 125

[8] SADI, M. A. H.; ALI, M. H.; DASGUPTA, D.; ABERCROMBIE, R. K. AND KHER, S. Co-Simulation Platform for Characterizing Cyber Attacks in

Cyber Physical Systems. In *Proc. IEEE Symp. Series Computational Intelligence*, 2015, 1244-1251 122

[9] MILNER, R.; PARROW, J. AND WALKER, D. A calculus of mobile processes, I . *Information and Computation*, Elsevier BV, 1992, 100, 1-40 9, 122

[10] SANGIORGI, D. AND WALKER, D. The pi-calculus: a Theory of Mobile Processes. *Cambridge university press*, 2003 9, 82, 93, 122, 140, 159

[11] ABADI, M. AND GORDON, A. D. A Calculus for Cryptographic Protocols: The Spi Calculus . *Information and Computation*, Elsevier BV, 1999, 148, 1-70 9, 122, 148

[12] ABADI, M. AND FOURNET, C. Mobile values, new names, and secure communication. *ACM SIGPLAN Notices, Association for Computing Machinery (ACM)*, 2001, 36, 104-115 9, 122, 134

[13] KREMER, S. AND RYAN, M. Analysis of an Electronic Voting Protocol in the Applied Pi Calculus. *Programming Languages and Systems*, Springer Berlin Heidelberg, 2005, 186-200 122

[14] LANOTTE, R.; MERRO, M.; MURADORE, R. AND VIGANÒ, L. A formal approach to cyber-physical attacks . *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, 2017, 436-450 15, 16, 17, 18, 19, 85, 122, 126, 129, 148, 149, 150, 155, 157

[15] PARROW, J. An Introduction to the pi-Calculus. *Handbook of Process Algebra*, Elsevier, 2001, 479-543 11, 123

[16] MO, Y.; CHABUKSWAR, R. AND SINOPOLI, B. Detecting Integrity Attacks on SCADA Systems. *IEEE Transactions on Control Systems Technology,*, IEEE, 2014, 22, 1396-1407 17, 124

[17] PASQUALETTI, F.; DORFLER, F. AND BULLO, F. Attack Detection and Identification in Cyber-Physical Systems. *IEEE Transactions on Automatic Control*, IEEE, 2013, 58, 2715-2729 17, 124

[18] AMIN, S.; CÁRDENAS, A. A. AND SASTRY, S. S. Safe and Secure Networked Control Systems under Denial-of-Service Attacks. *Hybrid Systems: Computation and Control*, Springer Berlin Heidelberg, 2009, 31-45 7, 17, 124

[19] DOOSTMOHAMMADIAN, M. AND KHANC, U. A. Vulnerability of CPS inference to DoS attacks. In *2014 48th Asilomar Conference on Signals, Systems and Computers*, IEEE, 2014 124

[20] MO, Y. AND SINOPOLI, B. Secure control against replay attacks. In *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, 2009 124

[21] SÁNCHEZ, H. S.; ROTONDO, D.; ESCOBET, T.; PUIG, V.; SALUDES, J. AND QUEVEDO, J. Detection of replay attacks in cyber-physical systems using a frequency-based signature. *Journal of the Franklin Institute*, Elsevier BV, 2019, 356, 2798-2824 17, 124

[22] NWEKE, L. O.; WELDEHAWARYAT, G. K. AND WOLTHUSEN, S. D. Adversary Model for Attacks Against IEC 61850 Real-Time Communication Protocols. In *2020 16th International Conference on the Design of Reliable Communication Networks DRCN 2020*, IEEE, 2020 124, 155, 157

[23] MO, Y.; GARONE, E.; CASAVOLA, A. AND SINOPOLI, B. False data injection attacks against state estimation in wireless sensor networks. In *49th IEEE Conference on Decision and Control (CDC)*, IEEE, 2010 17, 124

[24] BEG, O. A.; JOHNSON, T. T. AND DAVOUDI, A. Detection of False-Data Injection Attacks in Cyber-Physical DC Microgrids. *IEEE Transactions on Industrial Informatics*, IEEE, 2017, 13, 2693-2703 124

[25] CHEN, B.; LI, H. AND ZHOU, B. Real-Time Identification of False Data Injection Attacks: A Novel Dynamic-Static Parallel State Estimation Based Mechanism. *IEEE Access*, IEEE, 2019, 7, 95812-9582 124

