Detection and Mitigation of Cyber Attacks on Time Synchronization Protocols for the Smart Grid

Bassam Moussa

A Thesis
In
The Concordia Institute
For
Information Systems Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Information and Systems Engineering)
Concordia University
Montréal, Québec, Canada

September 2018

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Bassam Moussa

Entitled: Detection and Mitigation of Cyber Attacks on Time Synchronization Protocols for the Smart Grid

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________ Chair
Dr. Robin Drew

______________________________ External Examiner
Dr. Frederic Cuppens

______________________________ External to Program
Dr. Mustafa K. Mehmet Ali

______________________________ Examiner
Dr. Lingyu Wang

______________________________ Examiner
Dr. Amr Youssef

______________________________ Examiner
Dr. Marthe Kassouf

______________________________ Thesis Co-Supervisor
Dr. Chadi Assi

______________________________ Thesis Co-Supervisor
Dr. Mourad Debbabi

Approved by ____
Dr. Anjali Awasthi, Graduate Program Director

Monday October 22, 2018

____
Dr. Amir Asif, Dean
Gina Cody School of Engineering and Computer Science

ABSTRACT

Detection and Mitigation of Cyber Attacks on Time Synchronization Protocols for the Smart Grid

Bassam Moussa, Ph.D.
Concordia University, 2018

The current electric grid is considered as one of the greatest engineering achievements of the twentieth century. It has been successful in delivering power to consumers for decades. Nevertheless, the electric grid has recently experienced several blackouts that raised several concerns related to its availability and reliability. The aspiration to provide reliable and efficient energy, and contribute to environment protection through the increasing utilization of renewable energies are driving the need to deploy the grid of the future, the smart grid. It is expected that this grid will be self-healing from power disturbance events, operating resiliently against physical and cyber attack, operating efficiently, and enabling new products and services. All these call for a grid with more Information and Communication Technologies (ICT). As such, power grids are increasingly absorbing ICT technologies to provide efficient, secure and reliable two-way communication to better manage, operate, maintain and control electric grid components.

On the other hand, the successful deployment of the smart grid is predicated on the ability to secure its operations. Such a requirement is of paramount importance especially in the presence of recent cyber security incidents. Furthermore, those incidents are subject to an augment with the increasing integration of ICT technologies and the vulnerabilities they introduce to the grid. The exploitation of these vulnerabilities might lead to attacks that can, for instance, mask the system observability and initiate cascading failures resulting in undesirable and severe consequences.

In this thesis, we explore the security aspects of a key enabling technology in the smart grid, accurate time synchronization. Time synchronization is an immense requirement across the domains of the grid, from generation to transmission, distribution, and consumer premises. We focus on the substation, a basic block of the smart grid system, along with its recommended time synchronization mechanism - the Precision Time Protocol (PTP) - in order to address threats associated with PTP, and propose practical and efficient detection, prevention, mitigation techniques and methodologies that will harden and enhance the security and usability of PTP in a substation. In this respect, we start this thesis with a security assessment of PTP that identifies PTP security concerns, and then address those concerns in the subsequent chapters. We tackle the following main threats associated with PTP: 1) PTP vulnerability to fake timestamp injection through a compromised component 2) PTP vulnerability to the delay attack and 3) The lack of a mechanism that secures the PTP network. Next, and as a direct consequence of the importance of time synchronization in the smart grid, we consider the wide area system to demonstrate the vulnerability of relative data alignment in Phasor Data Concentrators to time synchronization attacks. These problems will be extensively studied throughout this thesis, followed by discussions that highlight open research directions worth further investigations.

Acknowledgments

I would like to express my gratitude and acknowledgment to those who supported me to start and finish my PhD studies.

I am grateful for my supervisors, Prof. Chadi Assi and Prof. Mourad Debbabi, for their selfless support and guidance through this endeavor. I would have never made it without your consistent motivation, exemplary supervision, great guidance, and unconditional support. I specially appreciate your vision in seeing how pieces of this research developed and connected to each other. I am thankful for giving me the full experience any PhD student could aspire to have, including attending conferences, being involved with industry, and participating in developing grants. I would like to express my sincere gratitude to Dr. Marthe Kassouf for all the time she dedicated to meetings, thoughtful discussions and valuable feedback. Her contribution and support was essential to improve the quality of my thesis.

I am also thankful to the members of my supervisory committee: Prof. Mustafa Mehmet Ali, Prof. Amr Youssef, and Prof. Lingyu Wang, for their valuable constructive feedback and insightful suggestions. Also, it is a pleasure to truly acknowledge Prof. Frédéric Cuppens for accepting to serve as a delegate in my Ph.D. thesis examining committee.

Furthermore, I am thankful to all of my colleagues in the research lab at Concordia University including those who have left to their next adventure, and those who are still around. Thank you for the valuable discussions, help, support, and warm caring conversations. It was a pleasure to share all those moments with you.

I am blessed with precious friends, here in Montreal and back at home, to whom I am indebted for their unconditional love, support and valuable advice. Thank you for your encouragement ever since pursuing a PhD was just an idea.

I would like to express my deep gratitude to my family. To my parents, you are a true blessing in my life, you gave more than anyone can ever give, unconditionally

and endlessly. I hope that this achievement of mine meets your expectations, makes you proud, and pays back but little of what you have always given. My brothers, sisters, nephews, and nieces, I love you all and dedicate this thesis to you. You are the reason I see the world a better place everyday.

Finally, to my beautiful wife Assile, you believed in me from day one and supported me throughout this long journey. You always inspire me to be a better person. For all the good times and the tough ones, I love you and will always do.

To the one who believed in us, "the generations not born yet!"

Contents

List of Figures

List of Tables

Abbreviations

PTP	Precision Time Protocol
NTP	Network Time Protocol
PMU	Phasor Measurement Unit
PDC	Phasor Data Concentrator
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
GMC	Grand Master Clock
PPS	Pulse Per Second
IRIG	Inter-Range Instrumentation Group
UTC	Coordinated Universal Time
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
SNMP	Simple Network Management Protocol
MIB	Management Information Base
NSM	Network and System Management
BMCA	Best Master Clock Algorithm
TC	Transparent Clock
NTR	Network Time Reference
CTL	Computational Tree Logic
PCTL	Probabilistic Computational Tree Logic

ICV	Integrity Check Value
WAMS	Wide Area Monitoring System
WAMPAC	Wide Area Monitoring Protection and Control

Chapter 1

Introduction

1.1 Overview and Motivation

The current power grid, a system engineered in the early twentieth century, is considered outdated when contrasted to other systems tangled to our daily life and the technological advancements they witnessed. Its success in delivering power to consumers has been acceptable so far. However, its adequacy for future systems is questionable when considering the several blackouts experienced over the past years. This gave rise to the need for a self healing grid, a grid providing quality power for 21st century, clean and renewable energy, and active participation from customers. Those characteristics inspired the motive to integrate the advances in the information and communication technologies to define the grid of the future - the smart grid.

Today, the robust operation and the availability of the power grid is a critical requirement. The grid is exposed for threats both on its cyber and physical sides. Cyber-attacks are a consistent threat that is intensified with advances in the deployment of the smart grid. The increased dependency on the communication network and the integration of both systems present a potential attack surface for cyber-attacks. Further, the physical components of the grid are subject to attacks targeting their functionality, such as the reported attack on the high voltage transmission line in the United States in 2013 [113] and Canada in 2014 [24]. More recently, in the last week of 2015, a cyber attack targeted the Ukrainian power utilities and resulted in a blackout leaving hundreds of thousands of people without electricity for several hours[71]. Those blackouts demonstrate that our critical infrastructure is susceptible to faults and attacks that threaten its availability and functionality. The presence of such threats call for an innovative analysis of the functionality of the grid that results in a robust design of a smarter grid: a grid capable of restraining the effects of attacks and survive the loss of any of its components.

Moreover, control operations, effective monitoring and management of the grid require the presence of accurate synchronization of the grid events. Modern components are introduced to timestamp the observations they make about the grid status and the data they collect about its conditions. Indeed, having precise time available across the entire grid enables utilities to better monitor and control power systems with faster response times to effectively manage disturbances and ultimately prevent system-wide blackouts [10]. The need for accurate timing in power systems and alignment of data to a unified time source was stressed by the North American blackout in August 2003 [36]. Furthermore, with the adoption of the North American Electric Reliability Cooperation (NERC) Standard PRC018-1 in 2006, it is now a legal obligation that all recorded data must have an accuracy of 2 ms or better in relation to universal coordinated time scale (UTC) [23]. Thus, timing is a major issue in the design of such systems which typically use a time-slotted control protocol to perform sensing, computation, networking, and actuation on a periodic schedule [28].

On the other hand, the advance in deployment of smart grids depends on the ability to secure their operations. Such a requirement is of paramount importance for critical infrastructure including the power sector. Those concerns are escalated with the increase in dependence on information and communication technologies, global positioning satellite systems, and communication networks among others. The integration of those technologies into the power grid improves its availability and reliability. However along with their advantages, they carry a lot of security threats that need to be addressed, and they expand the attack surface used to target the grid's cyber side. As such, those additional security concerns should be addressed with proper prevention, detection, and mitigation mechanisms to ensure cyber and physical security of the future smart grid.

Hence, the deployment and wide spread of the smart grid is dependent on securing its components. Indeed, the transition phase from traditional power systems to the grid of the future carries a lot of challenges. The secure functionality of the grid, and an analysis of threats targeting its operations are on top of the list of challenges. To incite this transitional phase, we aim at assessing the security of one of the essential building blocks of the grid functionality, namely the time synchronization mechanism used to distribute accurate timing signals to the grid's components (substations, phasor measurements units, etc.). Availability of accurate timing is an enabler of the grid's monitoring, protection, and control applications on a wide scale which is currently referred to as WAMPAC systems (wide-area monitoring, protection, and control). Security concerns associated to the synchronization mechanism in use are brought forward by this mechanism to smart grid components relying on its services, and thus imposing a major threat on the availability and functionality of those components. IEEE 1588, more commonly known as the Precision Time Protocol (PTP) [4], is one of the recommended time synchronization mechanisms for use in the smart grid. PTP, in its current version, is vulnerable to a multitude of threats that affect its usability. Thus, our security assessment of time synchronization mechanisms is centered around PTP and its associated cyber attacks. We believe that securing PTP is one fundamental step towards a secure and cyber-attack resilient smart grid.

On the other hand, time synchronization is a candidate for use as an attack surface to target the functionality of other grid components. Indeed, due to the critical nature of the smart grid, there is a need to align and correlate events dispersed across its domain. Such a correlation is made possible through accurately timestamped and synchronized measurements sampling the grid dynamics in real time, thus intensifying the dependency of monitoring, protection and control systems on time synchronization. In this aspect, a thorough analysis of the specifications governing the functionality of different intelligent electronic devices, and their use of accurate time synchronization is needed to identify gaps and vulnerabilities that may be exploited by attackers to target the smart grid. The impact of such vulnerabilities escalate in the smart grid due to its dynamic nature, the interdependency between its power and communication components, and its cyber-physical nature. We tackle the presence of such vulnerabilities in phasor data concentrators, a key component of the wide area monitoring system (WAMS), to expose the impact of exploiting the system dependency on time synchronization, and leveraging existing vulnerabilities in mechanisms providing those services to target essential monitoring, protection, and control applications. This highlights the vulnerability of our critical infrastructure to cyber attacks, and emphasizes the need to consider security as the main requirement in each of their enabling technologies. We take a first step in that direction by addressing the security of time synchronization, and proposing solutions to prevent, detect and mitigate cyber attacks targeting those mechanisms and all systems built on top of the services they provide.

1.2 Thesis Contributions

This thesis aims to supplement the existing and ongoing research efforts towards a more secure and attack resilient smart grid. A first step in that direction is addressing threats that target time synchronization in the smart grid mainly by improving the security posture of PTP being the main candidate for time distribution at the substation level. The first contribution of this thesis manifests itself in a brief, yet comprehensive overview of the state-of-the-art security assessment of the Precision Time Protocol. Precisely, Chapter 2 of this thesis presents an in-depth review of PTP along with an introduction and classification of other available time synchronization mechanisms, and the time-dependent power system applications with an emphasis on the accuracy requirements of each of those applications.

Next, Chapters 3 through 6 will mainly address three problems pertaining to PTP and smart grid security. These problems are briefly presented next, and detailed in their dedicated chapters.

1.2.1 PTP Security Vulnerabilities

PTP is well-known for its capability to synchronize clocks of different qualities to a common time source while providing accuracy of the order of microseconds. However, PTP was not standardized with security in mind and is found vulnerable to a multitude of attacks targeting its services. For this purpose, in Chapter 3, we identify and address a security vulnerability in the authentication scheme followed by PTP security extension. In Chapter 4, we leverage the IEC 61850 substation synchronization requirements to devise a detection and mitigation schemes for the well-known PTP delay attack. The proposed mechanisms are formally modeled, validated, and evaluated on a real implementation of PTP. In addition to that, we introduce an extension to PTP that allows the collection of synchronization status from connected clocks for security purposes. The introduced extension is defined, formulated, and verified in Chapter 5. Through the proposed extension, we believe that a PTP network becomes more security aware and more resilient to cyber attacks which eases up the use of PTP for time synchronization in the smart grid.

1.2.2 Vulnerability of WAMS to Time Synchronization Attacks

Time synchronization is of immense importance for situational awareness in the smart grid, and for wide area monitoring and control. Through accurately timestamped sampling and collection of power parameters, the control center can devise necessary actions to maintain the grid stability and availability. This sampling and collection is enabled through the deployment of phasor measurement units (PMUs) and phasor data concentrators (PDCs) at selected locations in the grid. However, this dependency on accurate timing can be leveraged to exploit vulnerabilities in the specifications and functionality of those devices, and eventually impact reliant power system applications. We examine this problem in Chapter 6, where we identify and capitalize on a vulnerability in one of the methods used for phasor alignment at the PDC. We approach this problem using a linear program, and we consider system observability as the targeted power application. Through the presented model, we can identify an attack that is enough to prevent full observability of the power system, and thus open a window for an attack that leverages this weakness to initiate a cascading failure in the smart grid.

1.3 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 presents a brief overview of applications of precise timing in the smart grid along with the candidate time synchronization mechanisms for providing this precise timing, followed by a survey of the existing literature that addresses security concerns associated with PTP. Chapter 2 also presents a gap analysis for PTP that highlights open research problems that need to be addressed to secure PTP. Chapter 3 of this thesis introduces a shortcoming of the authentication scheme associated with PTP through its security extension, and identifies a potential amendment for this issue based on existing network and system management solutions. Chapter 4 addresses one of the well-known attacks against packet exchange based time synchronization protocols, the delay attack. We consider the use of PTP in a substation as recommended by IEC 61850 [57], the substation automation standard, to propose a detection and mitigation mechanism for the delay attack. We use formal model checking to evaluate relevant security properties of the proposed solution, and we demonstrate its usefulness on an actual implementation of the protocol. In Chapter 5, we build on top of the theory established in Chapter 4 to propose an extension for PTP that allows to collect messages from the network and analyze the collected information to assess the security posture of a PTP network. We once again use formal model checking and verification to validate the soundness of the proposed extension, and we demonstrate its usefulness using numerical simulation. Chapter 6 considers the wide area monitoring system as a scope of interest, and exploits the vulnerability of a data aggregation scheme followed by PDC to a time synchronization based attack. Through a linear program, we identify a PMU as an attack target along with an attack vector to be injected in its timing. As an outcome of this attack, the PDC receiving measurements from this PMU will drop phasors received from other benign PMUs. The outcome of this attack is formulated in terms of system observability, and the approach is demonstrated using hardware-in-the-loop simulation. Finally, Chapter 7 concludes the thesis and highlights potential research problems for future consideration.

Chapter 2

Preliminaries and Literature Review

In this chapter, we will overview time synchronization in the smart grid. Power system applications dependent on precise time will be presented. A review of mechanisms used for time synchronization will follow along with a security assessment of PTP. Standardization efforts related to our scope of interest will be shortly presented along with security related cuts. We conclude this chapter by carrying out a security gap analysis for PTP that highlights open research problems to be addressed.

2.1 Applications of Precise Time in Smart Grid

Critical applications in the smart grid require the presence of synchronized time across the infrastructure. These applications demand a common notion of time. Their measurements and monitored events need to be correctly aligned to enable proper actions and decisions. These actions define the self-healing characteristics of the smart grid. Indeed, providing real-time situational awareness to grid operators will decrease the impact of the outages by isolating the problem areas and avoiding the cascading events [110] such as transmission line fault detection/localization, and grid event locating.

Time synchronization is required for interoperability; the precision depends on the application and ranges from seconds to one microsecond [26] in various substation applications in the smart grid as Figure 2.1 shows.

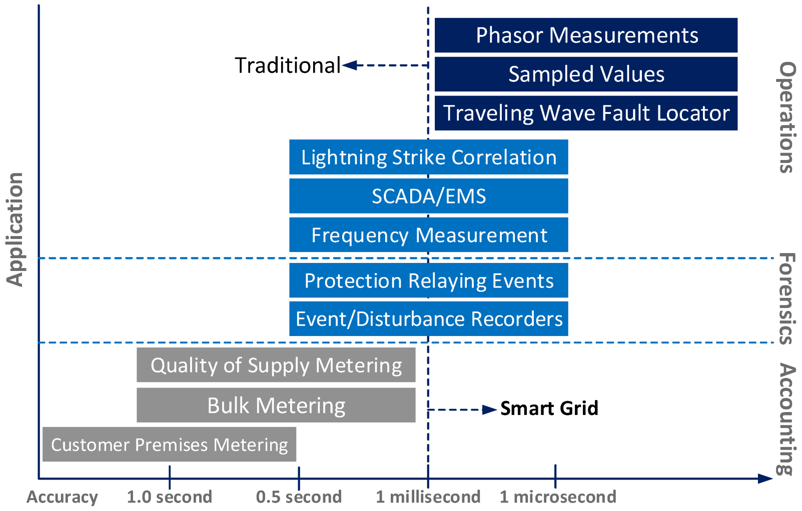

Figure 2.1: Timing requirements for substation applications [111].

Figure 2.1 states the accuracy requirements of various applications while contrasting the needs of the traditional grid to that of the smart one. Within a smart grid environment, accuracies in the order of few microseconds are needed. This presents a challenge in the choice of time synchronization mechanisms to be used.

We will overview those applications related to the smart grid and highlight their accuracy requirements starting with the broadly used synchrophasor measurements.

2.1.1 Synchrophasor Measurements

A synchrophasor is the estimate of magnitude and instantaneous phase angle of a signal function relative to the cosine function at nominal system frequency synchronized to UTC time [3]. Phasor measurement units (PMUs) are dispersed across the power system to supply real-time voltage and current synchrophasors. These readings are synchronized to absolute time, and used to analyze the state of the power system and maintain its stability. Synchrophasors increasingly contribute to the reliable and economical operation of power systems as real-time control and protection schemes become broadly used [49].

With a fixed temporal reference frame, synchrophasor measurements may be used to determine useful information about operation of the grid [15]. Compared to traditional SCADA measurements, synchrophasor measurements have higher sampling frequency, are able to provide direct measurement of power system states, and allow for more accurate monitoring of power systems and faster remedial actions [126]. Voltage stability monitoring, and stabilization of large disturbances rely on phasor measurements of voltage and current supplied by synchrophasors as pointed out in [79], [107] and [98]. PMU measurements play a fundamental role in power system state estimation as demonstrated by [51], [30], and [66]. Interested readers can refer to [34] and [13] for a survey on the usage of synchrophasor measurements in power system stabilizers among other applications.

A key requirement by synchrophasors is the precise time synchronization of PMUs that are sampling the readings across the power system. IEEE C37.118 standard [3] specifies that accuracy limits for the measurements shall not exceed a 1% total vector error (TVE). This translates into a maximal time error of ±31.8 or ±26.5 microseconds for 50 or 60 Hertz systems, respectively.

2.1.2 Disturbance/Fault Recording

Due to the complexity of the power grid, a disturbance taking place in one part of the grid affects operation elsewhere. When these interactions result in major events such as cascading failures and large blackouts, recording devices installed at various points in the grid generate large numbers of reports and data files [15].

To make sense of the collected files, there is a need to align the data recorded by several intelligent electronic devices (IEDs) at various locations to a common frame of reference. This data is used in post-event analysis. It allows to identify what happened where, and what happened when. It serves in finding the root cause of the disturbance, assessing the severity and duration of the fault, and taking any necessary remedial actions. The interpretation and alignment of fault records are eased by accurately time stamping the events during recording.

Recorded data from recording devices can be synchronized to assess the impact of a disturbance such as loss of generation, line trips, and loss of load. They have been installed in several power systems in North America [29]. A disturbance identification scheme to analyze the disturbance events recorded by those devices [29] and [127], accurately identifying the location of a fault upon its occurrence based on the integration of information available from disturbance recording devices [131], and rapid stability assessment of wide-are post-disturbance records [63] are available in the literature.

Nowadays, the trend is to equip all recorders with proper time synchronization. A one millisecond error is often regarded as sufficient for such applications [108].

2.1.3 Differential Protection

Numerical differential protection [132] works by evaluating Kirchhoff circuit law with current values obtained in numerical form from the different terminals involved. Depending on the principle, the current values can be delivered as phasors or as instantaneous values [132].

In a fault free system, the total of the currents is zero. The so-called differential current occurs in a faulty system when the currents are not balanced and their sum is distinct from zero. This is considered as a criteria for tripping [108]. However, differential relays support safety margins to account for time error among other possible errors. In protection systems, relays at the terminals of a differential protection line are synchronized. Current differential protection which utilizes wide-area current data will be effective for wide-area backup protection although such protection needs system-wide timing synchronism for the simultaneous current sampling at all remote terminals and data exchanges among them [102]. A fair time synchronization error is within the limits of 100μs.

2.1.4 Sampled Values

The IEC 61850 process bus involves the exchange of high-speed, real-time instantaneous voltage and current measurements using an Ethernet network [10]. Voltage and current sampled values are delivered to protection and control IEDs along with control commands sent to switchgear. These values are produced at high rates (typically 4 to 16 kHz). Merging Units continuously send sampled values of currents and voltages acquired from primary equipment. These digitized sampled values have to be received in synchronism by the relays so that the protection algorithm functions properly. Data shifted at the receiving IEDs by just 30 microseconds will result in half of degree phase angle error [77]. A technique to assess the overall network performance of sampled value process buses in IEC 61850 is presented by [59].

As Sampled Values (SV) are distributed to independent devices throughout the substation, time synchronization becomes critical for all applications that require data from multiple locations (e.g., bus differential protection) [101]. The demanded time synchronization precision is less than 1 microsecond.

2.1.5 Sequential Events Recorder (SER) Reports

This report contains a chronological order of the state changes of an IED through its operation (closing/opening a tele-protection contact output, alarms, logic status on internal logic elements) [10]. Such reports are produced by digital relays and event recorders. The availability of these reports is essential for troubleshooting the IED and observing its state changes. The required time accuracy is 1 millisecond.

2.1.6 Power System Fault Location

The traveling wave principle is used in power systems to locate faults on long transmission lines. This principle uses the time of arrival of fault generated waves at both ends of the transmission line to locate the fault. These waves reach the line ends at different times based on the fault location. Fault location is calculated from the time difference in arrivals and the speed of wave propagation in the line. This phenomenon is depicted in Figure 2.2. Using the speed of light as propagation speed, terminals A and B, supplied by accurate timing, exchange arrival information through a communication channel and locate the disruption targeting the transmission line. As the wave travels in the speed of light, a small timing error will result in large distance errors. Thus, a one microsecond time error results in fault location error of 300 meters. Fault location problem formulation and error calculation are presented in [108].

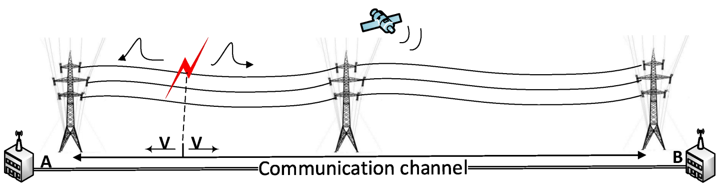

Figure 2.2: Traveling wave fault locating

Locating faults using the traveling wave principle received much interest from the power community. Early approaches on fault localization using digital relay data in the literature is available in [91] and [67]. With the introduction and wide use of PMUs in the power transmission system, the literature presented approaches relying on the measurements supplied by PMUs for fault localization as indicated by [61], [42], and more recently in [76]. A recent manuscript on using joint PMU and SCADA data for fault localization on a multiterminal transmission line is presented in [86].

After highlighting the critical time dependent applications in the smart grid, we will present the mechanisms used to provide the timing signal for use in power systems. Some of these mechanisms have been deployed in power systems for decades. However, as our overview shows, they are no longer suitable to meet the accuracy requirements in the modern power systems or their deployment and maintenance requires a separate infrastructure which is not favored by power utilities.

2.2 Time Distribution Mechanisms

The time distribution mechanisms we will discuss are the ones currently in use, and candidate for use in the future smart grid. We will outline the basic characteristics of these mechanisms, and the accuracy level they provide. This will filter the ones that do not satisfy the smart grid applications' accuracy requirements.

2.2.1 Pulse Per Second

A pulse per second (PPS) is an electrical signal of less than one second width, and a sharply rising or falling edge that accurately repeats once per second. PPS is considered a simple and accurate time distribution mechanism. A time server distributes pulses synchronized to the second rollover over a dedicated network to connected devices. This signal is limited by the care taken in the quality of the connection of the source to the device being synchronized [15]. PPS does not have the notion of absolute time, or clock changes (e.g. time of day). PPS is suitable for devices requiring synchronization within the second [108]. The supported accuracy is of the order of micro-seconds. However, its use in power applications is taken over by IRIG time codes. PPS is still commonly used in standards laboratories, to compare time and frequency sources at the highest level of accuracy [15].

2.2.2 IRIG-B

The IRIG time codes were originally developed by the Inter-Range Instrumentation Group (IRIG), part of the Range Commanders Council (RCC) of the US Army. The standard was first published in 1960 and has been revised several times by the Telecommunications and Timing Group (TTG) of the RCC [16].

The IRIG standard defines a family of serial time codes with different pulse rates. These codes use a continuous stream of binary data to transmit information on date and time. Each of these time code formats are distinguished by the signal characteristics (modulated, unmodulated), signal transmission techniques, data rate, and by the information carried in the transmitted data. Among the family of IRIG time codes, IRIG-B is the most known and used time format.

IRIG-B has a pulse rate of 100 pulses per second, through which it produces 100 bits of data. Out of these bits, 74 bits contain time, date, time changes, and time quality information of the time signal. IRIG-B code may be used in either logic-level (unmodulated) format, or as an amplitude-modulated signal with a 1 kHz carrier. IRIG-B presents time as a set of logical ones, zeroes, and position identifier bits. Connected IEDs to the IRIG-B service synchronize their clocks based on the data collected from this signal. IRIG-B has three functional groups of bits: Binary Coded Decimal (BCD), Control Functions (CF), and Straight Binary Seconds (SBS). IRIG-B supports time-of-year and year information in a BCD format, and an optional seconds-of-day in its SBS.

IRIG-B was extended in 2004 to use reserved bits of the CF part of the time code. Additional feautures such as calender year, leap seconds, daylight saving time, local time offset, time quality, parity and position identifiers are assigned to previously reserved bits in IRIG-B time code CF portion.

IRIG-B can be transmitted using various techniques either when its time code signal is moduled or unmodulated. Typical techniques for transmission of unmodulated IRIG-B include RS-485 differential signal over shielded twisted-pair cable, and RS-232 over shielded cable for short distances among others. For the transmission of modulated IRIG-B, coaxial cable terminated in 50 ohms or shielded twisted-pair cable can be used.

Most substation IEDs that accept the unmodulated IRIG time code use an optically-isolated input. This breaks ground loops, making possible direct connection througout a control room without excessive concern for grounding and potential differences. Such optocouplers only require a few milliamperes of input current, making it possible to connect many loads to a single IRIG-B driver [14].

IRIG-B supports accuracy of the order of microseconds and is currently used by electric utilities to provide time synchronization to critical power system devices such as protection relays, PMUs, and digital fault recorders (DFRs) [94]. However, IRIG-B code signaling is unidirectional, with minimal error checking capability (single parity bit) [112]. Undetected by receiving devices, the processing of bad IRIG-B time frames result in faulty time synchronization.

2.2.3 Network Time Protocol

The Network Time Protocol (NTP), defined in the RFC 5905 - Network Time Protocol Version 4: Protocol and Algorithms Specification [78], is a widely used time transfer protocol over data networks. Through a message exchange, a NTP daemon synchronizes the local device clock with that of one or more external reference time sources. Information included in the NTP message allows the daemon to determine the server time with respect to local time and adjust the local clock accordingly. In addition, the message includes information to calculate the expected timekeeping accuracy and reliability, as well as select the best server. This daemon plays the role of a client in collecting time references from servers. As a server, it can make its own time available as reference for other clients. Moreover, the daemon can be a peer in a comparison of different system times with other daemons before agreeing on a "true" system time to synchronize to.

These features can be used to set up a hierarchical time synchronization structure. Each of these hierarchical levels is labeled as a stratum. A smaller stratum number means a higher level in the hierarchy structure. The daemon with the most accurate time has the smallest stratum number and is located on top of the hierarchy.

Each NTP daemon can be configured to use several independent reference time sources. The daemon polls these sources periodically to classify them as good or bad sources. This aids the daemon in choosing a new system peer once its current peer becomes unavailable.

To achieve synchronization, NTP includes methods to estimate the round trip delay between the server and the client. NTP also ignores estimates that vary significantly from the typical delay values. NTP uses time clock drift estimation to compensate time deviation and provide time stability in the absence of the time source [94]. Accuracy level achieved by NTP depends on the performance of the devices' operating systems, and the nature of the connection between the client and the server. Best accuracy levels are achieved when the logical connection between client and server is kept as short as possible.

The achievable accuracy through NTP time synchronization is of the order of milliseconds. Thus, NTP does not guarantee the accuracy level required by merging units (MU) in a substation. However, most IEDs satisfied by NTP services use its simplified version, Simple NTP (SNTP). SNTP uses the same messages as NTP, and achieves the same accuracy level. However, it does not consider some algorithms that maintain clock stability over long periods of time.

NTP's standard level of performance is adequate to resolve the one-second ambiguity of a 1-PPS signal, so NTP and 1-PPS together make an acceptable method of accurate time synchronization in a substation [15]. However, this means that IEDs in a substation will receive time information over two connections. This is not feasible when compared to other available mechanisms and protocols especially PTP.

2.2.4 Precision Time Protocol

Precision Time Protocol (PTP) [4] is introduced in the IEEE 1588 standard as a candidate to fulfill the timing requirements of forthcoming systems. PTP allows heterogeneous systems that include clocks of various resolution, precision and stability to synchronize to a single time reference with a sub-microsecond accuracy [4]. Moreover, PTP power profile allows the usage of PTP for power system protection, control, and automation applications. PTP is recommended for time synchronization at the substation level by IEC 61850[57]. A more detailed overview of PTP will follow later.

2.2.5 Global Navigation Satellite Systems

Global Navigation Satellite Systems (GNSS) provide timing and location information for receivers over the globe. GNSS mainly consists of GPS[75], GLONASS[37], Galileo[68], and Beidou[1]. GNSS satellites are constantly transmitting signals which are collected and processed by receivers. Although GPS is the most widely used system, the services provided by these systems are similar. The timing accuracy achieved is below 1 microsecond and is most suitable for use by power systems protection and control applications. Thus, time distribution mechanisms use this timing signal to synchronize devices in a substation since it is infeasible to equip all devices with a GNSS receiver.

The accuracy levels supported by the presented mechanisms vary and thus their suitability for use in various substation applications. In Table 2.1, we present a comparison of the different capabilities and drawbacks of these mechanisms. The table also highlights the ability of these mechanisms to fulfill the timing requirements of various substation applications. As can be concluded from the table, PTP and GNSS are the most suitable mechanisms for time supply. They are both capable of meeting the accuracy requirements while not needing any dedicated network. Thus, power utilities rely on GPS-synchronized clocks to synchronize devices in substations, control centers, and distribution feeder circuits [10]. However, since the use of GNSS signals requires dedicated receivers, GNSS supplies the accurate timing to a designated IED in the network. This IED will play the role of a PTP master and distribute timing information to other IEDs connected to the already available Ethernet network.

Table 2.1: Summary of time distribution mechanisms and their fulfillment of smart grid applications requirements

Mechanism	**Typical Accuracy Level**	**Synchrophasor Measurements**	**Fault Recording**	**Differential Protection**	**Sampled Values**	**SER Reports**	**Fault Localization**
1-PPS	1 μs	✓	✓	✓	✓	✓	✓
IRIG-B	100 μs	×	✓	✓	×	✓	×
NTP/SNTP	1-10 ms	×	✓	×	×	✓	×
PTP	1 μs	✓	✓	✓	✓	✓	✓
GNSS	1 μs	✓	✓	✓	✓	✓	✓

2.3 Precision Time Protocol

2.3.1 Overview

Precision Time Protocol (PTP) [4] is a time-transfer protocol defined in the IEEE Standard 1588, a standard for a precision clock synchronization protocol for networked measurement and control systems [4]. It was developed to improve precision over current Ethernet protocols achieving a microsecond synchronization accuracy [4]. Moreover, PTP is capable of using the communication infrastructure available without the need to setup a PTP dedicated one.

PTP follows a packet-based message exchange approach to maintain time synchronization in the network. A designated time master sends periodic time-stamped messages to communicate accurate timing to the connected devices. Through the best master clock (BMC) algorithm, PTP establishes a master-slave hierarchy in the network. The established setup includes a single grand master clock (GMC), and a set of slave clocks synchronizing their time to that of the master. In addition to that, one can distinguish other types of clocks in the system with PTP defined functionality. Such clocks include boundary clocks and transparent clocks. Boundary clocks are used to maintain the timescales used in a domain. On the other hand, transparent clocks (TC) measure the residence time of a PTP event message at a TC, and supplies this information to recipients of the message in transit.

To achieve clock synchronization at slave devices, PTP provides two mechanisms: end-to-end synchronization, and peer-to-peer synchronization. Both mechanisms rely on synchronization messages sent by the GMC, yet they follow different approaches to measure the master to slave path delay. The end-to-end synchronization mechanism is depicted in Figure 2.3. As Figure 2.3 shows, four timestamps are collected by a slave. t_1 is the master time when the *Sync* message is sent, t_2 is the time the slave receives the *Sync* message, t_3 is the slave time when the slave sends a *Delay_Req* message, and t_4 is the time the master receives the *Delay_Req* message. Those timestamps define the trip time from the master to slave, t_{ms}, and slave to master, t_{sm}. The slave calculates the round trip path delay using the collected time stamps as equation (2.1) specifies. The calculated path delay is used in clock offset computation using equation (2.2). The slave uses this offset to update its clock as per equation (2.3), where $Time_{slave}$ and $Time_{slave}^{new}$ are the time at slave before and after synchronization respectively. At the end of this process, the slave clock is synchronized with that of the master.

$$Path\_Delay = \frac{(t_{ms} + t_{sm})}{2} = \frac{(t_4 - t_3) + (t_2 - t_1)}{2} \tag{2.1}$$

$$Clock\_Offset = (t_2 - t_1) - Path\_Delay \tag{2.2}$$

$$Time_{slave}^{new} = Time_{slave} - Clock\_Offset \tag{2.3}$$

The computation of offset and propagation time assumes that the master-to-slave

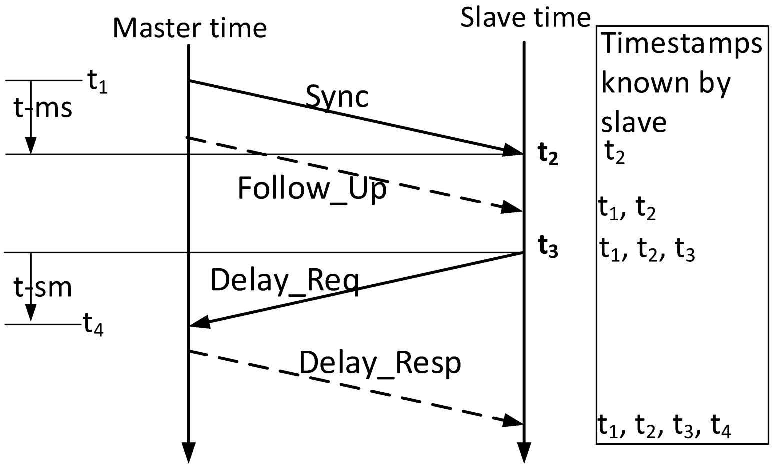

Figure 2.3: End-to-end synchronization message exchange [4].

and slave-to-master propagation times are equal. Any asymmetry in propagation time introduces an error in the computed value of the clock offset [4].

As for the peer-to-peer mechanism, the devices at the two ends of the link exchange timestamped messages (*Pdelay_Req*, and *Pdelay_Resp*) to collect the four timestamps (t_1^i, t_2^i, t_3^i, and t_4^i) associated with link i. Those timestamps are used to compute the link delay as in Eq. (2.4):

$$\lambda_i = \frac{(t_4^i - t_1^i) - (t_3^i - t_2^i)}{2} \tag{2.4}$$

where λ_i is the communication delay over link i, t_1^i and t_2^i are the timestamps associated with the *Pdelay_Req*, t_3^i and t_4^i are the timestamps associated with the *Pdelay_Resp*.

At intermediate links between the master and slaves, transparent clocks measure the message residence time. As depicted in Fig. 2.4, each message at TC$_i$, is associated with a residence time denoted by ρ_i (TC$_i$ refers to the TC at the receiving end of link i, and ρ_i at the slave clock is 0). The availability of TCs allows the replacement of random queuing delays at intermediate switches by a deterministic value measured for the sent *Sync* messages at each TC.

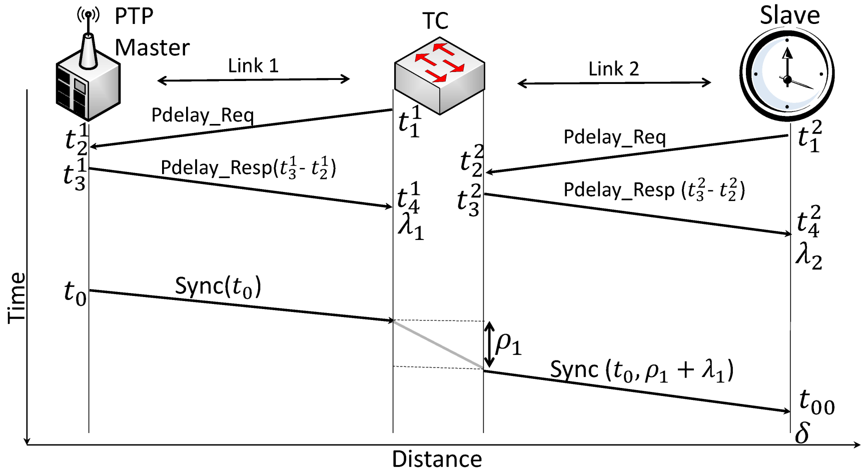

Figure 2.4: Slave synchronization using peer-delay mechanism and TCs [106].

To establish synchronization at the slave clocks, the master sends a periodic *Sync* message across the network. The *correctionField* in the *Sync* message header is updated at the intermediate TCs to report the incoming link delay λ_i, and ρ_i corresponding to the residence time of the *Sync* message.

The slave uses timestamps associated with the *Sync* message to calculate the offset using Eq. (2.5):

$$Clock\_Offset = t_{00} - t_0 - \sum_i (\rho_i + \lambda_i) \tag{2.5}$$

where t_0 is the *Sync* message egress timestamp at the master, and t_{00} is the *Sync* message ingress timestamp at the slave. Using the offset computed in Eq. (2.5), the slave synchronizes its clock to that of the master using Eq. (2.3).

Based on clock capabilities in the network, PTP either follows a one step or two step synchronization. One step synchronization is used when the sending clock is capable of providing an accurate timestamp that resembles the sending time of a message whether it was a synchronization message or a delay measurement message.

If that is not possible, clocks follow the two step synchronization where follow up messages are used to convey the accurate timestamp of the previously sent messages. Using either of the two approaches, the needed timestamps are collected at slave clocks, and used for accurate synchronization.

2.3.2 PTP Security Extension

A security extension, Annex K, was added to PTP to provide group source authentication, message integrity and replay attack protection for PTP messages. The extension specifies two mechanisms to achieve the security goals specified. The integrity protection mechanism verifies the source, integrity, and freshness of the received messages by using message authentication codes and counters. The challenge-response mechanism allows for the affirmation of new authenticated sources and the management of trusted relations.

It is worth noting that the implementation of this security extension is optional. Clocks requesting secure PTP message exchange indicate that by setting a flag bit in the message header to indicate the presence of the security authentication fields in the transported message.

2.4 PTP Security Assessment

PTP is vulnerable to a wide range of attacks targeting the services provided by the protocol. Systems relying on PTP time synchronization services suffer the impact of these attacks. We will next study those attacks, along with the countermeasures applied to detect and defend against those attacks as presented in the literature.

PTP secure functionality was the interest of much work in the literature before and after the introduction of Annex K extension. In[116], the authors provided a description of the security extension to PTP along with various attack points to target the PTP network and an attack targeting the master election algorithm. In [81], the authors discussed the delay attack in time synchronization through a game theoretic approach and suggests using multiple paths between master and slave clocks for time synchronization to mitigate the risks of that attack and its applicability to PTP. In [97], the authors studied the so-called selective packet delay attack on PTP to identify the fields of the message that need to be compromised to carry the attack. They study the presence of fake masters in the network and their effect on clock synchronization at slaves as well. Figure 2.5 presents a look at the attacks targeting time synchronization under PTP. Others study vulnerabilities present in the design of PTP along with the attacks targeting its specifications. The literature is summarized in Tables 2.4 and 2.5 and will be divided over three parts; approaches targeting PTP use in power grid, others discussing issues in the design of PTP, or suggesting usage of already available secure mechanisms in implementing PTP functionality and enhancing its security through the use of Transport Layer Security (TLS) [41] and Internet Protocol Security (IPsec)[17] along with approaches discussing the suitable algorithms for PTP message authentication codes (MACs).

2.4.1 PTP in power grid

The use of PTP in power grid systems is analyzed in [115] where the authors provide an approach to enable determining clock drift at electric devices when a connection inside the substation is broken resulting in desynchronization between the master clock and its slaves, a phenomenon known as islanding. They start their approach by summarizing the various threats targeting time synchronization. They sum up their discussion by dividing these threats into two categories, either the slaves are aware of their desynchronized clocks after the attack or they are unaware. They discuss the

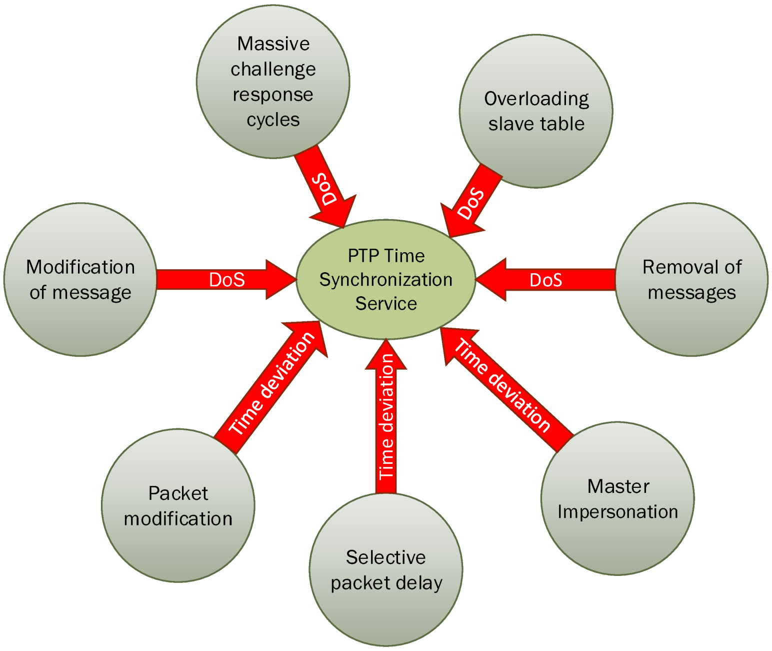

Figure 2.5: Attacks on PTP time synchronization service

countermeasures to these threats according to the annex K security recommendations. Their main contribution is the presentation of three techniques to calculate clock drift based on historical data collected during normal operation. The first approach predicts the clock drift by calculating the average drift over the past N observations. The second is probabilistic-based where only drift values with a probability greater than a specified threshold over the past N observations are included in the prediction. The prediction in the third approach is based on an auto-regressive model used to identify a pattern of the drift in the historical data. These approaches provide different accuracy levels in drift prediction. However, the choice between these approaches depends on the computing capabilities of the node and computation time needed.

2.4.2 Design

Through the analysis of PTP design, several vulnerabilities were pointed out in the literature. The authors in [119, 50, 117, 120, 125] expose PTP weakness, improvise

the applicable attacks, and suggest countermeasures to guard against them.
The discovered attacks include denial of service (DoS), byzantine master, interruption of control loop, removal of packets from control loop, packet manipulation, packet insertion, replay attack, and selective packet delay attack[119, 50].

The security properties of PTP before the introduction of the optional Annex K extension are investigated by Gaderer et al. in [50], while Tsang et al. in [119] provide a compilation of attacks targeting PTP. The output presented is a set of attacks applicable to a PTP network where the optional security extension, Annex K, is not deployed. To carry on these attacks, the authors assume that the adversary has access to the network, can monitor, collect and analyze exchanged messages. The attacker targets the master, slave clocks, and the control loop. The effect of his attacks range from introducing incorrect offset to the slave clocks, to complete control of the time synchronization mechanism and the prevention of this mechanism through DoS.
The countermeasures presented to defend against these attacks are addressed in the annex K extension of PTP especially the ones related to authenticating the nodes, protecting message integrity and preventing replay attacks. Other countermeasures [50] include using cryptographic techniques along with QoS monitoring to protect against these attacks. It is worth noting that, Annex K security extension mitigate most of the causes of these attacks. However, other attacks such as selective packet delay is still a threat targeting time synchronization under PTP.

In computing the master-to-slave path delay, PTP assumes that the communication network is symmetric. This assumption can be targeted through what is known as delay attack, or selective packet delay attack. This attack is pointed out in [119, 50, 120, 125], but is formulated and studied in [120, 125]. Indeed, Ullmann et al. [120] indicate the vulnerability of PTP to delay introduced in the communication channel. This delay affects the accuracy of path delay calculation based on the arrival time of synchronization and delay request messages. A similar approach is followed by Yang et al. [125] where the attack model presented defines a man-in-the-middle capable of introducing a random time resembling quantity in path delay calculation by manipulating the master-to-slave and slave-to-master message exchange. The attack analysis shows that delaying the synchronization message sent by the master affects all the slaves in the network while delaying request messages affects only the slave sending the message. The analysis quantifies the error in the offset calculation in terms of the introduced delay. Such an attack succeeds in jeopardizing the time synchronization mechanism and is hard to detect by the involved parties.

To defend against such attacks, the authors in [120] suggest implementing specific network security mechanisms to protect the network, and monitoring the usual propagation delays across the network. They also point out that security mechanisms ensuring the authentication of the communicating nodes and the integrity of this communication do not counter the described attacks. On the other hand, Yang et al. [125] propose a detection mechanism based on hypothesis testing. Their hypothesis monitors the ratio of the master clock to the slave local clock. They assume that the system is under attack when this ratio exceeds a threshold specified by 1 micro second. However, deploying such a mechanism means that the slave and master know each other clock values which is unrealistic in a PTP network; or the existence of a monitoring entity capable of observing the clock values at various nodes in the network.

PTP specifications are targeted by Treytl et al. in [117] to reveal vulnerabilities in the specifications. The identified weaknesses assume the presence of a man-in-the-middle capable of capturing and modifying the exchanged synchronization messages. The first flaw allows the attacker to modify the source and destination in the time synchronization messages. This is eased by the fact that those addresses are not included in the calculation of the integrity check value (ICV) at the receiver side. The attacker can create forged security associations and change the clock value at the slave side. The authors suggest using a source port identity field in the PTP header to retrieve a unique source address which is included in the calculation of the ICV. The second flaw is related to the presence of transparent clocks in the IEEE 802.1 network, and the fact that those clocks modify the MAC address present in the exchanged messages. Transparent clocks terminate the incoming link and create a new frame with the PTP payload, the source MAC address of the outgoing bridge and a modified correction field. Security associations relying on unmodified source protocol address will discard such frames. The authors suggest creating double entries for the core functions of the standard containing the master MAC address along with that of the last transparent clock or use the source port identity field in security associations. However, the authors indicate that to target PTP with the above indicated flaws is implementation dependent, and would require a brute force breakdown of the random value of the lifetime field available in the requests. Their practical applications is related to the feasibility of that brute force attack.