[26] ZALEWSKI, J.; DRAGER, S.; MCKEEVER, W. AND KORNECKI, A. J. Threat modeling for security assessment in cyberphysical systems. In *Cyber Security and Information Intelligence, CSIIRW '13, Oak Ridge, TN, USA*, ACM, 2013, 10 17, 124

[27] MARTINS, G.; BHATIA, S.; KOUTSOUKOS, X.; STOUFFER, K.; TANG, C. AND CANDELL, R. Towards a systematic threat modeling approach for cyber-physical systems. In *Proc. Resilience Week (RWS)*, 2015, 1-6 17, 125

[28] KHAN, R.; MCLAUGHLIN, K.; LAVERTY, D. AND SEZER, S. STRIDE-based threat modeling for cyber-physical systems. In *Proc. IEEE PES Innovative Smart Grid Technologies Conf. Europe (ISGT-Europe)*, 2017, 1-6 18, 125

[29] ALMOHRI, H.; CHENG, L.; YAO, D. AND ALEMZADEH, H. On Threat Modeling and Mitigation of Medical Cyber-Physical Systems. In *Proc. Systems and Engineering Technologies (CHASE) 2017 IEEE/ACM Int. Conf. Connected Health: Applications*, 2017, 114-119 18, 125

[30] REKIK, M.; GRANSART, C. AND BERBINEAU, M. Cyber-physical Threats and Vulnerabilities Analysis for Train Control and Monitoring

Systems. In *Proc. Computers and Communications (ISNCC) 2018 Int. Symp. Networks*, 2018, 1-6 18, 113, 125

[31] ATIF, Y.; JIANG, Y.; JIANGUO, D.; JEUSFELD, M.; LINDSTRÖM, B.; ANDLER, S.; BRAX, C.; HAGLUND, D. AND LINDSTRÖM, B. Cyber-threat analysis for Cyber-Physical Systems . *Technical Report*, University of Skövde, 2018 18, 113, 125

[32] ROCCHETTO, M. AND TIPPENHAUER, N. O. On Attacker Models and Profiles for Cyber-Physical Systems. *Computer Security – ESORICS 2016*, Springer International Publishing, 2016, 427-449 18, 125

[33] ADEPU, S. AND MATHUR, A. Generalized attacker and attack models for cyber physical systems. In *2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)*, IEEE, 2016, 1, 283-292 16, 18, 84, 125

[34] ADEPU, S. AND MATHUR, A. An Investigation into the Response of a Water Treatment System to Cyber Attacks . In *2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE)*, IEEE, 2016 18, 125

[35] NWEKE, L. O. AND WOLTHUSEN, S. D. A Review of Asset-Centric Threat Modelling Approaches. *International Journal of Advanced Computer Science and Applications*, 2020, 11, 1-6 126

[36] KROTOFIL, M.; CÁRDENAS, A. A.; LARSEN, J. AND GOLLMANN, D. Vulnerabilities of cyber-physical systems to stale data - Determining the optimal time to launch attacks. *Int. J. Crit. Infrastructure Prot.*, 2014, 7, 213-232 7, 126

[37] HUGHES, J. AND CYBENKO, G. Three tenets for secure cyber-physical system design and assessment. In *Cyber Sensing 2014*, 2014, 9097, 90970A 126

[38] PLOTKIN, G. D. A structural approach to operational semantics. *Computer Science Department, Aarhus University Aarhus, Denmark*, 1981 132, 150

[39] MILNER, R. A Calculus of Communicating Systems . *Springer-Verlag Berlin Heidelberg*, 1980, 174 132, 148

[40] WU, G.; SUN, J. AND CHEN, J. Optimal Data Injection Attacks in Cyber-Physical Systems. *IEEE Trans. Cybern.*, 2018, 48, 3302-3312 133

[41] ARAPINIS, M.; LIU, J.; RITTER, E. AND RYAN, M. Stateful Applied Pi Calculus. *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2014, 22-41 134

[42] KORDY, B.; MAUW, S.; RADOMIROVIĆ, S. AND SCHWEITZER, P. Foundations of Attack–Defense Trees. *Lecture Notes in Computer Science*, pringer Berlin Heidelberg, 2011, 80-95 134, 136

[43] LAMPORT, L. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, ACM, 1978, 21, 558-565 136

[44] DOLEV, D. AND YAO, A. On the security of public key protocols. *IEEE Transactions on Information Theory*, 1983, 29, 198-208 16, 78, 84, 88, 137