Recently, Moreira et al. [85] performed a comparison of the proposed approaches to secure PTP in its future version based on the discussion in the standardization committee, and proposed a hop-by-hop group authentication and integrity solution using MACsec and IEEE 802.1X standards. On the other hand, Narula et al. [89] established a fundamental theory for secure clock synchronization. The authors found that PTP is not secure based on the necessary and sufficient conditions of this theory, and they presented a specialization of specific conditions for PTP secure clock synchronization. Those conditions include the availability of an authenticated encryption scheme between the communicating parties, negligible difference between forward and backward path delays, and a previous knowledge of this path delay. Although those conditions provide a secure version of PTP, their applicability is subject to much concerns especially when it comes to a previous knowledge of the estimated path delays in a local area network. An analysis of PTP security is provided by Itkin et al. in [60] where the authors exploit PTP vulnerability to a multitude of threat models and subsequent attacks. The presented attacks can be carried by in-band and out of band weak and skillful attackers targeting the communicated messages and various entities of the PTP network. The described attacks are addressed with suitable security mechanisms, and a revised PTP security extension is proposed. However, even with a revised extension, PTP will remain vulnerable to attacks due to its nature and design considerations namely the delay attack performed by an in-band attacker.

The study of PTP showed some vulnerabilities in its design as the previously discussed works show [120, 125, 117]. However, the most critical among these vulnerabilities is the delay attack. Such an attack is hard to differentiate from network congestion and delays, difficult to detect, and succeeds in targeting PTP slaves. The countermeasures presented neither detect nor prevent this attack [120], or base their detection on non-realistic assumptions [125]. Moreover, although the issues raised by Treytl in [117] threaten PTP functionality, the feasibility of using such issues to affect time synchronization under PTP can be questioned. However, these efforts present a good starting point for assessment of PTP security based on its design.

2.4.3 Implementation

To secure PTP services, the use of IPsec and MACsec is investigated in [118] and [80]. While Mizrahi et al. [80] presented a threat analysis in the presence of IPsec and MACsec, Treytl et al. analyzed the impact of the use of IPsec tunnels on PTP clock synchronization and the accuracy levels it provides in [118]. Indeed, the author in [80] described the common IPsec and MACsec deployment scenarios and present a subsequent threat analysis. In this analysis, internal and external attackers are considered. These attackers are capable of intercepting and manipulating the exchanged messages, capturing and injecting messages into the network. The attacks presented target the integrity and authentication functions. Packet injection and manipulation, spoofing, replay attacks, rogue master, packet interception and removal, packet delay manipulation, layer 2 and/or 3, DoS, and time source spoofing are the attacks discussed. The applicability of these attacks to networks protected by IPsec, MACsec, or Annex K specifications is presented. These attacks result in slave nodes aligning with a false time value or inability to synchronize their clocks. The author concluded that a hybrid approach deploying a combination of these mechanisms can securely support PTP operations.

On the other hand, IPsec effect on PTP time synchronization is compared by Treytl et al. [118] to that of the native security measures in IEEE 1588[4]. IPsec usage is illustrated in the protocol stack used to synchronize PTP clocks. A unit for message protection using cryptographic operations is introduced at the network layer in the transmit path along with another for the verification of incoming messages. Additionally, two security state machines responsible for the security management are directly integrated in the IP stack. The analysis shows that the delay presented by IPsec results from the use of encryption algorithms, packet size, and security schemes. The performed measurements show that the jitter introduced is noticeable over the receive path compared to little jitter on the send path. The results contrast the use of IPsec to that of unprotected IP. The authors analyzed the use of a hardware timestamping unit to eliminate the pre-mentioned jitter. They concluded that the use of a MAC based timestamper for IPsec with an adjusted clock frequency of the SHA unit can be an effective solution. This modification is a limiting factor in embedded systems and might affect IEEE 1588 clock synchronization over IPsec.

2.4.4 PTP MACs

Algorithms used for generating PTP message authentication codes (MACs) are covered in [93, 83, 84]. As a conclusion, these papers suggest an alternative MAC protocol implementation that can satisfy the need of PTP authentication security and provide better performance than the ones specified in PTP Annex K extension.

Indeed, in [93], the authors point out that the use of HMAC-SHA256 MAC specified in Annex K is suboptimal in terms of delay resulting from MAC calculations. Based on testing of other MAC protocols implementation, they suggest the use of Chained MAC (CMAC) and claim that it allows on-the-fly calculation of the MAC. Another major modification they suggest to the Annex K is dropping the three-way handshake and replace it by a one-way authentication. As a node joins the network, it shall send periodic authenticated supervision frames to introduce itself to other nodes. These nodes can validate the authenticity of these frames by checking the ICV value. This approach makes use of the pre-shared keys in the network and avoids the additional overhead caused by message exchange in the three-way handshake. In addition, the authors find that the used sequence numbers are too short for effective protection against replay attacks and suggest using absolute time instead. Through their thorough analysis of Annex K, they suggest other modifications to its specifications. These alterations require removal of some parts (challenge-response exchange, security association update exchange, etc.), modification of the ICV test to start from the destination protocol address rather than the PTP header, a collective replay protection mechanism that uses a 48-bit register, and changes in the security association structure, secure message transmitting, transparent clock rules, shared key distribution and authentication TLV. The suggested modification prove that there is enough room for modifying the Annex K and enhancing its efficiency. However, Moriera et al. in [83] and [84], demonstrated the feasibility of using SHA-3 (KECCAK) as the MAC function in PTP message security. Their study is based on a comparison between AES-128 and SHA-3 where the hardware implementation of SHA-3 provides the same security level and latency but with lower area consumption.

The literature review exposed the threats associated with the use of PTP for time synchronization. Although there are much countermeasures proposed, some of these threats (as the case with PTP delay attack) rise as a main concern for any system relying on PTP to synchronize its devices. In an upcoming section, we highlight the gaps threatening PTP secure functionality and we address some of them in the upcoming chapters.

2.5 Standardization Efforts

This section covers the developed standards that guide the time synchronization in the smart grid. First, we will cover the PTP power profile which aims at specifying PTP options to be implemented in clocks for use in power industry. It also specifies the default values for a set of PTP attributes to suit power industry requirements. Second, IEC standardization efforts related to time synchronization in power substations will be covered. This mainly includes IEC 61850-5[55] and IEC 61850-90-4[56].

2.5.1 IEEE C37.238

The second release of PTP in 2008 included the definition of new devices (e.g., transparent clocks) along with a set of attribute options (e.g., transport over the IEEE 802.3 Ethernet or UDP) and optional features (e.g., unicast messaging). However, aiming at ease of setup with minimal administration, the concept of PTP profile was introduced to identify a set of required features and assign default values for attributes based on the needs of industry. Thus, the profile specifies a subset of the protocol features to be implemented based on specific industry requirements.

A PTP power profile customized for power system applications is introduced in the IEEE C37.238 [5]. This profile defines PTP features and attributes for use in power system protection, control, automation, and data communication applications utilizing an Ethernet communications architecture. The profile specifies a well-defined subset of PTP mechanisms and settings aimed at enabling device interoperability between different vendors, robust response to network failures, and deterministic control of delivered time quality [5].

Among the profile specifications, the IEEE 802.3 Ethernet is specified as the preferred physical layer for PTP related communication and parameters configuration. The profile also specifies the use of peer-to-peer delay mechanism in measuring the propagation delay over the communication link. It recommends using one-step operation in communicating time information in the network. However, it also allows using two-step operation for less expensive silicon solutions. An overview of main PTP power profile specifications are available in Table 2.2.

Table 2.2: IEEE C37.238 main profile specifications

Profile Option	**Value**
BMCA	Default BMCA
Transport	IEEE 802.3/Ethernet
Delay mechanism	Peer delay only
Management	SNMP MIB (mandatory for grandmaster-capable devices only)

The profile defines strict requirements to ensure the time accuracy and quality required by substation applications. It demands that inaccuracy introduced by a transparent clock must not exceed 50 nanoseconds. This allows achieving an accuracy of one microsecond by a slave clock connected to the GMC over 16 network hops. The profile also specifies that there should be at least two or three devices in the network capable of being GMC in case the later fails. Finally, SNMP Management Information Base (MIB) use is specified for configuration and status messages.

2.5.2 IEC 61850

The IEC 61850[57] standard is developed to make substation automation interoperable and cost-efficient. It was designed to operate over modern networking technologies. The standard ensures interoperability in power systems among many other features. In Part 5 [55], IEC 61850 covers communication requirements for functions and device models. And Part 90-4, technical report, network engineering guidelines for Ethernet networks are presented. Among the contents of Part 5 and Part 90-4, issues related to time synchronization in substation and power systems in general are presented.

Through these two parts, the IEC 61850 defines time models and time synchronization requirements at the substation level. It targets the synchronization of precise clocks at various levels of substation automation, and aims at specifying required accuracy levels for various events(e.g., time-stamped measurements, sequence-of-events). The standard specifies the need for only one time base in substation, and a unified time tagging format for all devices in the power system. It specifies the use of absolute time synchronization for synchrophasors while relative time synchronization is used for protection functions.

The IEC 61850-5 defines different synchronization classes related to the application using the time signal. These classes are indicated in table 2.3. According to table 2.3, accuracy requirements vary from $\pm 1\mu$ sec for protection functions to ±1 msec for event logging.

Table 2.3: Synchronization classes of IEC61850-5[57]

Class	Accuracy	Usage
T1	±1 ms	Event logging
T2	±100 μs	Zero crossing for the distributed synchrocheck. Time tags to support point on wave switching
T3	±25 μs	Class P1 protection functions
T4	±4 μs	Class P2 protection functions (e.g. busbar protection function). Time tagging of samples
T5	±1 μs	Class P3 protection functions and high precision time tagging of samples

Time synchronization specifications presented by IEC 61850 can be summarized as follows:

- A dedicated time server present in the substation receives time signal from an external source outside the substation (GNSS, long-wave radio, etc).

- In case of absolute time usage, two time servers of different types must be available.

- For PTP use in substation:

  - Only layer 2 communication and peer-to-peer delay can achieve the required accuracy.

  - Alternate master option is recommended for implementation.

  - Hold over time for slave is 5 seconds in case of master failure.

  - Reference clock should be located on the station bus, and used for synchronization of devices on the process bus as well.

- The time signal for time synchronization shall be easily derived from a global time reference system like GPS.

- The time tag of the transmitted binary or analogue events/values shall be as accurate as possible for post-fault/failure event sequence analysis.

- The time tag of the transmitted binary or analogue events/values shall need no correction at the receiver.

- The time synchronization procedure shall fulfill the performance classes shown in Table 2.3 as far as applicable.

- Time synchronization telegrams should use the same communication infrastructure as the data exchange to facilitate the system design.

2.5.3 PTP Gap Analysis

Although the Annex K extension provided solutions for message integrity and replay attack protection, the examination of PTP with Annex K reveals security gaps that need to be analyzed and mitigated to enhance the secure behavior of PTP. Among these gaps we highlight:

- Availability of PTP: when Annex K specifications are violated, the violating message is silently discarded. This opens a window for a denial of service attack. An attacker interested in interrupting PTP functionality may intercept and modify the exchanged messages, which will be eventually discarded. This attack can target messages used in clock offset calculation and management of security associations resulting in denial of time synchronization in the targeted networks.
To mitigate this vulnerability, a strategy to allow clock updates based on past experience at the slave could be defined. This strategy is to be used upon interruption of time synchronization.

- Design: We highlight the following gaps in PTP design.

1. Secure and non-secure PTP coexistence: in the presence of slaves using non-secure PTP and communicating with a master using secure PTP in the same network, the slaves will be able to hear *Announce* and *Sync* messages sent by the master. However, the master will simply ignore *Delay_Req* messages sent by slaves not using secure PTP and thus will deny them to correctly synchronize their time. PTP lacks a security policy to handle such situations. This is of much importance if PTP is to be used over deployed devices with limited resources while avoiding to change these devices. As a mitigation approach, a device capable of interfacing slaves using PTP with Annex K disabled to a master clock using secure PTP can be used. Such a device will play the role of a master clock on the slaves side, and obtain accurate timing as a slave from the grandmaster clock.

2. Delay Attack: PTP relies on the measurement of path delays in the network and assumes a symmetric path connecting the GMC to each of the slaves. This assumption is a vulnerability that allows an attacker to target PTP services even in the presence of security measures. Such an attack constitutes a major threat to PTP since it does not violate authentication, integrity nor confidentiality constraints. We will target this attack and provide suitable detection and mitigation approaches in a PTP network.

3. Open loop protocol: PTP lacks knowledge about the synchronization status of connected slaves. The only output provided by the slaves are the messages sent to measure the path delay which are not enough to convey any information about the status of the source clock nor are used for this purpose. Thus, in the presence of an attack or malfunctioning of the network, the impact of such an incident will go unnoticed since it only affects slave clocks. Hence, there is a need to close the PTP loop and enable the collection of timestamped messages from slave clocks to evaluate their synchronization status. We will tackle this problem and use this feedback collection

to detect attacks targeting PTP synchronization services.

- Resource handling: as pointed by one of the surveyed work in the literature[93], the three way handshaking mechanism specified by PTP Annex K is resource consuming. An alternative for that would be the deployment of an authentication mechanism that allows master/slave clocks to authenticate in a single step when needed thus avoiding network congestion and saving its resources.

- Key establishment: PTP Annex K does not specify a key establishment mechanism and considers it out of its scope. Shared keys specify the MAC protocols to be used and trigger the maintenance of established secure associations. Thus, a mechanism to securely distribute and manage the shared keys between slave, master, and transparent clocks is needed.

- Annex K Authentication: PTP Annex K deploys a group source authentication scheme which makes it vulnerable to an in-band attacker. A successful compromise of a PTP clock allows an attacker to generate fake *Sync* messages and manipulate the network time synchronization. We will address this problem using available network and system management solutions while introducing minimum modifications to the existing PTP deployment.

Table 2.4: Summary of PTP security analysis from the literature

Classification	Description	Advantage	Disadvantage	Reference
Design	Attacks targeting PTP including message tampering, DoS, delay and replay attacks	Presentation of attacks on PTP and suggestion of countermeasures	No evaluation of the countermeasures' effect on PTP functionality	[119]
	Delay attack on time synchronization targeting NTP and PTP	Quantitative evaluation of the delay attack consequences on PTP	Absence of experimental verification of the attack, its effects, and the suggested countermeasures	[120]
	Time desynchronization attack on PTP managed networks	Analysis of the delay attack effects on PTP synchronization	Assumptions made for suggested countermeasures pose a large overhead on the master side	[125]
	Study of PTP security aspects before the introduction of Annex K extension	Classification of attacks on PTP and attack surfaces in a PTP network	Absence of technical evaluation of the suggested countermeasures	[50]
	Vulnerabilities in PTP specifications	Study of PTP security in presence of transparent clocks	No experimental evaluation for the described attacks and countermeasures	[117]
	Analysis of the delay attack using a game theoretic approach	A different approach in analyzing the delay attack	Use of multiple path in clock synchronization does not guarantee the path symmetry needed by PTP	[81]
	Lab simulation of the selective packet delay attack	Simulating the delay attack in a lab environment	Simulation of fake masters in the network is irrelevant to performing PTP delay attack	[97]
	Security analysis and revised security extension for PTP	Experimenting the different attacks and security strategies	Fails to address all attacks and propose protocol messages modifications	[60]

Table 2.5: Summary of PTP security analysis from the literature (cont.)

Classification	Description	Advantage	Disadvantage	Reference
Implementation	Use of IPsec and MACsec to secure PTP functionality	Setting up IPsec and MACsec in a PTP network, then analyzing PTP security	No analysis of the effect of the approach on PTP functionality	[80]
	Use of IPsec tunnels and their impact on clock precision in PTP	Analysis of the effect of IPsec use on PTP time synchronization	Suggested approach introduces modifications to the PTP stack	[118]
PTP MACs	Analysis of mac generation code algorithms to replace the HAMC-SHA256 Annex K specification	Discussion on the usage of new MAC algorithms for securing PTP traffic	No analysis on the computational capabilities needed by PTP devices to implement these approaches	[93, 83, 84]
PTP in Power Grid	Maintaining time synchronization and managing clock drifts in electric substations during islanding	Presentation of different synchronization strategies to suit devices computational capabilities	No indication of how long those strategies will be effective in maintaining synchronization	[115]

Chapter 3

Securing The Precision Time Protocol Against Fake Timestamps

Time distribution mechanisms favored for use in the smart grid, such as PTP, were not designed with security in mind, and thus suffer several security vulnerabilities. PTP is vulnerable to fake timestamp attack through master impersonation and *Sync* message injection. Such an attack will synchronize clocks to a false time reference. In this chapter, we consider an IEC 61850 substation, and propose an approach to detect fake timestamps communicated through false PTP *Sync* messages. This approach builds on top of existing network and system management (NSM) solutions. We introduce new SNMP data objects to monitor PTP functionality, and detect the existence of fake timestamps in a PTP synchronized network. We implement the approach on a testbed, and comment on the collected results.

3.1 Introduction

The nature of the smart grid as a distributed, complex, and connected system gives rise to the need for innovative solutions to monitor and react to the grid dynamics in real time. Sensors and advanced measurement systems such as Phasor Measurement Units (PMUs) dispersed across the domains of the grid serve to meet this need, and enable the collection of measurements that reflect the status of the grid. However, the alignment of the collected measurements and coordination of the performed actions is not possible without a unique pulse that governs this operation. Through time synchronization to a unified time reference, a pulse is introduced to the smart grid, and thus enables real time observability of the grid dynamics.

Indeed, in the next-generation "smart grid" infrastructure, accurate timing signals will be broadly required - from generation plants to distribution substations to individual smart grid components [96]. The increased demand for better reliability in the grid resulted in higher accuracy requirements, and was reflected through standards that defined the needs of various grid applications. IEC 61850 [57] detailed the synchronization classes for different applications with accuracy requirements of the order of microseconds. Those specifications resulted in favoring the Precision Time Protocol (PTP) [4] as a time distribution mechanism at the substation level over other available mechanisms.

PTP, as defined in IEEE 1588, is recommended for use at the substation level for its ability to achieve synchronization accuracy of the order of microseconds [4]. Moreover, PTP is advantageous due to its use of the existing communication infrastructure without any need for dedicated cabling. However, the security of PTP remains a big concern. PTP is subject to various cyber attacks that target its availability, and impact the achieved synchronization accuracy at end devices [87]. Even in the presence of the security extension, Annex-K, PTP is still subject to some attacks such as the delay attack [120], [88] among others. Moreover, the introduction of Annex K does not address all vulnerabilities in PTP specifications. In particular, the authentication scheme proposed by Annex K is not sufficient to prevent a connected device from spoofing PTP messages.

In this chapter, we address PTP vulnerability to the presence of false messages injected through a compromised network component. The availability of false synchronization messages, injected by a threat agent and circulating in the network, impacts the time accuracy at synchronized devices. The usage of PTP at the substation level leaves the smart grid vulnerable to such an attack, and thus there is a need to address this issue. However, introducing changes to existing deployments and implementations of PTP is not favored by utilities. Thus, we address this concern using existing deployments and software solutions at the substation level. We leverage network and management solutions (NSM) as presented by IEC 62351-7 [58] to propose a detection mechanism for the existence of fake timestamps injected using false messages on the behalf of the PTP master.

3.1.1 Novel contributions

The main contributions of this chapter can be highlighted as follows:

1. We formulate the fake timestamp injection attack against PTP, and highlight the vulnerability of PTP networks to this attack through a compromised PTP device in the presence of PTP - Annex K.

2. We devise a detection approach through exploiting NSM solutions as mandated by IEC 62351-7. The presented detection approach leverages unicast security associations established through SNMPv3 to communicate data relevant to the received Sync messages and the timestamps they carry.

3. We define new SNMP data types and objects as part of PTP MIB. The introduced definitions complement the detection approach, and enable the collection of relevant attack information through SNMP requests and responses.

4. We analyze the overhead associated with the approach in terms of latency and network traffic using a NSM testbed resembling the IEC 61850 substation. The performed analysis demonstrates the applicability and usefulness of the presented detection mechanism.

The remainder of this chapter is structured as follows. System model is discussed in Section 3.2. Section 3.3 details our attacker model. The attack detection approach is presented in Section 3.4, while Section 3.5 provides a theoretical evaluation of the approach. Experimental results are covered in Section 3.6. Concluding remarks are provided in Section 3.7.

3.2 System Model

The system under consideration is a power substation as mandated by the IEC 61850 standard [57]. Accurate timing for IEDs available in the substation is distributed through the Precision Time Protocol (PTP) [4] as defined in IEEE 1588-2008. In its current version, through the optional Annex K, PTP uses multicast authentication to authenticate messages sent by the GMC using symmetric key encryption. Using a secret key, the GMC computes a hash based integrity check value (ICV) over the complete *Sync* message, and appends that value to the *Sync* message. Other types of clocks in the network including slave, transparent and boundary clocks, share the same secret key used by the GMC. Those devices verify the received ICV using the shared key, and decide to drop or use the synchronization message based on the ICV test.

On the other hand, network and system management (NSM) [58] in an IEC 61850 substation is made possible through the deployment of SNMPv3 [104]. SNMPv3 uses a unicast security association to maintain secure communication between the SNMP

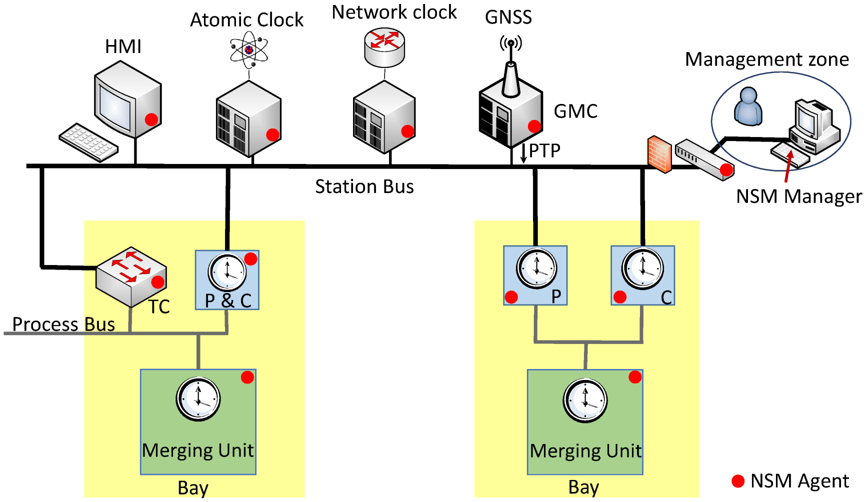

Figure 3.1: Schema of IEC 61850 substation with NSM entities.

manager and agents. Hence, each of the agents associated with the PTP clocks, available in the system depicted in Fig. 3.1, communicates with the manager using a secret key different from those used by the other slaves. This enables the manager to verify the authenticity of reports sent by any of the agents.

3.2.1 Problem Definition

Precision Time Protocol (PTP) gained a lot of interest from different systems due to its success in achieving high levels of accuracy in clock synchronization. However, PTP was not standardized with security as a core requirement. The protocol includes an optional security extension, Annex K, to provide group source authentication, message integrity, and replay attack protection. Through the proposed message authentication scheme, a secret key is shared among all devices in the network. The GMC uses this key to sign *Sync* messages among others, while other devices use this key to verify received messages' source and integrity. in our work, we assume that the optional Annex-K security extension is implemented.

However, the proposed multicast authentication scheme, based on symmetric key encryption, puts the entire PTP functionality at risk when the secret key is disclosed. Indeed, if an attacker manages to break into a single device and retrieve the authentication key, the attacker can prepare *Sync* messages on behalf of the GMC, generate an ICV using the disclosed key, and reshape the network synchronization.

Devices receiving the Sync messages sent by the attacker on behalf of the GMC will verify the message integrity, extract the timestamp it carries, and later use the timestamp value to synchronize their time to what they believe is the time of the GMC. Hence, those devices are not capable of identifying false *Sync* messages carrying fake timestamps that will manipulate their synchronization.

Therefore, the current authentication scheme does not meet the expectations of securing PTP in challenging environments. An attacker exploiting this vulnerability can manipulate time synchronization services, and impact other system functionality relying on accurate timing. Moreover, introducing changes to the protocol is not favored nor does it fix the security concerns with the existing PTP deployments. Thus, we need to detect the presence of fake *Sync* messages to better protect PTP against cyber attacks.

3.3 Threat Model

We consider a Dolev-Yao attacker [43] who gains access to the system after PTP clocks elect the GMC. We assume that the PTP GMC and the SNMP manager are trustworthy and can not be compromised. The attacker can break into one or more slave clocks to initiate his attack. By breaking into a device, the attacker has full control over this device, and gains access to the multicast key used in PTP communication along with the unicast key used by SNMP. Moreover, our attacker is capable of dropping, injecting, and delaying messages everywhere in the network. In particular, the attacker is interested in spoofing slave clocks through the injection of fake *Sync*

messages. To carry on this attack, the attacker will intercept and drop the GMC *Sync* messages, impersonate the GMC, inject new *Sync* messages with a fake timestamp, and generate an ICV for those messages using the PTP authentication key available at the compromised device and the slave clock ability to generate timestamped messages. The attacker is also capable of communicating with the SNMP manager on behalf of the device he is in control of.
On the other hand, the attacker performs this attack rather than a more complicated one as he aims at eventually damaging the system while being undetected.

3.4 Approach

Our approach is centered around the interests of the system operator who wants to detect the insertion of *Sync* messages with fake timestamps, while avoiding any changes to the synchronization protocol. To this end, and in the context of an IEC 61850 substation, we will leverage the existing infrastructure and deployed NSM solutions to improve PTP security, and prevent the manipulation of time synchronization through fake *Sync* messages.

3.4.1 Feedback Introduction

PTP, in its current version, lacks a mechanism to monitor the protocol performance at slave devices. PTP adopts a master-slave approach for synchronization, where the master issues timestamps through Sync *messages.* Slave clocks collect those messages and use the timestamps they carry for clock synchronization.

To monitor the protocol performance and received PTP messages at slave clocks, we will introduce a feedback mechanism into the network. Through this mechanism, slave clocks will report to a reference entity information pertaining to the received

Sync messages. In a similar manner, the GMC will provide feedback to the same reference entity. Thus, an analysis of the received feedback will enable this entity with the capacity to detect mismatches between the information sent by the master in *Sync* messages, and that received by the slave clocks through the same messages.

Moreover, to prevent a possible compromise of the feedback channel, we need to associate proper security mechanisms with this channel. Thus, there is need for a secure unicast association to deliver the feedback from different clocks to the reference entity. Such a mechanism is already established in the substation through the deployed SNMP for NSM as mandated by IEC 62351-7 [58]. Hence, we will assign the reference entity role to the SNMP manager which already communicates with SNMP agents located on all substation devices (see Fig. 3.1). Through SNMP communication, agents answer requests and report to the manager on PTP activity including contents of *Sync* messages issued by the GMC. In addition to that, the SNMP agent located at the GMC will report to the SNMP manager information pertaining to the *Sync* messages sent over the network. Through the collection of the communicated information, the SNMP manager is capable of detecting anomalies between the GMC *Sync* messages and the ones received at slave clocks.

3.4.2 Feedback Contents

The feedback collected from agents available at clocks in the network aim at detecting the presence of malicious *Sync* messages. To enable this detection, we will periodically send requests, and collect responses from the GMC and other PTP clocks. To identify those messages, we will need the sequence numbers and timestamps of *Sync* messages sent by the GMC, and those of *Sync* messages handled by other clocks in the network. This information is collected through requests issued to agents at the GMC and slave clocks.

3.4.2.1 Agent at GMC clock

This agent will periodically respond to the manager's request by sending a message containing a table composed of rows, each row contains the sequence number and timestamp of a single *Sync* message. Each response sent by the agent includes information about recent *Sync* messages that are not reported yet to the manager. Through these responses, the manager will not miss any timestamp issued by the GMC. The collected responses provide the manager with a complete view of the GMC activity, and allow it to detect the presence of fake *Sync* messages. To fulfill this exchange, we will use the objects as defined in Appendix A.

3.4.2.2 Agent at slave clock

Agents located at different slave clocks will receive requests and generate responses pertaining to the recently received *Sync* messages. Each request demands a block of n consecutive *Sync* messages. The sent response includes the sequence number of the first *Sync* message in the reported block, a bitmask of size n representing the consecutive sequence numbers of the *Sync* messages reported in this block, and a hash of the timestamps carried in those messages using SHA3 [45] with 256 bit output. The bitmask identifies the received and reported sequence numbers using a bit set to 1. Moreover, the agent uses a zero timestamp for the missed *Sync* messages (identified through a 0 bit in the bitmask). This allows the manager to identify the sequence numbers of the reported *Sync* messages, and prepare the needed timestamps hash value for comparison. To enable this exchange, we will use the objects as defined in Appendix A.

It is worth noting that the manager is capable of associating those messages with their sources. Moreover, this communication is protected through unicast security associations already established using SNMPv3. Such associations prevent an attacker

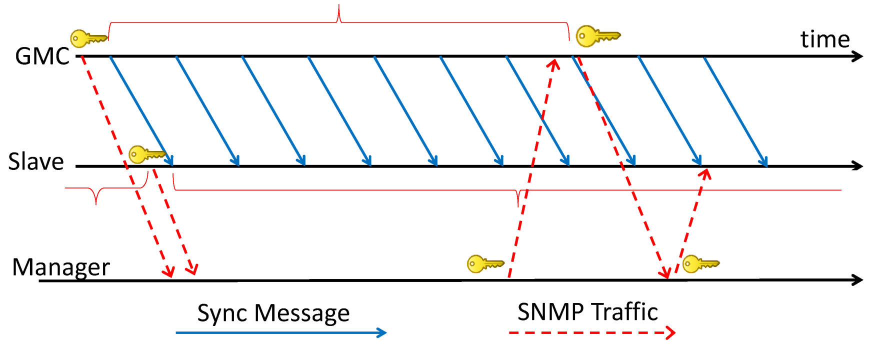

Figure 3.2: Feedback collection based on *Sync* messages

from spoofing the manager by generating fake SNMP responses.

3.4.3 Periodicity of Feedback

The feedback collection from GMC and other clocks happens based on the manager's requests frequency as seen in Fig. 3.2. Taking into consideration that the PTP power profile specifies that *Sync* messages are to be sent once every second, the manager can send requests to collect information from the agent at the GMC every 10 seconds. Each response will include a table of 10 rows pertaining to the sent *Sync* messages. On the other hand, the bitmask field introduced in Section 3.4.4, and used by agents at slave clocks, is composed of 16 bits and thus allows to report up to 16 *Sync* messages. The size of the bitmask, and the periodicity of requests issued to the GMC agent, can be subject to updates based on the overhead introduced to the network through this communication.

3.4.4 SNMP Introduced Objects

To realize the detection of fake timestamps communicated through fake *Sync* messages, we will define and introduce new objects as an add-on to the PTP Management Information Base (MIB) defined in the PTP power profile [6]. This includes defining new types such as *PTPClockTimestamp*, *ptpSyncSeqNumber*, *ptpSeqBitmask*, *ptpTimestampHash* as shown in Appendix A.

The newly introduced types are used to define structures and objects that will hold the information of relevance for the detection. Those new objects include *ptpTimestampTable*, *ptpSeqTSEntry*, *ptpTSHashTable*, and *ptpTSEntry*. We list those objects under the *ieeeC37238Objects* for each clock supporting the IEEE C37.238 profile [6], and assign them an identification number. This associates instances of those objects to clock ports in slave or master state. Through the assigned identities, those objects can be requested by the manager.

The introduced objects constitute of two tables to hold relevant information for detection approach. The first table, *ptpTimestampTable*, is composed of entries where each entry contains the sequence number and timestamp pertaining to a single *Sync* message sent by the GMC. This table is requested from the agent at the GMC. The second table, *ptpTSHashTable*, is used by agents at the slave clocks to store information on blocks of received *Sync* messages. Each entry in this table contains sequence number of the first *Sync* message in block, the bitmask identifying the *Sync* messages in block, and a hash of the timestamps for these messages.

3.4.5 Detection Logic

Our proposed detection approach is performed by the manager. The manager has access to the data available at the agent side through requests sent to agents, and responses issued by agents to answer the manager's requests.

The manager collects *ptpTimestampTable* from the GMC, and is able to associate *Sync* messages sequence number with the timestamps they carried. Later, through *ptpTSHashTable* collected from slave ports, the manager can identify the reported timestamps using the sequence number corresponding to the first *Sync* message identified through the *firstSeqNum* field, and the bit mask identified through the *seqNumBitmask* field. The next step will be to compute the hash of the identified timestamps, and compare it to the *hashOfTimestamps* as reported by the slave. The manager detects the presence of a fake timestamp if the compared hash values are not equal.

3.5 Evaluation

In this section, we analyze the introduced network overhead through the added SNMP requests and responses. We discuss the expected false alert rates as well.

3.5.1 Network Overhead

Through the introduction of new requests and responses into the NSM communication, we add more load on the substation network. Taking into consideration the criticality level of network availability and performance in the substation, we present a theoretical computation of the additional SNMP traffic injected into the network.

Each SNMP messages is composed of two parts a header and payload. The header includes fields with fixed size (Message Version Number, Message Identifier, etc.) and variable sizes (Context Engine Id, Context Name, etc.). The payload has a similar structure however it carries identifications of MIB objects (as in case of requests), and values stored in those objects (as in the case of responses). Thus, the additional traffic has fixed-size components and others with variable size. We will consider the worst case for the variable size parts.

The fixed size fields in SNMPv3 header has asize of 25 Bytes, while the variable size fields can reach a maximum of 192 Bytes. Thus, the maximum header size is 217

Bytes. On the other hand, the fixed size fields in the payload have a size of 16 Bytes. As for the variable size fields we will distinguish between two cases:

1. *SNMPv3 Request:* The request payload includes the variable bindings which identifies the requested MIB objects and a value of NULL associated with them. The manager can request the *ptpTimestampTable*, and *ptpTSHashTable* MIB objects from the agents located at GMC and other clocks in the system respectively. Thus, the size of the variable fields can sum up to 12 or 10 Bytes. Therefore, the maximum size of the introduced request is $217 + 16 + 12 = 245$ Bytes.

2. *SNMPv3 Response:* The response has a similar structure to that of the request. However, the response carries a combination of the requested objects and the values they carry. We will have two different responses sent over the network, one sent by agent at GMC clock and another sent by other agents.

- *Response from GMC agent:* This response carries a table of 10 rows identifying the *Sync* messages sequence number and the timestamps they carried. Thus, this response size exceeds that of the request by $10 \times$ row_size
where row_size = sizeof (PTPClockTimestamp) + sizeof(PTPSyncSeqNumber) $= 6 + 2 = 8$ Bytes. Thus, the total response size is $10 \times 8 + 245 = 325$ Bytes.

- *Response from other agents:* This response carries a table entry of 3 rows identifying a block of *Sync* messages containing firstSeqNum, seqNumBitmask, and hashOfTimestamps. Thus, this response size exceeds that of the request by the size of the previously indicated objects. Hence, this reponse size is sizeof(request) + sizeof(firstSeqNum) + sizeof(seqNumBitmask) + sizeof(hashOfTimestamps) $= 245 + 2 + 2 + 32 = 281$ Bytes.

Noting that this traffic is periodic as discussed in Section 3.4.3, the network will witness the following augmentation in traffic:

1. *Traffic addressed to GMC agent:* This includes a request and matching response of total size $245 + 325 = 570$ Bytes once every 8 seconds.

2. *Traffic addressed to other agents:* This includes all the requests sent to agents along with their matching responses. If we consider the total number of agents in the network to be N, then the total traffic size is $N \times (245+281) = 526 \times N$ Bytes. Thus, in a network composed of 1000 devices, we may witness an overhead of approximately 500KB every 16 seconds.

In a substation environment, the IEC 61850-90-4 recommends the usage of 100 Mbit/s links to connect end nodes to bridges, and 1 Gbit/s links for bridge-to-bridge connections [106]. Therefore, the introduced approach adds negligible traffic compared to the network data rate.

3.5.2 Flase positives and negatives

False positives in the detection approach are due to reporting a mismatch between the collected data from the GMC agent and other agents in the absence of fake timestamps. However, this contradicts the deterministic property of a hash function where the same input to a hash function always results in the same hash. Thus, using the timestamps collected from GMC agent, and respecting the bitmasks received from other agents, we will get the same hash value as communicated by agents on slave clocks.

On the other hand, false negatives represent cases where the manager can not detect the presence of fake timestamps as reported by the agents. In this case, the manager will match fake timestamps reported by the agents to those sent by the GMC agent, and will not find any difference in the resulting hash values. However, this contradicts the collision resistance property of SHA3 hash function where different

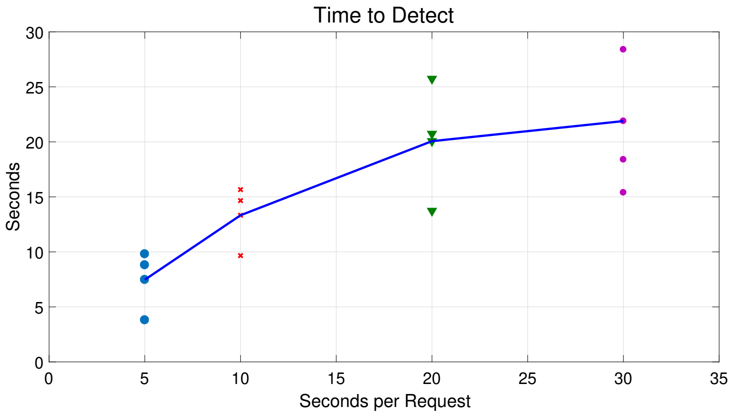

Figure 3.3: Detection time based on request period.

messages result in same hash. Thus, hash values computed from fake timestamps will not match the hash values of legitimate timestamps.

3.6 Experimental Results

To validate the usefulness of our approach, we deployed PTP daemon (PTPd) [8] on a testbed resembling the IEC 61850 substation. Two machines of the testbed function as PTP master and PTP slave, and has a deployed SNMPv3 agent with the updated MIB objects that communicates with an SNMPv3 manager deployed on a third machine. We configured PTP to send 1 *Sync* message every second.

In our experiments, we run PTP on the network for 300 seconds and we execute the fake timestamp injection attack at three random instances. We vary the period at which the SNMP manager sends requests to PTP devices, and we evaluate the latency in attack detection. The collected results are presented in Fig. 3.3. The selected periods at which the manager sends requests to agents are 5, 10, 20 and 30 seconds. With the increase in the request period, the time to detect the attack increases as indicated by the average latency plotted as a line in Fig. 3.3. Fig. 3.3 shows the results for all the tests as well, which indicates that the best and worse

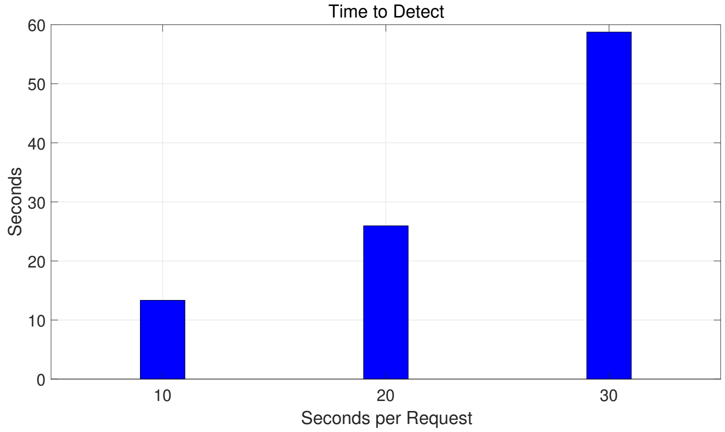

Figure 3.4: Detection time based on variable request size and period.

latency increase with the increase in the request period. This gives an advantage of short request periods in detecting the attack at an early time.

In another set of experiments, using the same settings as before, we varied the number of records reported by the agents at different PTP devices along with the request period. With the variation in the number of records, we chose a request period that matches the response size since sending requests at a higher rate than the agents can respond will introduce an overhead in the network without collecting any outcome from those requests. The variation in the response size did not have a positive impact on the detection latency, since more information need to be communicated to the manager before actually detecting the attack. Thus, the best performance is ensured through a short request period as for the case of 10 seconds shown in Fig. 3.4.

Finally, we report on the network overhead imposed by the proposed approach. In Fig. 3.5, we plot the additional traffic due to PTP-based SNMP requests and responses for different request periods. The presented graph shows the average additional traffic over a period of 100 seconds in a network composed of one manager and two agents. As can be seen in Fig. 3.5, the collected results match our theoretical analysis and the introduced overhead is minimal compared to the collected outcome

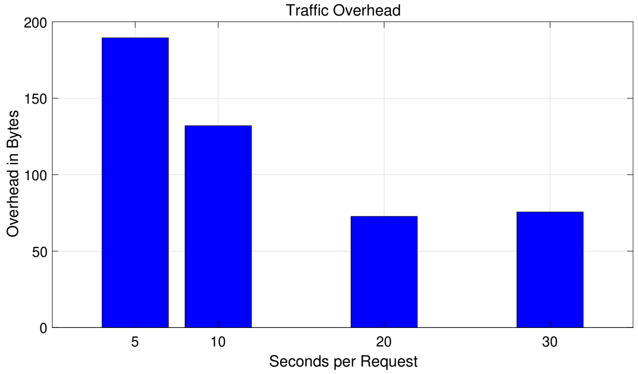

Figure 3.5: Traffic overhead based on request period.

and network capabilities.

As an outcome of the collected results, and based on a combination of the time to detect and network overhead, we recommend using a short request period of 5 seconds for the manager to agent communication.

3.7 Conclusion

In this chapter, we addressed one of the shortcomings of the authentication protocol used by Annex K to secure PTP operations in a network. We leveraged the availability of secure SNMPv3 as recommended by IEC 62351-7 for a substation to harden PTP security against attacks that exploit the authentication scheme, and inject fake timestamps to target synchronization at slave clocks. We defined new SNMP objects, and demonstrated their use for effective detection of fake timestamps in a PTP network. Results collected demonstrated the effectiveness of the approach, and the light overhead imposed on the network in the form on additional traffic.

Chapter 4

Detection and Mitigation of PTP Delay Attack in IEC 61850 Substation

In the previous chapter, we devised a solution for the injection of fake timestamps through a compromised device into the PTP network. In this chapter, we again consider an IEC 61850 based substation to devise a detection and mitigation approach for the well-know delay attack. We formally model, and validate the proposed approach, and we provide experimental results from tests run on a physical system.

4.1 Introduction

Time synchronization is a core requirement in modern systems. The functionality of such systems demands a common notion of time and rely on the ability to align their various activities to an accurate time scale. Indeed, precise time has become a critical component of systems/applications integrated in our daily life such as electrical power systems, telecommunications systems, and networking systems [101].

The smart grid, a forthcoming cyber-physical system, is characterized by its critical operations and their effect on our daily life. The services provided by the smart grid demand the presence of an accurate time signal suitable for the alignment of its various events. Indeed, modern components are or will be introduced to monitor and manage the grid. These components timestamp the observations they make about the grid status and the data they collect about its conditions. Phasor measurement units (PMUs), sensors, merging units, intelligent electronic devices are dominant across the domains of the grid, mainly at the substation level as can be illustrated in Fig. 4.1. These devices reflect the real-time state of the grid among other functions such as protective line measurements, and analog measurements. The accurately timestamped measurements they provide affect the observability of the grid, the state estimation, and voltage stability [92]. Moreover, accurate timing in the smart grid is a requirement for disturbance recording, sequential events recorder (SER) reports, power system fault location, and sampled measured values.

Several protocols (Network Time Protocol (NTP), Inter Range Instrumentation Group-B (IRIG-B)) have emerged to align these systems to a specific time reference. These protocols do not guarantee that the time-synchronized nodes in the distributed system advance synchronously. Instead, they guarantee that the difference in time values of the clocks of different nodes is bounded [39]. Clock values at different nodes do not necessarily step/advance at the same pace. However, when synchronized, their values adjust within a certain acceptable offset. NTP [78] and IRIG-B [105] are examples of such mechanisms used to provide timing solutions for a range of systems. However, these mechanisms fail to support the requirements of the smart grid applications that demand an accuracy in the order of 1 microsecond or less [101].

Precision Time Protocol [4] is introduced in the IEEE 1588 standard as a candidate to fulfill the timing requirements of forthcoming systems. PTP allows systems that include clocks of various resolution, precision and stability to synchronize to a single time reference with a sub-microsecond accuracy [4].

The security of PTP has attracted researchers' attention recently and is the subject of various efforts [119, 50, 120, 125]. While the literature includes some methodologies targeting security aspects of PTP, in this work we leverage model checking and formal verification of temporal logic properties to quantitatively and qualitatively evaluate the security of PTP. Model checking is an automated, algorithmic method to systematically explore all possible executions of a model to check if it satisfies the required specifications [27]. We will employ this technique on PRISM [53], a probabilistic model checker, to study the functionality of PTP, formally prove the vulnerability of PTP to the delay attack, propose a delay-attack detection model and a mitigation mechanism, and formally prove the effectiveness of these mechanisms in detecting this attack and mitigating its effects.

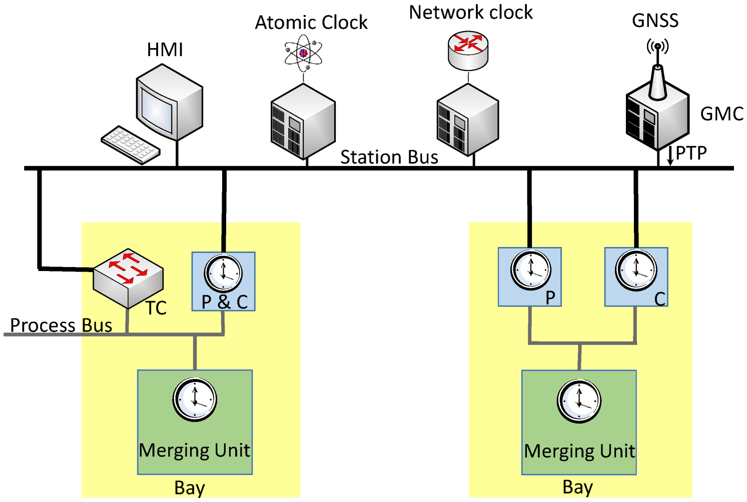

Figure 4.1: An IEC 61850 smart grid substation architecture

The presented model is the first to target detection of delay attacks. The upcoming sections will illustrate its difference from other approaches targeting PTP delay attacks, and will justify its usefulness for use in a substation in smart grid.

4.1.1 Novel Contributions

This chapter presents a model that leverages the system capabilities to enable the detection of the delay attack against PTP. The proposed model introduces modifications into PTP slave functionality to mitigate the attack impact upon detection. The main contributions are detailed as follows:

1. We develop a formal model of PTP time synchronization mechanism under the PRISM model checker. The model considers two step PTP message exchange and clock synchronization. Using this model, we can use PCTL quantitative and qualitative properties to verify PTP behavior and functionality in the presence of the proposed mechanism.

2. We formally prove the existence of the delay attack and its effect on PTP time synchronization. We establish theory for the detection of the delay attack, and falsify the claims that rely on PTP offset calculations to detect the attack. On top of the established theory, we present the detection logic while considering PTP usage in a substation.

3. We modify the slave functionality and its usage of the calculated offsets, and present a mitigation methodology centered around retaining and analyzing historical synchronization records at slave clocks to recover the impact of the attack and maintain the slave synchronization status.

4. We formally validate the presented techniques through probabilistic properties using PRISM model checker. And we experimentally evaluate their effectiveness using a PTP deployment on a physical system.

The remainder of this chapter is structured as follows. Section 4.2 surveys a selection of the existing literature and highlights the differences that distinguish the work presented in this chapter from its existing counterparts. Section 4.3 presents the system under study. Section 4.4 details our threat model and provides an analysis of the targeted attack. The developed model for PTP is presented in Section 4.5. Sections 4.6 and 4.7 detail the proposed detection and mitigation models respectively. Experimental results are provided in Section 4.8, and Section 4.9 presents concluding remarks.