[45] THE INTERNATIONAL ELECTROTECHNICAL COMMISSION (IEC). Communication Networks and Systems in Substations. *IEC 61850 Standard*, 2011 138

[46] CLAVERIA, J. AND KALAM, A. GOOSE Protocol: IED's Smart Solution for Victoria University Zone Substation (VUZS) Simulator Based on IEC61850 Standard. In *Proc. IEEE PES Asia-Pacific Power and Energy Engineering Conf. (APPEEC)*, 2018, 730-735 138

Chapter 7

Corrigendum to Threat Modelling of Cyber-Physical Systems Using an Applied π-Calculus

Corrigendum to Threat Modelling of Cyber-Physical Systems Using an Applied π-Calculus

International Journal of Critical Infrastructure Protection

Livinus Obiora Nweke, Goitom Kahsay Weldehawaryat and Stephen D. Wolthusen

Abstract

The article [8] described the derivation and adaptation of the applied π-calculus to enable the integration of an explicit adversarial action model. In this, we had relied on earlier works by the authors as well as a contribution by Lanotte et al. [4, 5]. Several additional and related contributions to the applied π-calculus may, however, also be noted and are included in this supplement, specifically the family of models exemplified by Milner [7] and Hennessy and Regan [3].

In this corrigendum, the following addendum are to replace the first, the second and the fourth paragraphs of Section 5.1, page 5 of our original article.

The applied π-calculus models are extensions of the original π-calculus proposed by Robin Milner [7]. The work by Abadi and Gordon [1], in which the π-calculus was extended with cryptographic primitives is among the earliest extensions of the π-calculus. Several other extensions of the π-calculus can be found in the literature. For example, the π-calculus has been extended by Lee and Zic [6] to model the real-time aspect of systems in a mobile environment, and by Ciobanu and Prisacariu [2] to model the temporal aspects of distributed systems. A brief tutorial of the applied π-calculus and related formalisms can be found in [10]. In line with these developments, our work extends work of Lanotte et al. [4, 5], which builds on the earlier

work by Hennessey and Regan [3] but retains notation where possible to define the physical component as follows: Let $\bar{\mathcal{X}} \subseteq \mathcal{X}$ be a set of state variables, $\bar{\mathcal{A}} \subseteq \mathcal{A}$ be a set of actuators, and $\bar{\mathcal{S}} \subseteq \mathcal{S}$ be a set of sensors; the physical environment G is represented as $\{\xi_x, \xi_u, \xi_w, evol, \xi_e, meas, inv, safe, secure\}$, where:

- $\xi_{\mathcal{X}} \in \mathbb{R}^{\bar{\mathcal{X}}}$ is the *state function* that returns the current value associated to each variable in $\mathcal{X}$
- $\xi_u \in \mathbb{R}^{\bar{\mathcal{A}}}$ is the *actuator function* that returns the current value associated to actuators in $\mathcal{A}$
- $\xi_w \in \mathbb{R}^{\bar{\mathcal{X}}}$ is the *uncertainty function* that returns the uncertainty/accuracy associated to each state variable
- $evol$: $\mathbb{R}^{\bar{\mathcal{X}}} x \mathbb{R}^{\bar{\mathcal{A}}} x \mathbb{R}^{\bar{\mathcal{X}}} \rightarrow 2^{\mathbb{R}^{\bar{\mathcal{X}}}}$ is the *evolution map* that models the evolution law of the physical system, where changes made on the actuators may reflect on state variables
- $\xi_e \in \mathbb{R}^{\bar{\mathcal{S}}}$ is the *sensor-error function* that returns maximum error associated to sensors in $\bar{\mathcal{S}}$
- $meas$: $\mathbb{R}^{\bar{\mathcal{X}}} x \mathbb{R}^{\bar{\mathcal{S}}} \rightarrow 2^{\mathbb{R}^{\bar{\mathcal{S}}}}$ is the *measurement map* that returns the set of next admissible sensor measurements based on the current state function
- inv: $\mathbb{R}^{\bar{\mathcal{X}}} \rightarrow \{true, false\}$ is the *invariant set* that returns the set of state functions that satisfy the invariant of the system
- $safe$: $\mathbb{R}^{\bar{\mathcal{X}}} \rightarrow \{true, false\}$ is the *safety function* that represents the set of state functions that satisfy the safety conditions of the system
- $secure$: $\mathbb{R}^{\bar{\mathcal{X}}} \rightarrow \{true, false\}$ is the *security function* that represents the set of state functions that satisfy the security properties of the system. Specifically, if a CPS gets into an insecure state, then its security property may get compromised