4.2 Related Work

The literature is rich with papers covering PTP security. PTP security aspects and attacks targeting PTP are covered in [119] and [50]. The delay attack is formulated and discussed in [120], while [125] present the time desynchronization attack in power systems centered around delay asymmetry in PTP communication paths. Those approaches do not provide any quantification of the security of PTP. Their analysis is based on PTP specifications and implementation. Moreover, to detect the delay attack, these papers ([119], [120], [125]) suggest monitoring changes in computed delay and offset at the slave side as an indication of an ongoing attack without suggesting any mitigation techniques. However, in our upcoming analysis, we show that relying on changes in the computed values at the slave side is not enough to detect the delay attack. Moreover, we present a mitigation approach to counteract the effects of the delay attack.

On the other hand, PTP models were built by researchers to verify its functionality or validate properties related to the best master clock algorithm (BMC) used in PTP master election. Indeed, the BMC algorithm was modeled by [40] and the built model is used to perform systematic testing. BMC behavior is also analyzed in [27] based on discrete event simulation and model checking. Through their model analysis, the authors deny future suggested modification to BMC functionality. Moreover, a PTP model was built by [22]. The built model is studied to verify the suitability of PTP for clock synchronization in large heterogeneous systems. Their study targeted the effect of clock drift and scheduling policies in the network on time synchronization as well. Basu et. al [22] succeed in deriving jitter-related precise bounds that guarantee proper synchronization in a large heterogeneous system.

The main difference between our approach and the ones in the literature is that we quantify security properties of PTP using model checking rather than just exposing PTP vulnerabilities. We also propose modification to the PTP standard by defining a new clock type capable of detecting the delay attack, and adjusting the functionality of the slave clocks to introduce a mitigation strategy that maintains synchronization in the presence of the delay attack. We also verify PTP security-related properties. The verified properties demonstrate the effectiveness of the proposed detection and mitigation mechanisms to protect and support PTP time synchronization.

4.3 System Model

The system under consideration is an IEC 61850 [57] substation and can be best depicted as shown in Fig. 4.1. The communication medium (staion bus, process bus) is an Ethernet bus as standardized by IEC 61850. The grandmaster clock is located on the station bus as the standard specifies. It receives the accurate time signal through a GPS connection or any timing mechanism satisfying the timing requirements. Using PTP multicast, the master supplies time synchronization signal to all devices connected to the station bus. A PTP request-response mechanism is deployed for path delay measurement and clock synchronization over the station bus.

IEC 61850-90-4 [56] recommends that bridges used in the substation communication network are PTP transparent clocks to limit the effects of queuing time on synchronization. The maximum number of transparent clocks over a path is specified by 8 to avoid accumulation of synchronization errors introduced by these clocks. IEC 61850-90-4 also specifies that, in a substation, the asymmetry over a link between two transparent clocks must be limited to 50 ns. In addition to that, IEC 61850-90-4 demands the use of the peer delay request response mechanism (see Fig. 2.4 for details).

The IEC 61850-90-4 standard requires the availability of a backup master clock which can replace the grandmaster upon any failure. The standard does not specify the backup master time source, but it indicates that the backup timing signal must satisfy the timing requirements of the substation. Hence, the IEC 61850 substation architecture limits the asymmetry in the network to few nanoseconds, and ensures the availability of an alternative accurate time source.

On the abstraction level using PRISM, the master and slave clocks are represented as peers. They are connected directly without the use of an intermediate switch. This does not violate the specifications of IEC 61850, and it simplifies our task of validating the model and verifying the usability of the proposed delay attack detection mechanism.

4.4 Threat Model

Our goal is to detect an ongoing delay attack targeting PTP time synchronization in a substation, and enable PTP slaves with an effective mitigation approach to recover the attack effects and maintain time synchronization at the slave devices.

We assume that the attacker is either an external entity or an insider with malicious intents. He has the expertise to perform long-term reconnaissance operations required to learn the environment and execute a highly synchronized, multistage, multisite attack [71]. The attacker is aware that the target substation uses PTP-based time synchronization, and is aware of the topology of the substation. Moreover, the attacker can identify the master clock on the network, and is capable of introducing the required delay into the PTP message exchange paths either through software or hardware.

Our attacker is interested in targeting the functionality of all the devices in the network rather than a particular IED. He will manipulate the clock of the connected devices by introducing a variable delay into the PTP master communication path. If the attacker wishes to introduce the delay through introduction of additional hardware to the network, he has the expertise necessary to choose suitable attack locations to introduce undetectable attack and avoid the surveillance available at the substation.

Moreover, our attacker aims at conducting a stealthy attack while remaining undetected. Thus, he has interest in conducting the delay attack, and controlling synchronization at slave devices rather than performing other more impactful attacks which might flag detection alerts at the substation.

4.4.1 Delay Attack Tree

Our attack model is formulated in the attack tree presented in Fig. 4.2. To perform the attack, the attacker must obtain access to the master communication on the station bus. The attacker may be an insider such as a disgruntled employee, or physically break into the substation. This gives the attacker an advantage to introduce the Undetectable Delay Box [19] to perform the delay attack. The Delay Box is a device capable of producing a delay of a few microseconds or more either in the forward or in the backwards direction relative to the master clock. It operates at the physical layer and is therefore undetected by any encryption or authentication mechanism at layer 2 or above [19].

On the other hand, access to the station bus can be also attained by the help of an insider who either grants physical access to the attacker or infects the system with a special crafted malware. As an alternative for physical access to the station bus, an attacker can spoof an operator using a forged web-page and install the required malware in the system [54].

Once the malware is inside the system, it targets devices connected to the station bus and identifies the PTP grand master clock. The malware infects the master clock, targets its network activity, and modifies the procedure followed by the master to timestamp and send *Sync* messages over the network. By altering this mechanism, the malware introduces chosen delay after the *Sync* message is timestamped and before it is sent over the network. This is referred to in the attack tree in Fig. 4.2 by delay *Sync* messages by δ.

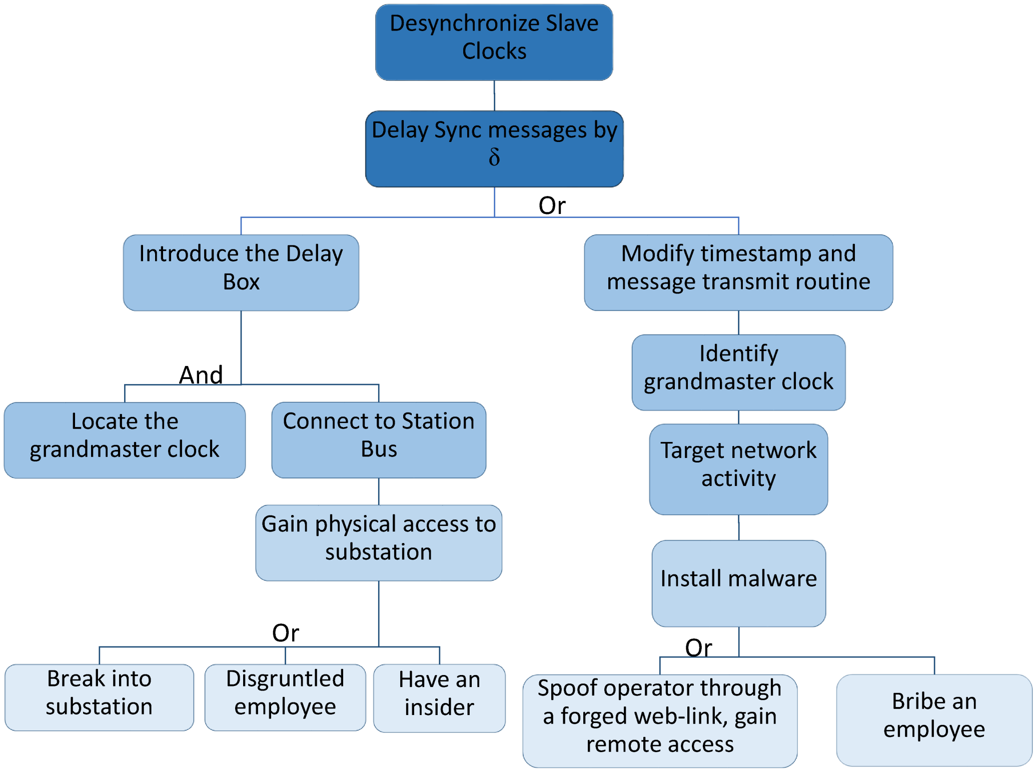
Desynchronize Slave Clocks
Delay Sync messages by δ
Or
Introduce the Delay Box
Modify timestamp and message transmit routine
And
Locate the grandmaster clock
Connect to Station Bus
Identify grandmaster clock
Target network activity
Gain physical access to substation
Install malware
Or
Break into substation
Disgruntled employee
Have an insider
Or
Spoof operator through a forged web-link, gain remote access
Bribe an employee

Figure 4.2: Delay attack tree

This process is clearly depicted in the attack tree. The specified goal is to desynchronize slaves in the network, and this is possible by following the indicated paths. It is worth noting that, by carrying this attack on time synchronization and with an incremental increase in δ, an attacker can control and alter the affected slave clocks while avoiding detection mechanisms.

4.4.2 Delay Attack Analysis

Through applying the delay to selected messages, mainly the *Sync* messages, the attacker performs the selective packet delay attack. The introduced delay, δ, aims at violating the path symmetry assumed by PTP.

To model this attack, we need to introduce additional delay on the communication path connecting the master to slaves. Thus, instead of reaching its destination at time t_2 in case of *Sync* message, the delayed message reaches at $t_2 + \delta$ affected by the additional delay introduced by the adversary, δ (see Fig. 2.3). The delay, δ, introduced will result in an incorrect offset and hence an inaccurate synchronization at the slave side. Using Eq. (2.1), the effect of introducing a delay δ on the master to slave path is shown in equation (4.1).

$$\begin{aligned} \mathit{Offset}_{attack} &= \frac{(t_2' - t_1) - (t_4 - t_3)}{2} = \frac{((t_2 + \delta) - t_1) - (t_4 - t_3)}{2} \\ &= \frac{(t_2 - t_1) - (t_4 - t_3) + \delta}{2} = \frac{(t_2 - t_1) - (t_4 - t_3)}{2} + \frac{\delta}{2} \\ &= \mathit{Offset}_{noattack} + \frac{\delta}{2} \end{aligned} \tag{4.1}$$

where t_2' is the timestamp of the delayed *Sync* message, Offset_{attack} and $\mathit{Offset}_{noattack}$ are the calculated offsets in the presence and absence of an attack respectively.

Thus, at the end of a synchronization event under attack a slave will update its clock value using the offset from equation (4.1), and the relation presented in equation (2.3) which results in a $\frac{\delta}{2}$ clock synchronization error.

Remark 4.1. The selective packet delay attack is difficult to detect by observing changes in offset at the slave clock.

Proof Using the offset in equation (4.1) and the formula specified in equation (2.3), the slave clock value after the first synchronization event can be represented as:

$$t_{slave}^{new} = t_{master} - \frac{\delta}{2} \tag{4.2}$$

where t_{slave}^{new} is the slave time after synchronization, and t_{master} is the time at master at time of offset calculation and δ is the delay introduced by the attacker.

Consider the start of a second synchronization event, the master sends a *Sync* message timestamped by t_1 which is delayed by δ before reaching the slave. The slave collects t_2, t_3, and t_4 as PTP specifies.

Let $t_1 = t_{master_1}$ then $t_2 = t_{slave} + \delta + \epsilon_1$

where t_{master_1} is the time at master when the *Sync* message is timestamped, t_{slave} is the time at slave when *Sync* message is sent, and ϵ_1 is master-slave path delay. Using equation (4.2), we get:

$$\begin{aligned} t_2 &= t_{master_1} - \frac{\delta}{2} + \mathit{Offset}_{noattack} + \delta + \epsilon_1 \\ &= t_{master_1} + \frac{\delta}{2} + \mathit{Offset}_{noattack} + \epsilon_1 \end{aligned} \tag{4.3}$$

Moreover, assuming the slave sends a delay request immediately after receiving the *Sync* message, we calculate t_3

$$t_3 = t_2 = t_{master_1} + \frac{\delta}{2} + \mathit{Offset}_{noattack} + \epsilon_1 \tag{4.4}$$

The master receives the delay request after an equivalent of a round-trip time since sending the sync message, thus:

$$t_4 = t_{master_1} + \delta + \epsilon_1 + \epsilon_2 \approx t_{master_1} + \delta + 2\epsilon_1 \tag{4.5}$$

where ϵ_2 is the slave-master path delay. Therefore, calculating the offset using equation (2.2) we get:

$$\begin{aligned}
\mathit{Offset} &= \frac{(t_2 - t_1) - (t_4 - t_3)}{2} \\
&\approx \frac{(t_{master_1} + \frac{\delta}{2} + \mathit{Offset}_{noattack} + \epsilon_1 - t_{master_1})}{2} \\
&- \frac{(t_{master_1} + \delta + 2\epsilon_1 - (t_{master_1} + \frac{\delta}{2} + \mathit{Offset}_{noattack} + \epsilon_1))}{2} \\
&= \mathit{Offset}_{noattack}
\end{aligned}$$

Thus, to introduce clock inaccuracy at the slave node, a single execution of the delay attack is enough by the attacker. However, that inaccuracy is resolved in the upcoming synchronization event. Therefore, to maintain the attack effect on the slave clock, the attacker must keep on delaying the master messages. Meanwhile, slave nodes behave as if time synchronization is successfully achieved. Thus, a single change in the offset is not enough to detect the delay attack.

4.5 PTP Model

PTP networks consist of a master node with an accurate timing and slave nodes with clocks that suffer a drift from the time reference due to various factors (e.g. quality of the oscillator, humidity, temperature, etc.). Other types of nodes such as transparent clocks might be available depending on the network architecture. However, we consider the simple case of a network formed of a master and slave nodes at the substation level as in Fig. 4.1.

We model the delay request-response mechanism in an IEC 61850 substation. The approach we are presenting can be easily extended to cover other types of networks and their respective PTP mechanisms. The mechanism is modeled as a Markov chain using the PRISM model checker. The network system we are studying exhibits non-deterministic and probabilistic behavior. This distributed system formed of components running in parallel with an undetermined interleaving of their actions is non-deterministic by nature. Moreover, the slave clock is modeled as a Bernoulli random variable to effectively introduce a clock drift.

4.5.1 PTP Master Model

The master behavior in the network is modeled according to Fig. B.3. This model presents the different states that the master transits through. Each state is associated with an action that resembles a PTP activity, send/receive a message. Transition from one state to another is controlled by a specified guard, for example a master will not send a *Delay_Resp* message until it receives a *Delay_Req* message. While transiting between states, the master keeps track of its clock represented by a timer. The master shares this value with the slave as defined by the states. As Fig. B.3 shows, the presented master model clearly complies with that of a PTP master clock.

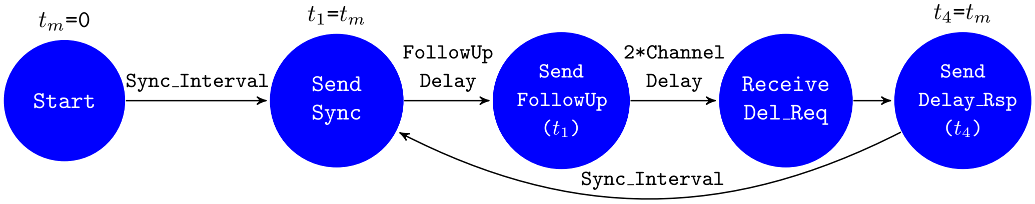

Figure 4.3: Model of the master clock in PTP

4.5.2 PTP Slave Model

The main characteristic of the slave clock is the drift it enjoys from the time reference. On the abstract model level, it is not possible to assign a distribution function for the clock drift. And as PRISM allows increments in integer values only, an approach to adjust the slave clock with a drift is needed. Thus, we made use of PRISM syntax that allows taking actions based on specified probabilities. Hence, we model the slave clock as a Bernoulli random variable. The slave clock advances by a value of 1 with probability P, and by 0 with probability *1-P*. This allows us to exhibit the difference between master and slave clocks, and introduce the required drift at the slave side.

The slave model in Fig. 4.4 represents the different states of the slave clock as indicated in PTP. Transitions between states are controlled by guards similar to those in the master model.

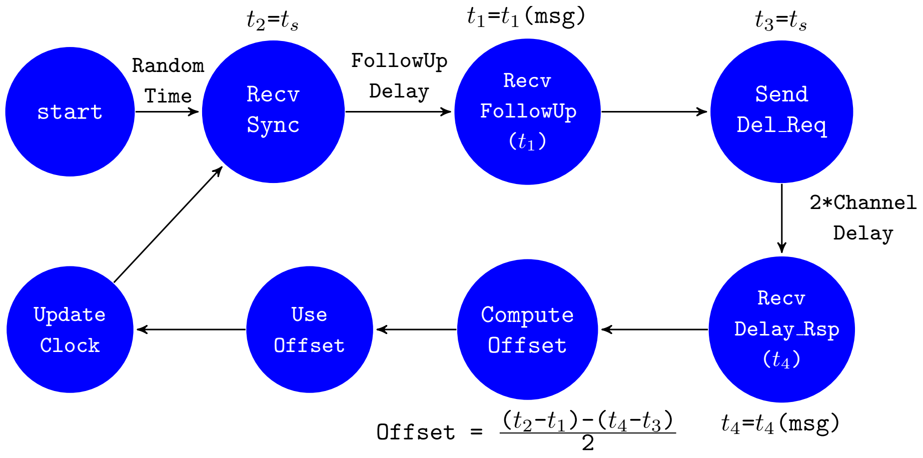

Figure 4.4: Model of the PTP slave clock

It is worth noting that the same approach can still be applied in the case of one-step clocks where the master does not issue any *FollowUp* messages to communicate timestamps to slave clock. In fact, this scenario would be easier to formally model since it includes less states, message exchanges, and eventually state synchronization activities.

4.6 Detection Approach

Our proposed approach consists of two main components: an introduction and definition of a new type of PTP clock in the network, the guard clock, and a modification of the PTP slave functionality to respond to alerts raised by the guard clock aiming at mitigating the effects of the delay attack.

The proposed model can be deployed over the station bus in an IEC 61850 substation. It introduces modifications to the functionality of a selected device. This device is chosen so that it enjoys the same time accuracy of that of the master. This can be achieved either by using two clocks of the same properties, or supplying the two devices with the same timing signal. The introduced modifications are described in the upcoming section.

4.6.1 Detection Model

PTP delay attack is characterized by being difficult to detect as it avoids tampering with the exchanged PTP messages, and by the inability of the slave devices to detect an ongoing attack. Indeed, PTP delay attack can not be detected at the slave side by relying on sudden changes in the computed offset values. PTP slaves depending on such a strategy will fail to defend against and mitigate the attack. To support this claim, we presented an analysis of the offset calculations performed under attack at slave clocks to show that notable changes are present only after the first instance of the attack. Offset calculations thereafter do not reflect any changes on the network during the attack as indicated in Remark 4.1.

The proposed detection approach through the guard model is based on the following remark.

Remark 4.2. PTP offset calculation between two clocks supplied by the same timing signal over a symmetric communication channel results in zero.

Proof Assume we have two clocks, C_1 and C_2, synchronized to the same timing signal (for example, through GPS), communicating over a symmetric channel. Let C_1 be the PTP master and C_2 the slave. Let D be the time taken by a message sent by C_1 to reach C_2. Since the communication channel is symmetric, D is also the time taken by a message sent by C_2 to reach C_1. Suppose that C_1 sends a *Sync* message at time $t_1 = t$. This message reaches C_2 at time t_2, where $t_2 = t_1 + D$ since the two clocks are supplied by the same time signal. C_2 sends the *Delay_Req* message at time $t_3 = t'$. C_1 receives the *Delay_Req* message at time $t_4 = t' + D$. C_1 communicates t_1 and t_4 using *FollowUp* and *Delay_Resp* respectively. C_2 now calculates the offset using equation (2.2).

The calculated offset is: $ClockOffset = \frac{(t_2 - t_1) - (t_4 - t_3)}{2}$

Substituting t_1, t_2, t_3, and t_4 by their respective values, we get:

$ClockOffset = \frac{((t+D)-t)-((t'+D)-t')}{2} = \frac{D-D}{2} = \frac{0}{2} = 0$

Thus the remark holds.

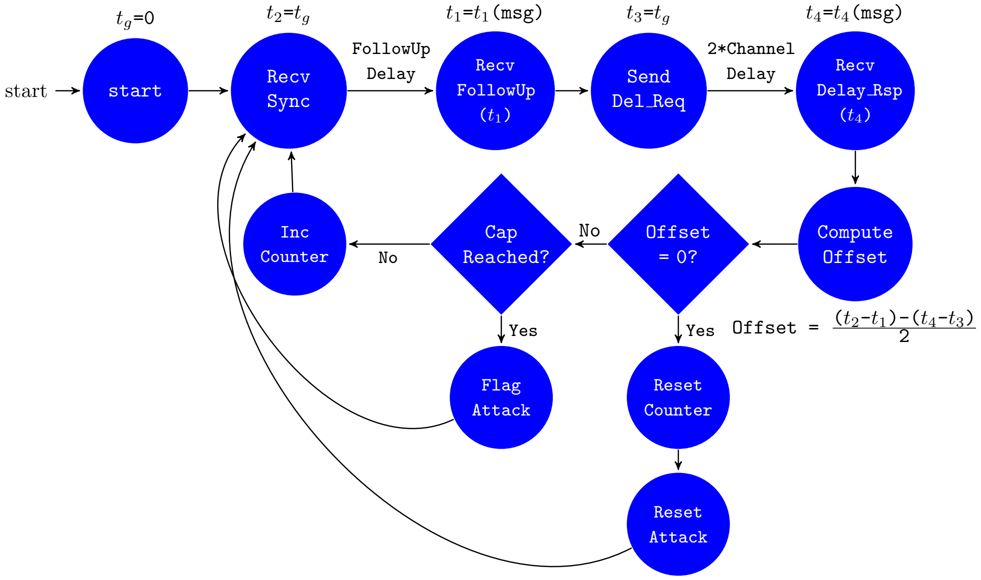

Figure 4.5: Model of the guard clock

The guard model shown in Fig. 4.5 is based on Remark 4.2. This model is designed as a detection mechanism for the delay attack in a smart grid substation environment. To improve the resiliency in a substation, multiple IEDs can be GPS enabled so that the failure of the master can be easily mitigated by a run of the BMC algorithm, see Fig. 4.1. However, during normal operations of PTP, these IEDs will play the role of a slave as BMC restricts the number of masters in a network to one.

To make a better use of the capabilities of these IEDs, we propose modifying the role of a GPS enabled IED per the guard model in Fig. 4.5. The guard transits between the states as indicated. However, being supplied with accurate timing, this IED does not need to synchronize its clock with that of the master. Instead, it transits to a verification state that validates the calculated offset as Remark 4.2 states. This transition allows to detect tampering with PTP services.

Thus, in the presence of an attacker delaying selected PTP messages, the offset is affected by the introduced delay δ and the calculated value differs from zero ($\mathit{offset} = \frac{\delta}{2}$, see equation (4.1)). The guard detects this violation as the state diagram shows. Later, it flags an attack when the violation repeatedly occurs for a specified number of times. This alert results in the activation of the mitigation mechanism at the slaves side until the attack flag is reset by the guard. Thus, the combination of the guard model and the modified PTP slave provide a detection and mitigation mechanism to improve the resiliency of PTP against delay attacks.

4.7 Mitigation Approach

To mitigate the effects of the delay attack, our approach consists of two parts. The first one is calculating an adjustment for the slave clock to annihilate the effects of the delay attack, and the other is to maintain synchronization at the slave side while under attack.

4.7.1 Slave clock adjustment

To determine the adjustment required to the slave clock, we need to analyze the delay effect at the slave side once the slave is notified by the guard of the undergoing attack.

We assume that k consecutive occurrences of the offset violation is specified in the guard model for the attack detection.

Referring to equation (4.1), we notice that after a single execution of the delay attack, the slave clock is behind that of the master by $\frac{\delta_1}{2}$. Thus, the time at the slave side can be represented as $t_{slave} = t_{master} - \frac{\delta_1}{2}$.

When the second synchronization cycle starts, time stamps collected are as follows(see Fig. 2.3):

$t_1 = t_{master_1}$

$t_2 = t_{slave} + \delta_2 + \epsilon_1 = t_{master_1} - \frac{\delta_1}{2} + \theta + \delta_2 + \epsilon_1$

where δ_2 is the delay introduced by the attacker, θ is the offset between the master and slave time in the absence of attack, and ϵ_1 is the master-slave path delay.

Similar to proof of Remark 4.1, $t_3 = t_2$ and $t_4 = t_{master_1} + \delta_2 + 2\epsilon_1$

Computing the offset using equations (2.1 & 2.2), we get:

$$\begin{aligned} \mathit{Offset} &= \frac{t_2 - t_1}{2} - \frac{t_4 - t_3}{2} = \frac{t_{master_1} - \frac{\delta_1}{2} + \theta + \delta_2 + \epsilon_1 - t_{master_1}}{2} \\ &- \frac{t_{master_1} + \delta_2 + 2\epsilon_1 - (t_{master_1} - \frac{\delta_1}{2} + \theta + \delta_2 + \epsilon_1)}{2} \\ &= \frac{\delta_2 - \frac{\delta_1}{2} + \theta + \epsilon_1}{2} - \frac{\epsilon_1 + \frac{\delta_1}{2} - \theta}{2} = \frac{2\theta + \delta_2 - \delta_1}{2} \\ &= \theta + \frac{\delta_2 - \delta_1}{2} \end{aligned}$$

Using this offset in equation (2.3) to update the slave clock, we get:

$$t_{slave} = t_{master} - \theta - \frac{\delta_2}{2}$$

Thus, the slave clock is behind that of the master by half of the current delay δ_2. In a similar way, we can show that the slave clock will be behind that of the master by $\frac{\delta_k}{2}$ when the guard detects the delay attack where δ_k is the delay introduced by the attacker during the k^{th} synchronization cycle.

Therefore, the slave should adjust its clock by the value $\frac{\delta_k}{2}$ once it is notified of the ongoing delay attack to retain synchronization with the master clock.

To determine an approximate value of $\frac{\delta_k}{2}$, we propose modifying the slave clock to enable it of computing a moving average of the computed offsets and store it for later use in the case of attack. The average calculation is presented in equation (4.6) where *Offset(t)* and *Avg(t)* are the calculated average and offset at time t respectively, *Avg(t-1)* is the cumulative average at time *(t-1)* and α is a parameter tuned to control the weight associated to *Avg(t-1)* and *Offset(t)*. In the absence of an attack, α is set to $\frac{1}{2}$.

$$\mathit{Avg(t)} = \alpha \times \mathit{Offset(t)} + (1-\alpha) \times \mathit{Avg(t\text{-}1)} \tag{4.6}$$

As part of the proposed mitigation, the slave should store the last *(k+1)* computed offsets and averages where k is the guard specified parameter.

Assume that the slave computes *Avg(n)*, and the attack starts at *(n+1)*. The guard will not alert the slaves about the ongoing attack before k synchronization cycles takes place. Meanwhile, slave clocks compute and store the following average values:

$$\begin{aligned}
\mathit{Avg(n{+}1)} &= \frac{\mathit{Offset(n{+}1)} + \mathit{Avg(n)}}{2} \\
&= \frac{\theta(n+1) + \frac{\delta_1}{2}}{2} + \frac{\mathit{Avg(n)}}{2} \\
&\vdots
\end{aligned}$$

$$\begin{aligned}
\mathit{Avg(n{+}k)} &= \frac{\mathit{Offset(n{+}k)} + \mathit{Avg(n{+}k\text{-}1)}}{2} \\
&= \frac{\theta(n+k) + \frac{\delta_k - \delta_{k-1}}{2}}{2} + \frac{\mathit{Avg(n{+}k\text{-}1)}}{2}
\end{aligned}$$

Hence when the slave clock is informed of the existence of an attack, the slave had computed and stored *Offset(n)* $\cdots$ *Offset(n+k)*, and *Avg(n)* $\cdots$ *Avg(n+k)* where the last k offsets and averages are affected by the delay attack.

The slave will use the stored offsets to compute the required δ_k using the following relations:

$$\begin{aligned}
\theta_{n+1} + \frac{\delta_1}{2} &= \textit{Offset(n+1)} \\
&\Rightarrow \delta_1 = 2(\textit{Offset(n+1)} - \theta_{n+1}) \\
\theta_{n+2} + \frac{\delta_2 - \delta_1}{2} &= \textit{Offset(n+2)} \\
&\Rightarrow \delta_2 = \delta_1 + 2(\textit{Offset(n+2)} - \theta_{n+1}) \\
&\vdots \\
\theta_{n+k} + \frac{\delta_k - \delta_{k-1}}{2} &= \textit{Offset(n+k)} \\
&\Rightarrow \delta_k = \delta_{k-1} + 2(\textit{Offset(n+k)} - \theta_{n+k})
\end{aligned} \tag{4.7}$$

As the clock drift can be assumed to enjoy a linear change over small time ranges [123], (humidity, temperature and other factors affecting the clock oscillator vary gradually over a long period of time and rarely show steep variations), the slave will replace each of the θ_i values in the system of equations (4.7) by *Avg(n)* to compute δ_i where $i \in$ *(1,k)*. This computation will result in:

$$\delta_k = 2 * \sum_{i=1}^{k} \textit{Offset(n+i)} - 2k\textit{Avg(n)} \tag{4.8}$$

with an error of the order *2ke* where $e = |\textit{Avg(n)} - \theta_i|$, $i \in$ *(1, k)*.

4.7.2 Maintaining slave clock synchronization

To maintain clock synchronization at the slave side, the slave clock needs to periodically adjust its value with a carefully computed offset as in PTP. However since the computed offsets are invalid and affected by the delay attack as we showed in our previous analysis, we propose to use the stored *Avg(n)* value to update the slave clock value using equation (2.3).

Thus after being notified by the guard of the ongoing attack, slave clocks compute the δ_k value and use it for a single time to adjust their clocks. Then, the stored *Avg(n)* value is used for periodic updates of the clock until the guard signals the end of the delay attack.

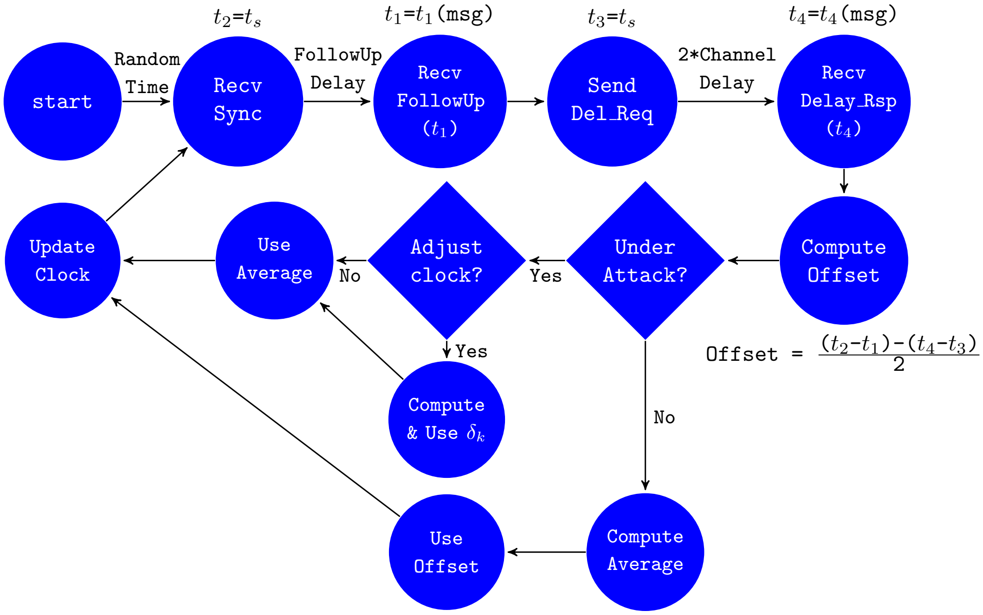

Figure 4.6: Mitigation Model at the PTP slave clock.

The presented mitigation model is proposed for integration at the PTP slave side as those slaves are the target of the delay attack. The resulting slave model after the integration of the mitigation model is presented in Fig. B.2.

4.8 Experimental Results

The aim of this section is to evaluate the performance of the developed PTP model, the effect of the delay attack, and the effectiveness of the proposed detection and mitigation mechanisms. This evaluation is twofold, probabilistic model checking of the presented model and numerical simulation.

Probabilistic model checking is a formal verification technique for the modeling and analysis of stochastic systems [69]. It is based on the construction and analysis of a probabilistic model, typically a Markov chain or process. The model is constructed in an exhaustive fashion, based on systematic exploration of all possible states that can occur. Once the model is constructed, it can be used to analyze a wide range of quantitative and qualitative properties of the original system. Probabilistic model checking has proved to be useful for studying a wide range of quantitative properties of models taken from different application domains [69].

To perform formal model checking, a tool is needed. Our tool of choice is PRISM [53]. PRISM is an open-source probabilistic model checker developed at the University of Birmingham and now at the University of Oxford. PRISM allows the description of models using a high level system description language. It provides support to build several types of models: Markov decision processes, probabilistic automata, probabilistic timed automata, and discrete and continuous Markov chains plus extensions of these models with costs and rewards. PRISM was chosen for the simple, textual modeling language it provides along with the ability to choose preferred engine for the processing and analysis of the model.

PRISM is used in the literature for the quantitative and qualitative verification of security properties for a variety of systems. We point out a formal analysis technique for probabilistic security properties of peer-to-peer communication systems based on random message routing among members as presented in [103], quantitative validation of a security protocol and probabilistic analysis of the success of attacks in [11], quantitative and qualitative evaluation of embedded systems with a proposed resilience strategy in [46], the analysis and formal demonstration of a free-space Quantum Key Distribution (QKD) system in [47], the quantitative analysis of the Certified E-mail

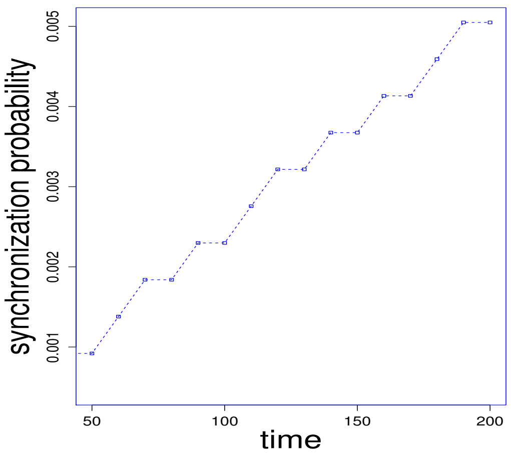

Figure 4.7: PTP model execution

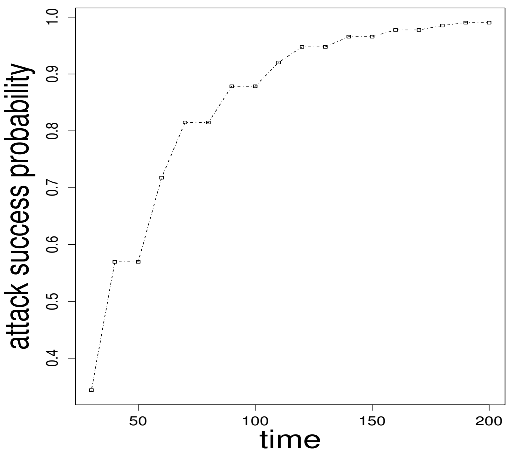

Figure 4.8: Delay attack effect

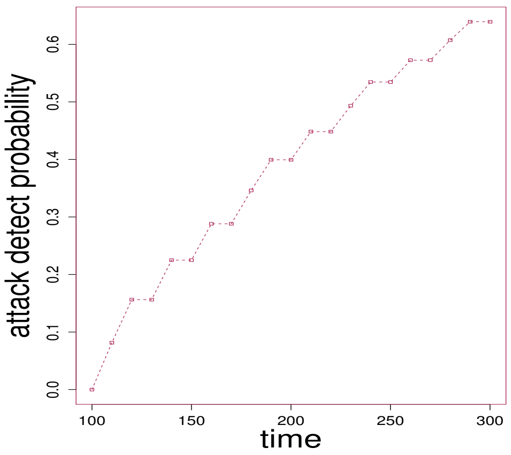

Figure 4.9: Delay attack detection

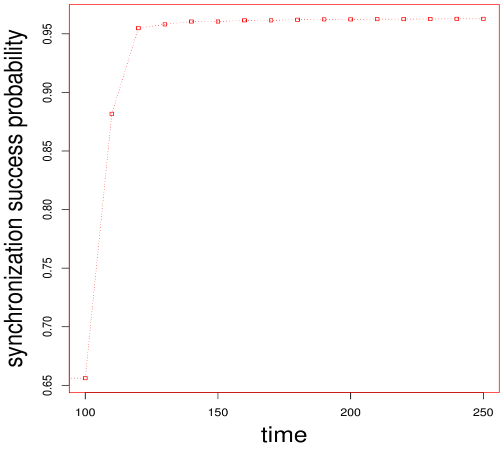

Figure 4.10: Guard model execution

Message Delivery (CEMD) protocol in [21], systematically quantifying DoS security threats using probabilistic model checking in [20], and the analysis of the probabilistic non-repudiation protocol quantitatively through probabilistic model checking and PCTL properties in [100].

Formal probabilistic model checking is made possible through property specification. PRISM property specification language is based on temporal logic. PRISM verifies properties specified using probabilistic computation tree logic (PCTL). This allows to specify probabilistic properties with respect to the likelihood of the occurrence of a desired state.

4.8.1 PCTL Properties

Since the model has no sense of real time, we set it up to reflect the elapse of time during execution. Thus, the steps are abstracted as time units rather than seconds. The values chosen reflect the execution of the synchronization mechanism, and the propagation of the messages in the network.

To setup the experiments, and as an input to the model on PRISM, we specify the probability controlling the clock tick at the slave side by P=0.9. This allows for a relative accuracy at the slave clock compared to that of the master. Other input includes the synchronization interval which is chosen as 10 time units, the delay between *Sync* and *Followup* messages specified as 1 time unit, and the time it takes a message to traverse the communication channel set to 2 time units. Moreover, the offset violating threshold set as 2 time units, and the acceptable number of such consecutive violations set as 3 are input for the guard model. The tests were run on a variable time interval, starting with 30 time units and 10 units increment. We will use t_m to represent the master clock and t_s to represent that of the slave.

To reflect the time skew between the clocks in the network, and the maximum probability of this skew bypassing a specified threshold after a synchronization event; the following PCTL property is specified and the output results are collected:

$$P_{max} =? \, [F((t_m - t_s > 2)|(t_s - t_m > 2)) \ \& \ \textit{synched=true}] \tag{4.9}$$

In this property, the temporal operator 'F' implies that the checked property eventually becomes true at some point along the path, and '|' resembles the logical OR as per PCTL syntax. The threshold is set to 2 time units, and the results collected are depicted in Fig. 4.7. As seen in the figure, the model preserves the required threshold with a confidence probability of the order 10^{-3}. This validates the developed PTP model success in synchronizing the slave's time to that of the master. In addition to that, this property validates that the PTP functionality is not affected by the introduced modifications. However, to assess any possible impacts of the proposed modifications on slave functionality in a substation environment, we need to modify PTP implementations on such devices before running our experiments. Such a modification is not possible since PTP implementations are setup by manufacturing utilities and usually they do not provide the capacity to alter the software component of those devices apart from configuration settings.

To demonstrate the effect of the attack, we introduced a delay of 5 time units to the master-to-slave path. The used PCTL property evaluates the effect of this delay on the slave time by determining the minimum probability that a 2 units time difference exists between the master and slave clocks after a synchronization event. The property used is:

$$P_{min} =? \, [F((t_m - t_s > 2)|(t_s - t_m > 2)) \; \& \; \mathit{synched}{=}\mathit{true}] \tag{4.10}$$

The collected results as plotted in Fig. 4.8 indicate that the attack succeeds in introducing a drift at the slave clock. The success rate increases with time elapse. After 200 time units, the success probability reaches 1.0. This proves that the delay attack succeeds in targeting time at the substation devices thus affecting the critical operations of the smart grid.

The effectiveness of the guard model presented is evaluated through another PCTL property. The property evaluates the minimum probability that an attack takes place and this attack is flagged by the model. The property used is:

$$P_{min} =? \, [F((t_m - t_s > 2)|(t_s - t_m > 2)) \; \& \mathit{synched}{=}\mathit{true} \; \& \; \mathit{attack}{=}\mathit{true}] \tag{4.11}$$

The results collected are plotted in Fig. 4.9 and show the minimum probability that the attack flag is set when an attack effect is noticeable in the network. The collected results show that the guard captures the on-going attack with a minimum success rate that increases as time passes. The scenario considered is that the attack takes place after the elapse of 85 time units on the master to slave path.

To quantify the mitigation approach, we hit the system with an attack after the elapse of 70 time units. This allows cumulative average calculation at the slave side under normal circumstances. We collected results to quantify the time difference between the master clock and synchronized slave clock when system is under attack. The property used is:

$$P_{min} =? \, [F((t_m - t_s \leq 2) \& (t_s - t_m \leq 2)) \ \& t_m > 70 \ \& \ \mathit{synched}{=}\mathit{true}] \tag{4.12}$$

This property computes the minimum probability that the time difference stays within 2 time units. As the collected results in Fig. 4.10 show, using the cumulative average after detecting the attack results in an acceptable time difference between the master and slave clocks. The minimum probability that this difference does not exceed 2 time units exceeds 0.95 for time intervals longer than 120 time units. Thus, we can say that time synchronization is maintained in the network and the mitigation strategy proved its effectiveness in preserving the system functionality.

4.8.2 Experimental Results

To evaluate and verify the validity of our proposed modifications to PTP, we setup a network to synchronize machines using PTP. For that purpose, we use the open source Precision Time Protocol daemon (PTPd) project [8] as a base for our implementation on four Ubuntu 14 machines. PTPd is a full IEEE 1588-2008 (PTPv2) protocol implementation with software-only timestamping. For our experiments, out of the four Ubuntu 14 machine, one machine plays the role of a master, one that of a guard, and two are PTP slave clocks. We equipped one of the slave machines with the detection and mitigation approach. To supply the master and guard with the same timing signal, both of them are equipped with two network cards; one card is for time synchronization through the internet and the other for PTP communication over the established network. PTP is used to synchronize the machines over the network in the absence of an attack. After the elapse of 20 minutes, the attack is performed by delaying the synchronization messages sent from the master for 6.5 milliseconds. The guard detection threshold is set to 3 milliseconds, and the threshold violation cap is set to 3. Results are collected over a time window of 2 hours.

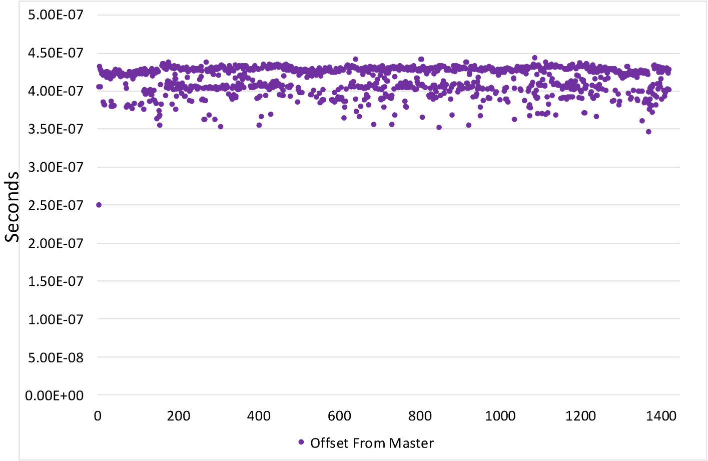

Figure 4.11: Offset calculation at guard machine

The first test scenario is used to validate Remark 4.2 where PTP is run over the network in the absence of an attack for one hour. The results of the offset calculation on synchronization events at the guard device are plot in Fig. 4.11. As can be seen from Fig. 4.11, the offset calculated at the guard side varies between 0.35 and 0.4 μseconds. This validates our remark as the calculated offset is negligible and thus the guard is capable of detecting an attack that violates the offset calculation.

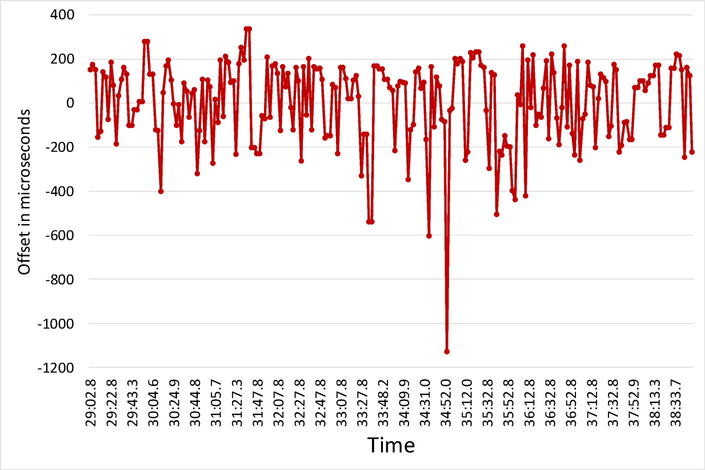

Figure 4.12: Offset calculation at slave machine

In the second scenario, we plot the offset variation at the slave not enabled with the detection and mitigation model in Fig. 4.12. The attack occurs at minute 34 second 52. As can be seen from the results, upon the initiation of the attack, the calculated offset varies at the slave by a value distinct from the previous ones. The calculated offset in absolute value, 1200 μseconds, is more than twice those calculated before or after the occurrence of the attack. However, this variation occurs only once and the calculated offset values thereafter conform with the previously calculated ones. This verifies by experimentation Remark 4.1.

In our third scenario, we aim to show the time error at the two slaves compared to the accurate reference time. The collected results showing the error at the two slaves are plot in Fig. 4.13. As Fig. 4.13 show, the slave enabled with the mitigation model maintains its synchronization with an error margin less than 0.1 milliseconds over around 90 minutes. The slave not enabled with the mitigation model shows a synchronization error more than 0.2 milliseconds. These results validate the usefulness of the proposed detection and mitigation model.

Therefore, the collected results through PTP execution on a physical systems

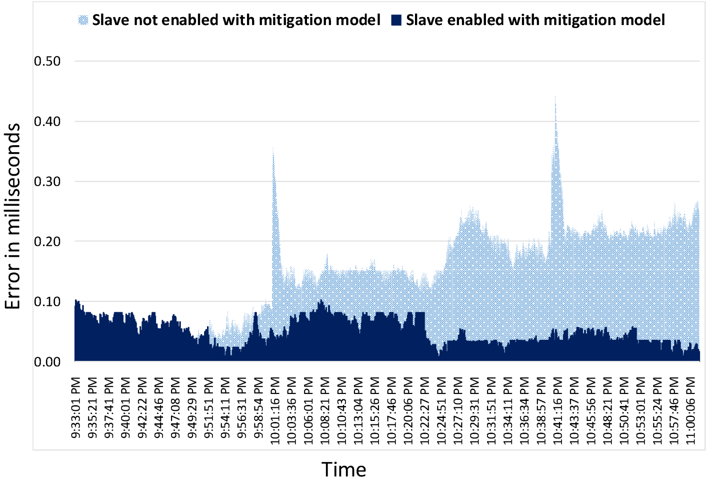

Figure 4.13: Time difference at the two machines

confirm the results collected using PRISM. Thus, the proposed approach is capable of detecting an ongoing delay attack and maintain the clock synchronization at the model enabled machines within acceptable accuracy limits.

On the other hand, it is worth mentioning that the evaluation of the impact on different slaves functionality in a substation environment is needed. As we have altered the synchronization logic followed by slave clocks, we have demonstrated that the synchronization status of slaves is not affected. However, those slave clocks are located in devices assigned several functionalities pertaining to their nature and availability in the substation such as GOOSE publisher or subscriber, etc. Thus, we need to analyze the effect of the proposed logic on those functionalities. Nevertheless, we consider this task out of the scope of our work due to the fact that PTP implementation on such devices is supplied by utilities who do not grant the capabilities to alter software components of those devices. Therefore, a modification of the PTP logic on a real device should be done at the manufacturing utility and is beyond our reach and capabilities.

4.9 Conclusion

In this chapter, we proved the vulnerability of PTP to the delay attack. We provided an analysis and quantification of the attack impact on slave clocks in a PTP network. We have shown that using variations of offset at slave clocks does not enable the detection of an existing attack, and thus proposed a detection mechanism based on the availability of a backup master clock. Our detection mechanism introduces a new functionality to the PTP network that uses timestamps collected from the master through synchronization messages to detect the presence of an ongoing delay attack. We formally modeled the detection mechanism and proved its usefulness through formal verification using PRISM model checker, and experimental evaluation using PTP implementation on a physical system.