The *cyber components* of the CPS are also defined using the applied π-calculus that can be found in [4, 5], which was derived from an earlier work [3] with constructs to read values detected at the sensors and write values on the actuators and they are given as follows:

$P, Q ::= nil \quad | \quad \tau.P \quad | \quad P|Q \quad | \quad \lfloor \pi.P \rfloor Q \quad | \quad if$
$(b)\ \{P\} \quad else\ \{Q\}$
$\pi ::= snd\ \bar{c}\langle v \rangle \,|\, rcv\ c(x) \,|\, read\ s(x) \,|\, write\ \bar{a}\langle v \rangle$
$\mu ::= forge\ p\langle v \rangle \quad | \quad drop\ a(x)$

The *nil* represents a *terminated process*. The process $\tau.P$ represents a silent action and then continues as P. $P|Q$ denotes the parallel composition of concurrent threads P and Q. Thus, $\lfloor snd\,\bar{c}\langle v\rangle.P\rfloor Q$ sends the value v on channel c, and it continues as P; otherwise, it evolves into Q. The process $[rcv\,c(x).P]Q$ represents the reception case. The process $[read\,s(x).P]Q$ reads the value detected by the sensor S, whereas $[write\,\bar{a}\langle v\rangle.P]Q$ writes on the actuator a. The process $if(b)\{P\}else\{Q\}$ is the standard conditional, where b is a decidable guard. For $\{\mu \in forge\,p\langle v\rangle, \quad drop\,a(x)\}$, the process $\lfloor\mu.P\rfloor Q$ denotes the threats targeting a CPS system. Specifically, the attacks represent integrity attacks on data coming from sensors to the controller and dropping of actuator commands.

Labelled Transition Semantics

Using the SOS style of Plotkin [9] as well as in [4, 5], we define the semantics of the applied π-calculus as a *labelled transition system* to highlight the CPS interactions with the environment and to enable the definition of observational equivalences such as *bisimilarity*. The operational semantics is given in Tables 1 and 2, which are similar to that of [4, 5] with no timing transitions in Table 1 and Table 2 includes security-related transition.

7.1 Bibliography

[1] ABADI, M. AND GORDON, A. D. A Calculus for Cryptographic Protocols: The Spi Calculus . *Information and Computation*, Elsevier BV, 1999, 148, 1-70 9, 122, 148

[2] CIOBANU, G.; PRISACARIU, C. Timers for distributed systems. *Electronic Notes in Theoretical Computer Science*, 2006, 164, 81–99 148

[3] HENNESSY, M.; REGAN, T. A process algebra for timed systems. *Information and Computation*, 1995, 117, 221–239 148, 149

[4] LANOTTE, R.; MERRO, M. A calculus of cyber-physical systems. In *Language and Automata Theory and Applications11th International Conference*, 2017, 115–127 15, 16, 17, 18, 19, 85, 122, 126, 129, 148, 149, 150, 155, 157

[5] LANOTTE, R.; MERRO, M.; MURADORE, R. AND VIGANÒ, L. A formal approach to physics-based attacks in cyber-physical systems. *ACM Transactions on Privacy and Security*, 2020, 23, 1–41 15, 16, 19, 148, 149, 150, 155, 157, 161, 171

[6] LEE, J.Y.; ZIC, J. On modeling real-time mobile processes. In *ACSC '02: Proceedings of the twenty-fifth Australasian conference on Computer science*, 2002, 139–147 148

[7] MILNER, R. A Calculus of Communicating Systems. *Springer-Verlag Berlin Heidelberg*, 1980, 174 132, 148

[8] NWEKE, L. O.; WELDEHAWARYAT, G. K. AND WOLTHUSEN, S. D. Threat modelling of cyber–physical systems using an applied pi-calculus. *International Journal of Critical Infrastructure Protection*, 2021, 35, 100466 148

[9] PLOTKIN, G. D. A structural approach to operational semantics. *Computer Science Department, Aarhus University Aarhus, Denmark*, 1981 132, 150

[10] SEWELL, P. Applied $pi – A Brief Tutoria. Technical Report. *Computer Laboratory, University of Cambridge.*, 2000 148

Chapter 8

A process algebraic approach to modelling cyber-physical systems security

This paper is awaiting publication and is not included in NTNU Open

ISBN 978-82-326-6216-6 (printed ver.)
ISBN 978-82-326-5875-6 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU

Norwegian University of
Science and Technology