On the other hand, to survive the impact of the attack and restore the synchronization status of connected slaves, we presented a suitable mitigation mechanism. Our mitigation mechanism is deployed at slave clocks, and makes use of historical synchronization records to adjust the slave clock upon attack detection, and allow them to maintain their synchronization using the same history. The presented mitigation scheme was verified through PCTL properties, and proved efficient through experimental evaluation.

Chapter 5

An Extension to the Precision Time Protocol to Enable the Detection of Cyber Attacks

In the previous chapters, we addressed security threats related to fake timestamps and delay attack targeting PTP time synchronization. In this chapter, we capitalize on the theory and outcome of Chapter 4 to contribute a more complete solution that addresses PTP cyber security. We propose to close the PTP loop through an extension that introduces new functionality and messages. This extension covers the PTP attack surface and enables the detection of attacks on PTP time synchronization. We formally model and verify the proposed extension using UPPAAL model checker. In addition, we validate the proposed extension using Omnet++ simulation.

5.1 Introduction

In the next-generation smart grid infrastructure, accurate timing signals will be broadly required from generation plants to distribution substations to individual smart grid components [96]. Time synchronization is a key enabling service that coordinates the actions of devices dispersed across these domains. A consistent notion of time shared across the smart grid will make it easier to monitor, control, ensure the availability of collaborative services provided by those devices, and fulfill the smart grid vision.

The value of time synchronization is best understood by recognizing that the power grid is a complex, interconnected and interdependent network. Thus, events in one part of the grid affect operations elsewhere, and extend beyond the grid to other systems that are reliant on stable power, much like what was observed in the 2003 Northeast Blackout [74]. Thus, utilities are seeking precise timing solutions to improve the resiliency and the monitoring of their transmission and distribution networks [52, 99, 64]. Such solutions should avoid additional costs in terms of infrastructure, and enable the convergence of the utilities timing and data communication networks [72]. Power utilities have recognized that PTP, standardized as IEEE 1588 [4], network-based precise and accurate time distribution protocol can deliver sub-microsecond synchronization accuracy and meet the needs of power applications, and directed their efforts to propose optimal PTP system design for wide area network environment using appropriate clock and network parameter settings [64].

IEEE 1588 outpaced previous timing solutions in terms of accuracy, scalability, and cost. It meets the emerging timing accuracy needs of the Station Bus and Process Bus in IEC 61850-based substations [106]. With PTP, precise and sub-microsecond time is distributed over the same Ethernet network used for data communications, and the PTP profile used in power system applications is now standardized [6]. PTP with its different clock synchronization mechanisms can adapt to the needs and specifications of various systems. Moreover, it promotes the use of transparent clocks (TCs) to cancel the effect of randomness in the network on the synchronization accuracy.

On the other hand, security concerns were associated with PTP functionality [119, 50, 120, 125] and more recently PTP security gained additional interest from the research community [90, 87, 60, 88]. Those concerns have escalated with the candidacy of PTP to be used in the smart grid especially with the demonstrated impact of time synchronization attack [128]. Requirements for secure clock synchronization are presented by [90] in the form of necessary and sufficient conditions applicable to one-way and two-way time transfer. Under those conditions, PTP is found to be non-secure, and specialized conditions for secure PTP are presented. On the other hand, Itkin et al. [60] proposed a revised security extension for PTP to address pitfalls in the current Annex-K, PTP security extension [4]. In our previous work [88], we leveraged the synchronization requirements in a substation as imposed by IEC 61850 to propose a detection and mitigation model for PTP delay attacks. We established theory on the capability of a redundant accurate clock to detect such attacks in PTP network. The proposed approach was successful in detecting and mitigating delay attacks in a substation with a flat network architecture. However, the detection approach in [88] may fall short under different network architecture. Moreover, the previous approach does not cover the entire PTP attack surface. Thus, there is a need for a more generic mechanism that covers the entire PTP attack surface, and unveil all security concerns associated with PTP components.

This chapter addresses cyber-attacks targeting PTP. We present an approach to harden PTP security through an effective cyber attacks detection mechanism. Our analysis is performed on PTP as used in an IEC 61850 substation. However, it is generic and can be adopted by other systems using PTP for time synchronization. In our approach, we propose an extension to PTP to enable the collection of time stamped messages from slave clocks to assess the clock synchronization status. Through the collected messages, PTP is therefore capable of detecting cyber attacks targeting the network and the connected devices. As far as we know, we are the first to propose closing the PTP loop from a security point of view. Since the impact of attacks is only noticeable at end devices regardless of preventive measures, feedback collection from PTP slaves is a major step towards a security-aware PTP implementation. Thus, compared to other approaches in the literature, we have the advantage of monitoring the status of time synchronization at end devices, and thus detecting all cyber attacks that target slave clocks regardless of their nature. We formally verify the proposed extension using UPPAAL model checker [70], and establish its correctness through Computational Tree Logic (CTL) properties. Moreover, its validity and overhead are evaluated through Omnet++ simulation.

5.1.1 Novel Contributions

This chapter presents a proposed extension to PTP that allows the collection of timestamped messages from slave clocks to assess the security posture of the PTP network. The proposed extension leverages the ability of slave clocks to send and receive timestamped messages, and the presence of alternate time sources to secure the PTP network. The main contributions of this paper can be summarized as:

1. We propose to extend the Precision Time Protocol to collect feedback from slave clocks on their synchronization status, and analyze this feedback to detect cyber attacks targeting different components of the PTP network. The extension allows to close the PTP loop, and increase the reliance on PTP slaves. We change the slave role from solely receiving timestamped messages to actively sending such messages to a reference entity that monitors the health of time synchronization in the network.

2. We formally model and verify the proposed extension using timed automata

through UPPAAL model checker. Through CTL we validate liveness and safety properties of the proposed extension and demonstrate its correctness and ability to detect attacks against PTP. Moreover, we evaluate the proposed detection logic using Omnet++ simulation which manifests the proposed extension capabilities.

The remainder of this chapter is structured as follows. Section 5.2 presents the system under study with an emphasis on the PTP attack surface. Section 5.3 introduces our attacker through a threat model. The proposed attack detection model is detailed in Section 5.4. A demonstration of attacks detection using the presented approach is provided in Section 5.5. Section 5.6 presents an evaluation of the presented approach through formal verification and numerical simulation. Concluding remarks are provided in Section 5.7.

5.2 System Model

The system under study is once again an IEC 61850 substation. As recommended by IEC 61850-90-4 [106], PTP is used for time synchronization at the substation level. PTP grandmaster and backup clocks are located at the station bus, and communicate PTP event messages to the connected devices such as IEDs, protective relays, sensors, etc. A specified by by IEC 61850-90-4 [106], we consider the substation as a single domain, and operate at layer-2 using multicast only. Moreover, since PTP shares the station bus with other substation traffic (GOOSE, SV, SNMP, etc.) [44], IEC 61850-90-4 demands the usage of transparent clocks to reduce the impact of random queuing delays on synchronization. Indeed, transparent clocks (TC) measure the residence time of a PTP event message at a TC, and supplies this information to recipients of the message in transit. TC enables the reduction of randomness in the system in the presence of background traffic, and the accurate estimation of queuing delays experienced by PTP synchronization messages at intermediate switches in the network. Through the introduction of TCs, PTP slaves are capable of performing an accurate computation of master to slave path delays and achieve a better synchronization accuracy. The specifications provided by IEC 61850 results in the setup depicted in Fig. 2.4 and detailed in Section 2.3.

Next, we will present the PTP attack surface that we are considering for this chapter.

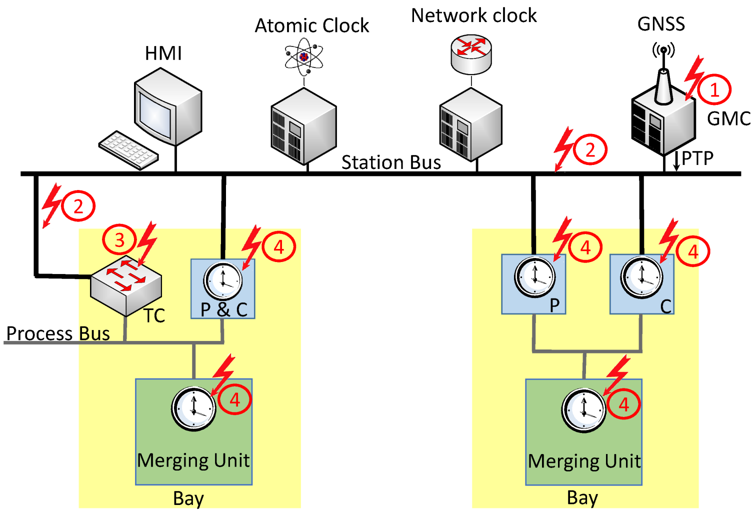

Figure 5.1: PTP attack surface for an IEC 61850 substation

5.2.1 PTP Attack Surface

Availability of accurate timing across the grid is fundamental for several applications such as voltage stability and fault localization [87], thus attacks on time synchronization will have an impact that goes beyond the targeted devices [128]. To attack PTP time synchronization mechanism, an attacker can target any of its main components as shown in Fig. 5.1:

5.2.1.1 Grand Master Clock (GMC)

Since PTP arranges the network in a hierarchy with the GMC as its only root, targeting the GMC will present a bad time reference for synchronization and impact all the connected devices. The attacker can carry this attack either by targeting the time source used by the master (as in GPS spoofing), through impersonating the master (as in Byzantine master), or through a manipulation of the timestamp generation mechanism at the master.

5.2.1.2 Communication Network

Time synchronization under PTP relies on an accurate measurement of the path delay between the master clock and the synchronized clocks. This dependency can be exploited, and is formulated in the literature in the form of the delay attack [120]. This attack introduces an error at slave clocks, which either lag behind or go ahead that of the master by a value specified by the attacker. Moreover, this attack affects all the slaves that receive the master messages over the compromised link.

5.2.1.3 Transparent Clocks (TCs)

Through targeting a TC, an attacker leverages the functionality of the TC to perform the attack, and avoid integrity-based detection mechanisms. The TC is allowed by PTP to reconstruct PTP event messages after updating the *correctionField*, among other modifications, to reflect the PTP event message residence time. This capacity of the TC will be targeted by the attacker to report faulty values for the residence time. The compromised TC reports a malformed value $\rho_i' = \rho_i \pm \mu$ instead of the benign residence time value ρ_i as in Eq. (2.2). Assume a compromised TC$_{i'}$ augments the measured residence time with a positive quantity μ, so that $\rho_i' > \rho_i$. The faulty values introduced in the *correctionField* are used by the slave clock as per Eq. (2.2):

$$\begin{aligned} \delta' &= t_{00} - t_0 - (\sum_{i,i\neq i'}(\rho_i + \lambda_i) + \rho'_{i'} + \lambda_{i'}); (\rho'_{i'} = \rho_{i'} + \mu) \\ &= t_{00} - t_0 - \sum_{i}(\rho_i + \lambda_i) - \mu = \delta - \mu \end{aligned} \tag{5.1}$$

This offset is used to adjust the slave clock per Eq. (2.3):

$$\begin{aligned} eTime_{slave}^{new} &= Time_{slave} - \delta' = Time_{slave} - \delta + \mu \\ &= Time_{slave}^{new} + \mu \end{aligned} \tag{5.2}$$

where $eTime_{slave}^{new}$ is the erroneous new slave time.
Thus, the slave clock will be ahead of that of the master by a time error equal to the introduced value μ. This effect is induced at all slave clocks whose communication with the master is manipulated by the compromised TC.

5.2.1.4 Slave Clocks

Attacks on a slave clock affect only the targeted slave in contrast to the other attacks that affect a subset of the connected slaves and all of them when the GMC is attacked. However, such an attack may impact the applications that use timestamped data from the targeted slave.

It is worth noting that, intelligent electronic devices (IEDs) targeted by cyber attacks in a PTP network are incapable of detecting an ongoing attack neither realize its impact as they lack the knowledge of accurate timing, and trust PTP messages delivered over the network. Thus, there is a need for an additional entity or channel capable of monitoring PTP operations and detecting cyber attacks targeting the entire PTP attack surface.

5.3 Threat Model

To carry out attacks on time synchronization, we assume that the attacker is either an external entity or an insider with malicious intents. He has the expertise to perform long-term reconnaissance operations required to learn the environment and execute a highly synchronized, multistage, multisite attack [71]. The attacker is aware that the target substation uses PTP-based time synchronization, and is familiar with the topology of the substation and the nature of the synchronized devices.

Moreover, we adapt the in-band adversary from [60] where the attacker has full control over the communication channel to delay and change fields of passing messages. Furthermore, the attacker is interested in targeting the functionality of all the devices in the network rather than a particular one. He will manipulate the clock of the connected devices by targeting the master, the network or the available transparent clock(s). If the attacker wishes to execute his attack through the introduction of additional hardware or software to the system, he has the expertise necessary to choose suitable attack locations to perform his attack, and avoid the available surveillance. Finally, our attacker aims at conducting a stealthy attack while remaining undetected. Thus, he has interest in conducting the delay attack, and controlling synchronization at slave devices rather than performing other more impactful attacks which might flag detection alerts at the substation.

5.4 Detection Model

Our proposed detection model is based on PTP functionality and the available system setup. It targets time synchronization in an IEC 61850 substation, however it is effective for similar systems. Taking into consideration that the PTP model is that of a master-slave, PTP network lacks information about the synchronization status of slave clocks. Hence, in the presence of cyber attacks, the impact on slave clocks cannot be perceived by other clocks in the network. Thus, it is essential to collect timing information from slave clocks to monitor their synchronization status. In this section, we present the details of our proposed cyber attack detection model.

5.4.1 Basic Blocks

The main components of our model make use of the outcome of PTP clock synchronization, the available system resources, and the collection of timing information from synchronized slave clocks.

5.4.1.1 Every slave is a master

The main goal of PTP is to accurately synchronize slave clocks to that of the GMC. Indeed, PTP successfully synchronizes clocks and achieves accuracy in the order of sub-microsecond at slave clocks [4]. Hence, if we look at the state diagram of a slave clock shown in Fig. 5.2, we see that the slave clock alternates between two states (Accurate as Master, and less accurate than Master). A slave clock is as accurate as that of the GMC when it is adjusted according to Eq. (2.3), and moves to be less accurate than the GMC when ticking. When the slave processes the *Sync* message and for a short time instance, the clock timing resembles that of the master with a tolerable error of the order of microseconds. A timestamp collected from the slave clock at this time, and handled properly across the network, is enough to judge the slave synchronization status. Thus, a PTP slave will issue a timestamped message to report the synchronization status of its clock. The path delay associated with this message will be calculated in a similar fashion to that of a *Sync* message issued by the GMC as shown in Fig. 2.4. Through the collection of this message, and a calculation of a respective offset using Eq. (2.5), an accurately synchronized clock can govern the synchronization status of the slave that issued the timestamped message.

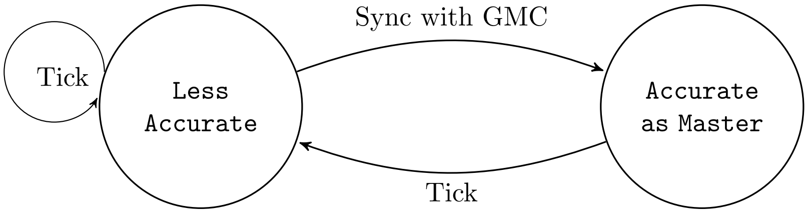

Figure 5.2: The state diagram of a slave clock

5.4.1.2 Redundant time sources

PTP specifications restrict the active GMC role to a single clock in the network. However, for better resiliency against master failures and according to IEC 61850 specifications, a substation must be equipped with multiple master-capable clocks as can be seen in Fig. 5.1. Our detection model leverages the existence of backup accurate time source, as a network time reference (NTR), to check the synchronization at slave clocks. The NTR time is always as accurate as that of the GMC even when ticking in contrast to the slave state machine in Fig. 5.2. Thus, such a clock is independent of the master and does not have to synchronize through the exchanged PTP event messages. In [88], we show that the clock offset between two accurate time sources is negligible. Thus, the offset value between the NTR and the synchronized slave should be negligible. The outcome of this calculation reflects best the health of the synchronization system and is thus at the core of the detection model.

We assume that the NTR time source is different from that of the GMC, secure and resilient to cyber attacks such as atomic clocks. Note that, in systems where backup time sources are not available, NTR functionality can be assigned to the GMC.

Table 5.1: Report message fields

Bits								Octets	Offset
7	6	5	4	3	2	1	0		
PTP Header								34	0
originTimestamp								10	34

5.4.1.3 Introduction of new PTP event messages

Through the current PTP specifications, slave clocks generate timestamped messages just to enable the measurement of path delay. We leverage this capability at the slave side to push timestamped reports to the NTR. On the other hand, the NTR will need to poll timestamped messages from the connected devices to improve the detection mechanism. Thus, we will introduce the following event messages to the current PTP message exchange:

- *Report*: This is a periodic timestamped message sent by slaves to report on their clock synchronization status. This message is addressed to the NTR (or to the master in case there was no redundant time sources in the network). The *Report* message has a similar structure to that of the *Sync* message and its fields are indicated in Table 5.1. This message carries a timestamp used in the calculation of an offset, and its header is updated at transparent clocks in a similar fashion to that of a GMC's *Sync* message. The output of this calculation reflects the synchronization status of the reporting slave. It is worth noting that sending a *Report* message is a task that can be served periodically by selected slaves (see Section 5.4.2.1).

- *Report_Req*: This message is issued by the NTR and addressed to a specific slave in the network. Through this message, the NTR requests a report on the synchronization status of the addressed slave. This message does not carry a timestamp. However, the reference entity expects a response from the addressed slave, otherwise a warning of the existence of a network anomaly can be raised.

- *Report_Resp*: This message is used by the slaves to answer requests made by the NTR. The *Report_Resp* message is similar in structure to the *Report* message, and is handled in the same manner by the network. The timestamp, and header fields of this message, are used to calculate the clock offset separating the reporting slave clock from that of the NTR, and thus reporting on the synchronization status of the slave clock.

5.4.2 Approach Realization

The detection of cyber attacks on PTP synchronization services is made possible by using the above basic blocks, and reshaping the functions of clocks in the system. This redefinition of tasks and responsibilities, and their impact on the detection of cyber attacks is detailed next.

5.4.2.1 Network Clustering

If we consider cyber attacks on the PTP attack surface, we can see that the attack impact is noticeable at a subset of the network. This subset can be a single slave, or the entire network depending on the attack nature. Thus, for attacks not directed at specific slaves, it is sufficient to monitor representative slave clocks from the system to detect the impact of an attack once it takes place. To setup the monitoring phase, the PTP network is logically clustered and slave clock are selected to issue periodic *Report* messages upon synchronization. Clusters are formed of slaves served by the same TC, or same network segment. This process is carried during network setup, and does not introduce any additional burden on a PTP network.

5.4.2.2 Slave Model

To enable detection of cyber attacks against PTP, we introduce few modifications at the PTP slave as indicated in Fig. 5.3. Those modifications are induced by the newly introduced PTP event messages. As shown in Fig. 5.3, upon clock synchronization, a slave clock may issue a *Report* message to the network time reference if the slave clock is assigned this role. Moreover, it may answer a previously received *Report_Req* through a timestamped *Report_Resp*. Apart from those two events, other actions are part of PTP specifications for clock synchronization (See Appendix B for a detailed description of the model).

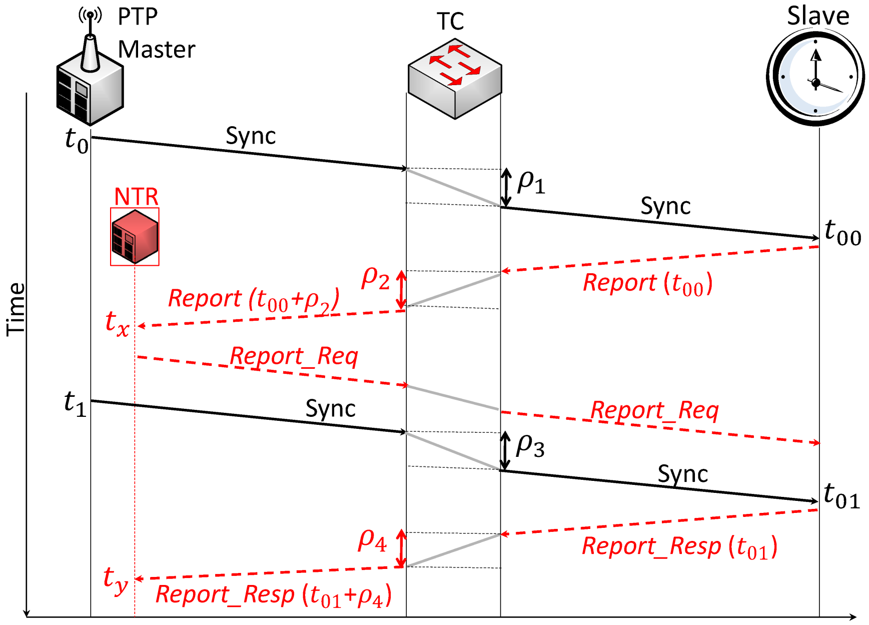

Figure 5.3: Time synchronization using the PTP extension.

5.4.2.3 Network Time Reference (NTR) Model

Through the proposed NTR model, we aim at introducing a new functionality to the PTP network as presented in Fig. 5.3. This functionality sets up the NTR communication with PTP slaves, in addition to processing PTP synchronization messages to detect errors in time synchronization resulting from cyber attacks. The NTR identifies problems with master time from *Sync* messages based on the computed offset which should be negligible as established in [88]. On the other hand, the NTR periodically receives *Report* messages from selected slaves in the network. Upon receiving a *Report* message, and based on our discussion in Section 5.4.1.1, the NTR will compute an offset from the reported *originTimeStamp*, *correctionField*, and the *Report* ingress timestamp. The result of this calculation is expected to adhere to a pre-specified threshold. A result bypassing this threshold is an indication of a malfunction in time synchronization, and this calls for further inspection which takes place in two steps:

1. *Verification of synchronization in same cluster*: The NTR issues a *Report_Req* to a slave in the same cluster as the faulty one, and waits for a *Report_Resp*. Upon the delivery of the *Report_Resp* from the addressed slave, the NTR computes an offset based on the collected information from the response. If the offset is less than a pre-defined threshold as expected, then there is a problem with the reporting slave. Thus, the NTR indicates that the problem is at the slave level. If the offset computed from the *Report_Resp* violates the threshold, the NTR moves to the second step of inspection.

2. *Verification of synchronization in different cluster*: In this step, the NTR issues a *Report_Req* to a slave in a cluster different from that of the faulty one, and waits for a *Report_Resp*. Once a *Report_Resp* is received, an associated offset is calculated and compared to the acceptable threshold. If the outcome is within threshold limits, the NTR indicates that a single cluster time is compromised. Otherwise, multiple clusters are experiencing timing problems.

It is worth noting that we prepared a model for the communication network connecting the GMC, NTR, and slaves (See Appendix B). The prepared model allows

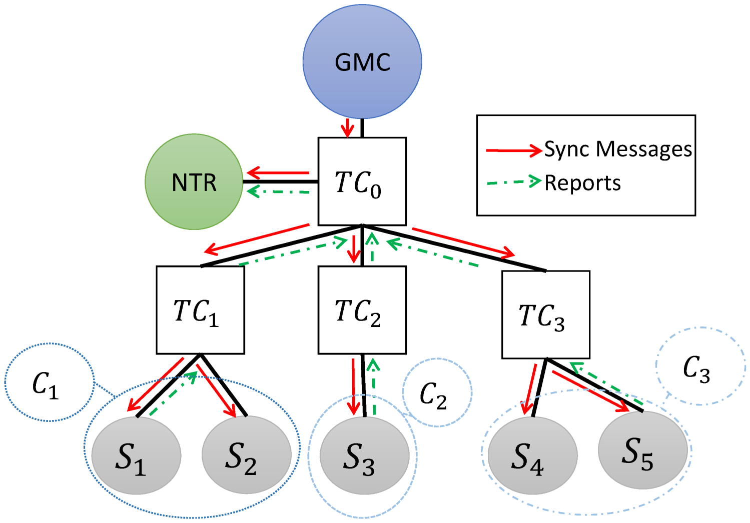

Figure 5.4: Network clustering in the presence of extended PTP.

the random selection of variable path delays to be experienced by messages, along with the capacity to specify a variable attack vector.

To enable the detection of changes in slave clocks as a result of cyber attacks, the threshold used by NTR has to be specified. This threshold corresponds to the system's tolerable synchronization error based on the application needs. Potential thresholds for use in substation are presented in Table 5.2 [87].

Table 5.2: NTR Synchronization Thresholds

Application	**Accuracy Requirement**
Phasor measurement	±31.8 μsec (50 Hz) ±26.5μsec (60 Hz)
Sampled Values	1 μsec
Traveling wave fault locator	1 μsec

On the other hand, to ease the application of the proposed detection mechanism, we assume that Annex K is implemented in the system under study. This assumption ensures that our detection approach is protected from message injection and replay through Annex K which is necessary for protection of PTP attack surface. We make this assumption keeping in mind that Annex K goals are only defined against non-PTP computers on the network, and has no effective way to differentiate between

PTP clocks [60]. Thus, it can not prevent the impersonation of the GMC or any other device in the network. However, it ensures the integrity and freshness of the communicated messages when implemented. Nevertheless, PTP can not be considered a secure protocol even in the presence of Annex K [85]. Indeed, Annex K is not sufficient to cover the attack surface presented in Section 5.2.1. The attacks on the GMC, the network, and TCs do not violate the authentication nor integrity constraints enforced through Annex K, and do not take the form of replay attacks that are prevented through message freshness. Thus, the PTP attack surface remains vulnerable to cyber attacks.

5.4.3 Approach Overhead

The presented extension introduces more load into the network in the form of new message exchanges. The overhead per second under normal conditions can be estimated as follows:

$$Overhead = Report\_Size \times \frac{Reports}{Sec} \times Number\_of\_Slaves \tag{5.3}$$

where *Report_Size* is the size of the *Report* message in bytes, and *Number_of_Slaves* corresponds to the number of slaves selected to issue *Report* messages to the NTR. This overhead can be best compared to the traffic introduced by the GMC in the form of *Sync* messages since the *Report* and *Sync* messages share the same structure and size. However, the frequency of sending reports is a factor of that of *Sync* messages. Thus, if n slaves are selected to issue reports every n synchronization events, the average introduced overhead in a second will be equivalent to that of n *Sync* messages. In the presence of an attack, this overhead is subject to an increase due to the report request and response mechanism used by the NTR to verify the synchronization state of slaves. This increase can be represented as:

$$Increase = 2 \times c \times (Report\_Req\_Size + Report\_Resp\_Size) \tag{5.4}$$

where c represents the number of clusters under attack.

It is worth noting that the extension does not demand new capabilities at slave devices. It is solely based on the ability of slave clocks to send and receive timestamped messages.

5.5 Attack Detection

Consider the network presented in Fig. 5.4. We will analyze different cyber attacks against the PTP attack surface, and demonstrate the ability of our proposed approach to detect those attacks. Through the detection setup, the network is logically divided into three clusters where slaves S_1, S_3, and S_5 are selected to periodically send reports to the NTR.

5.5.1 Attack on GMC

Suppose that the GMC time source is compromised, while other network components are trusted. Due to the attack, an error is introduced to the GMC leading to the generation of faulty timestamps. The GMC issues a *Sync* message carrying a timestamp t'_0 rather than t_0 where $t'_0 = t_0 \pm \mu$. The *Sync* reaches the NTR at t_1 with a path delay of d seconds. Thus, $t_1 = t_0 + d$. Using Eq. (2.2), NTR computes the offset associated with the *Sync* message as follows:

$$\delta = t_1 - t'_0 - d = t_0 + d - (t_0 \pm \mu) - d = \mp\mu \neq 0$$

The calculated offset does not comply with the expected outcome (see [88] for details). Thus, the NTR detects and signals the attack targeting the GMC.

5.5.2 Delay Attack on Communication Network

Suppose that the delay attack is targeting the (TC_0-TC_1) communication link by introducing an additional delay of μ seconds to this link, while other network components are trusted. Thus, using Eq. (2.4), the (TC_0-TC_1) link delay measurement at TC_0 and TC_1 results in a value $\lambda_1' = \lambda_1 + \mu/2$ where λ_1 is the true link delay. Now, the GMC issues a *Sync* message at t_0, this message reaches S_1 and S_2 at t_1 where $t_1 = t_{GMC} + \delta_{S_1}(\delta_{S_2})$ where t_{GMC} is the GMC time and can be expressed as $t_{GMC} = t_0 + \rho_0 + \lambda_0 + \rho_1 + (\lambda_1 + \mu) + \lambda_2$, where ρ_i is the queuing delay at TC_i, λ_0, λ_1, and λ_2 are the propagation delays on links GMC-TC_0, TC_0-TC_1, and TC_1-$S_1(S_2)$ respectively, and δ_{S_i} is the expected clock offset at S_i. However, the *Sync* message *correctionField* carries the value $(\rho_0 + \lambda_0 + \rho_1 + \lambda_1 + \mu/2)$. Using Eq. (2.2), S_1 and S_2 compute the following offsets:

$$\begin{aligned}
\delta_{S_{1(2)}} &= t_1 - t_0 - (\rho_0 + \lambda_0 + \rho_1 + \lambda_1 + \mu/2 + \lambda_2) \\
&= (t_0 + \rho_0 + \lambda_0 + \rho_1 + \lambda_1 + \mu + \lambda_2 + \delta_{S_1}(\delta_{S_2})) \\
&\quad - t_0 - (\rho_0 + \lambda_0 + \rho_1 + \lambda_1 + \mu/2 + \lambda_2) \\
&= \delta_{S_{1(2)}} + \mu/2
\end{aligned}$$

Using Eq. (2.3), S_1 and S_2 clocks are synchronized using the computed offset. Thus, a $\mu/2$ synchronization error is introduced to these clocks. Assume that S_1 issues a *Report* message at time t_s, $t_s = t_{GMC} - \mu/2$, the message will reach the NTR at time $t_{NTR} = t_{GMC} + d$, where d is the path delay from S_1 to NTR. Using Eq. (2.2) for a similar offset calculation to the one done above, the NTR computation results in an offset $\pm\mu/2$. Thus, the NTR will issue a *Report_Req* to S_2. The *Report_Resp* from S$_2$ will take the same path as the report sent by S$_1$, and will result in a similar error ($\pm\mu/2$). Based on this outcome, the NTR will issue a *Report_Req* to S$_3$. Upon clock synchronization, S$_3$ will send a *Report_Resp* to the NTR with a timestamp $t_s = t_{GMC} \pm \epsilon$ where ϵ is the synchronization error at S$_3$. This *Report_Resp* reaches the NTR at time t_{NTR} where $t_{NTR} = t_s + d$ where d is the path delay from S$_3$ to NTR. The *Report_Resp correctionField* is $(\rho_2 + \lambda_{S_3-TC_2} + \rho_0 + \lambda_{TC_2-TC_0} + \lambda_{TC_0-NTR})$. Using Eq. (2.2), NTR computes the offset corresponding to the received *Report_Resp* as follows:

$$\begin{aligned}\delta &= t_{NTR} - t_s - correctionField\\ &= t_s + d - t_s - correctionField\\ &= d - correctionField = 0\end{aligned}$$

Since *correctionField* corresponds to the propagation delay and queuing delay on the path from S$_3$ to NTR. Based on this outcome, the NTR issues an alert indicating that C$_1$ is under attack. Thus, the NTR detects the ongoing delay attack.

It is worth noting that a similar analysis can be performed for delay attack on other links. This analysis will result in a similar outcome to that of the delay attack on TC_0 - TC_1.

5.5.3 Attack on Transparent Clock

Suppose that TC_2 is compromised, while other network components are trusted. The attacker manipulates the queuing delay measurement mechanism at TC_2 ports to report a false residence time. We will distinguish between two cases:

1. *Single Port Manipulation*: Suppose that the attacker targets traffic originating from the GMC, and manipulates TC_2 port receiving the GMC traffic through TC_0. Due to this attack, TC_2 updates the *correctionField* in the *Sync* header with a value greater than the actual residence time ($\rho_2' = \rho_2 + \mu$), where ρ_2 is the queuing delay

at TC_2. Using Eq. (2.5) and (2.3), S$_3$ uses ρ_2' for offset calculation and clock synchronization respectively. Thus, S$_3$ clock is ahead of that of the GMC by μ. Once S$_3$ addresses the NTR with a *Report* message, NTR offset calculation will result in a value of $\pm\mu$ using a similar calculation to that in Section 5.5.2. Thus, the NTR requests a report from C_1 or C_3. Since neither C_1 nor C_3 are served by TC_2, the offset value calculated from the received *Report_Resp* will be negligible. Thus, the NTR marks C_2 as being attacked, and detects the ongoing TC attack.

2. *Multiple Port Manipulation*: In this attack variant, our attacker can handle queuing time measurement at all the TC ports, and is aware of the deployed attack detection mechanism. Through this manipulation, the attacker aims at hiding the error introduced at the slave clock, and avoid being detected by the NTR. To carry on with this attack, the attacker will report false queuing time ($\rho_2' = \rho_2 + \mu$) on the port receiving the GMC traffic as in the case of the single port manipulation. This will introduce an error of order μ in S_3 clock. However, the attacker will also report false queuing time on TC_2 port receiving traffic from S_3. Thus, the attacker will report a new residence time value ρ_2'' to be used in the *correctionField* of S_3 *Report* message. The attacker chooses the value $\rho_2'' = \rho_2 - \mu$ to adjust the previously introduced error. To demonstrate how this will go unnoticed by the NTR, assume that S_3 issues the *Report* message at time $t_1 = (t_{GMC} \pm \epsilon) + \mu$ and the NTR receives the *Report* at t_2 where $t_2 = t_{GMC} + d$ where d is the path delay from S_3 to NTR and can be expressed as $d = \lambda_{S_3-TC_2} + \rho_2 + \lambda_{TC_2-TC_0} + \rho_0 + \lambda_{TC_0-NTR}$. The value carried in the *correctionField*

is $\lambda_{S_3-TC_2}+\rho_2''+\lambda_{TC_2-TC_0}+\rho_0$. Using Eq. 2.2, NTR computes the following offset:

$$\begin{aligned}
\delta &= t_2 - t_1 - (\lambda_{S_3-TC_2}+\rho_2''+\lambda_{TC_2-TC_0}+\rho_0+\lambda_{TC_0-NTR}) \\
&= t_{GMC} + (\lambda_{S_3-TC_2}+\rho_2+\lambda_{TC_2-TC_0}+\rho_0+\lambda_{TC_0-NTR}) \\
&\quad - t_1 - (\lambda_{S_3-TC_2}+\rho_2''+\lambda_{TC_2-TC_0}+\rho_0+\lambda_{TC_0-NTR}) \\
&= t_{GMC} + \rho_2 - t_1 - \rho_2'' \\
&= t_{GMC} + \rho_2 - (t_{GMC} \pm \epsilon + \mu) - (\rho_2 - \mu) \\
&= t_{GMC} + \rho_2 - t_{GMC} \mp \epsilon - \mu - \rho_2 + \mu = \pm\epsilon
\end{aligned}$$

Thus, this stealthy attack will avoid detection by the NTR.

To protect against this type of attacks we need to use an alternative communication path for the delivery of slave reports to the NTR. This path should be in a different security domain, and separated from that used for regular PTP message exchange. We can use the existing infrastructure to communicate *Report* messages from the slaves to the NTR for monitoring purposes only. This is possible due to redundancy constraints enforced through IEC 61850-90-4 [56]. Thus, through the separation of GMC-slave path from slave-NTR path, we can protect the slave to NTR communication and enable the detection of cyber attacks against time synchronization.

5.5.4 Attack on slave clock

A compromise of a slave clock that is responsible of issuing *Report* messages allows the NTR to detect the attack based on the communicated timestamp in a similar fashion to the other attacks. However, an attack against a slave clock that does not issue periodic reports will go undetected unless this slave issues a *Report* or *Report_Resp* to the NTR.

Thus, to avoid such a situation, the NTR can be configured to periodically select random slaves to address with *Report_Req* messages and check their synchronization status.

Remark 5.1. Monitoring PTP slaves time in a PTP network enables the detection of cyber attacks on PTP services.

Proof Cyber attacks on PTP impacts slave clocks, and results in lack of synchronization at those clocks. As a result, slave clocks are either ahead or behind that of the GMC by an error μ. Without loss of generality, assume that the impacted slave clocks are behind the GMC by μ. Monitoring the slaves time, in the form of collection of timestamped messages, reduces this case to that of the delay attack (see Section 5.5.2), and thus can be detected.

5.6 Approach Evaluation

The evaluation of the proposed approach is two-fold. In the first part, we prepare a model of the extended PTP using timed automata. Through this model, we formally verify relevant properties that ensure the soundness of the protocol after extension. Secondly, using Omnet++ network simulator, we perform an experimental study of the proposed approach. We report on those experiments in the following sections.

5.6.1 Approach Modeling and Verification

We used automatic formal verification, namely model checking to formally investigate the correctness of the relevant properties of the proposed solution. We verified safety properties such as freedom from deadlock (Eq. (5.5)), as well as liveness properties (Eq. (5.6) - (5.9)). Model-checking is a well-established enumerative formal verification technique based on the exploration of the state space of the system under study to verify properties of interest. For this, the state space need to be represented as a transition system, while the properties of interest are generally specified in an adequate temporal logic then verified using Model-checkers.

To verify our approach, we used UPPAAL [70], a well established framework for the modeling and formal verification of real-time systems. UPAAL uses timed automata augmented with bounded variables for a better expressiveness. The framework offers a graphical user interface to graphically model real-time systems, a simulator and a model-checker for a subset of the CTL temporal logic which is sufficient in general to specify most properties of interest. We modeled the behavior of each participating component of our system, such as the Master, the NTR and slaves separately. The model of the whole system is obtained by properly synchronizing these individual models using the concept of communication channels offered in UPPAAL.

Using the built model, we verified several properties pertaining to PTP functionality along with that of the proposed extension. Note that the validity of the properties we checked is independent of the number of slaves. In fact, in our model, each slave and its communication channels with both the master and the NTR are totally independent of those of other slaves. This assumption is legitimate considering that the network is capable of assuring such an independence. Therefore, whatever is concluded for one slave can be generalized to all other slaves. We checked the following properties:

Correctness Property This means that extending PTP does not violate its primary functionality. This is validated through equations (5.5) and (5.6). Eq. (5.5) ensures that the extended protocol is deadlock-free using the *A[]* operator which validates that a property holds for all paths in the system, while Eq. (5.6) verifies that each slave receives and processes each GMC *Sync* message, and synchronizes its clock accordingly.

$$A[]\ \ not\ deadlock \tag{5.5}$$

$$Master.syncSent\ \rightarrow\ slave.synchronized \tag{5.6}$$

Combining these two properties together, we claim that the proposed extension maintains the performance and functionality of PTP. On the other hand, there is still a need to evaluate this extension in an environment resembling a substation while using real hardware deployed at the substation. However, the realization of such an experiment is of extreme difficulty as access to PTP implementations of intelligent electronic devices (IEDs) and transparent clocks is restricted to the manufacturing utility. Thus, we consider such an exercise, although essential, beyond our capabilities.

Attack detection Through these properties, we can verify that whenever an attack takes place, it is detected by the NTR. This is verified through Eq. (5.7) where the NTR goes to state indicating attack on slave through delay in network.

$$\begin{gathered} network\_MasterToSlave.pathDelayComputed\ \wedge \\ network\_MasterToSlave.attackDelay > 0 \rightarrow NTR.slaveProblem \end{gathered} \tag{5.7}$$

Moreover, we verify the NTR capability to successfully detect attacks on single or multiple clusters through the conjunction of Eq. (5.7) and Eq. (5.8)

$$\begin{aligned} NTR\_slave.slaveProblem \rightarrow NTR\_witnessSlave.multiClusterProblem \\ \wedge\ NTR\_witnessSlave.slaveTimeProblem \\ \vee\ NTR\_witnessSlave.singleClusterProblem \end{aligned} \tag{5.8}$$

In a similar manner, we verify the NTR capability of detecting attacks targeting the master through Eq. (5.9).

$$\begin{aligned} &\mathit{Master}.\mathit{syncSent} \wedge \ \mathit{Master}.\mathit{attackDelay} > 0 \\ &\qquad\qquad\qquad \to \mathit{NTR\_Master}.\mathit{MasterNotSync} \end{aligned} \tag{5.9}$$

False Positives It is worth noting that the same properties used above can be evaluated to verify the availability of false positives. We verify those properties in the absence of attacks, and the model successfully indicated that absence.

We also verify as anticipated that the NTR fails to detect a well-orchestrated attack where the attacker injects a delay on the path taken by the *Sync* message to the slave and cancels it when this slave reports back to the NTR. However, this attack is extremely difficult to implement.

To conclude, the proposed extension was formally verified to preserve the protocol functionality, and to effectively detect attacks on PTP time synchronization.

5.6.2 Simulation Results

To validate the usefulness of our approach through Omnet++ simulation [121], we modify the PTP implementation provided by [122] to introduce our detection mechanism, and carry-on different cyber attacks. We arranged 30 slaves in 6 bays using the hierarchical star network topology presented in IEC 61850-90-4 [106] and shown in Fig. 5.5. We consider each bay to be a cluster and select one slave per bay to send *Report* messages to the NTR. We performed a set of experiments to check the capability of the NTR to detect ongoing attacks, evaluate the time needed for detection, the synchronization error at the slave clocks at the time of detection, and network overhead while varying the frequency at which slaves push reports to the NTR. For each attack, different TCs and network links are separately compromised, and we report on the average of the recorded results for those attack instants.

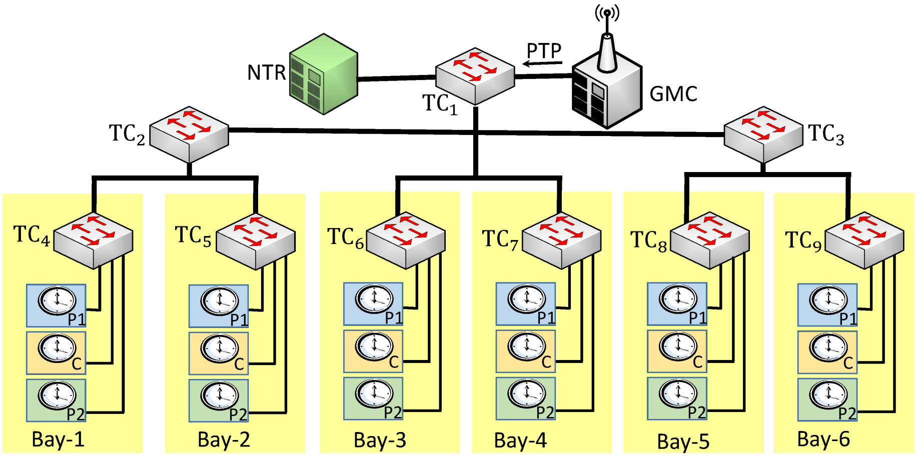

Figure 5.5: A sketch of the experimental network setup [106].

The usefulness of the approach is tested against attacks on GMC, TCs, and the delay attack. We consider that synchronization errors less than 1 millisecond are acceptable, and thus the NTR threshold was set to 1 millisecond. For the attack on GMC, we introduced an error of 2 milliseconds in the GMC time, while for the delay attack we introduced a variable error uniformly distributed between 1 and 4 milliseconds. For the TC attack, the compromised TCs report a multiple of the PTP message residence time. All the attacks are initiated after 5 seconds from the start of simulation.

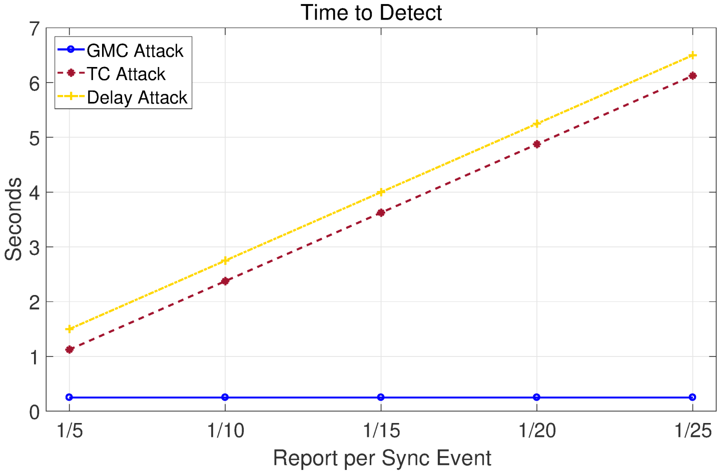

Figure 5.6: Time needed to detect an attack.

The time needed to detect an ongoing attack is affected by the nature of the attack, and is measured as the time elapsed since the start of the attack until it is flagged by the NTR. This is shown in Fig. 5.6, which shows that the NTR successfully detects all the attacks after a relatively short time. As Fig. 5.6 shows, the time needed to detect the attack on GMC is constant and not affected by the frequency of slave reports. This is a result of the ability of the NTR to detect this attack based on the periodic *Sync* messages issued by the GMC. However, the time needed to detect delay attack and attack on TCs is inversely proportional to the frequency of slave reports. Starting with a frequency of 1 report every 5 synchronization events, and decreasing that gradually to 1 report every 25 synchronization events, results in an increase in the time needed to detect the ongoing attack from 1 seconds up to almost 7 seconds. This is due to the fact that the NTR is neither served by the compromised TCs nor the network links, and thus has to wait for slave reports to compute the associated offset and detect the attack.

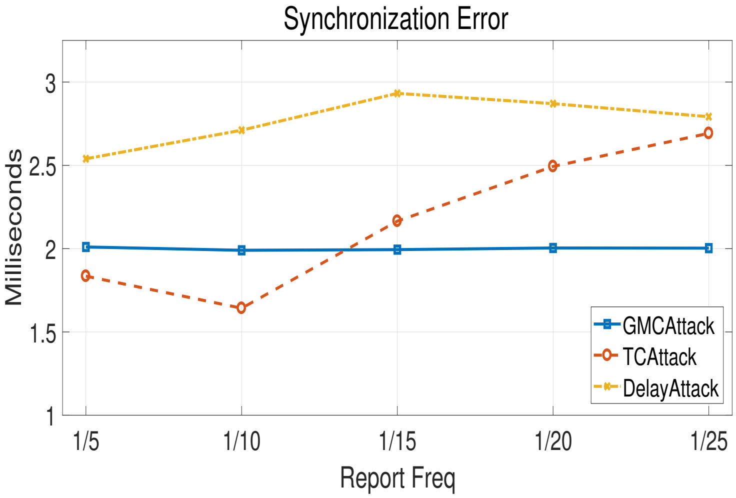

Figure 5.7: Synchronization error at attack detection.

On the other hand, we monitor the synchronization error at the slave side at the time of attack detection. The collected results are presented in Fig. 5.7, which shows different impact on the slave clock depending on the nature of the attack. The attack

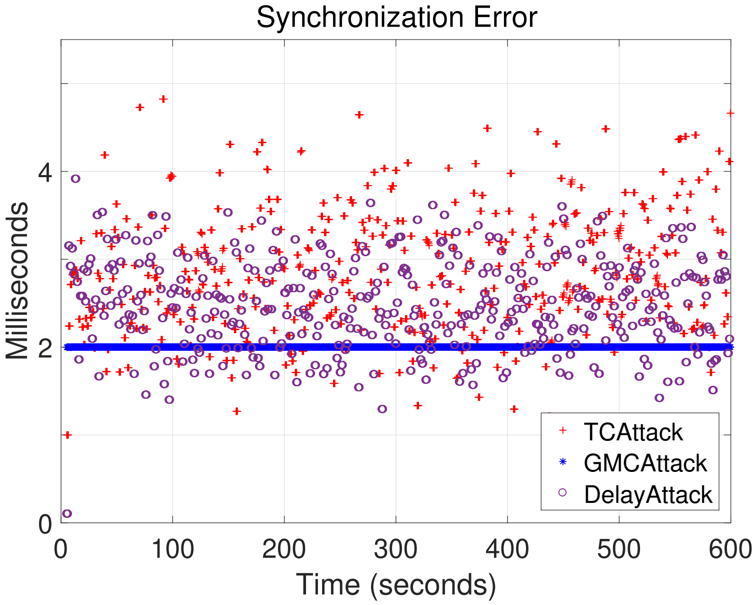

Figure 5.8: Synchronization error due to cyber attacks

on GMC introduces a constant error of 2 milliseconds at the slave side. However, the attack on TC and the delay attack introduce a varying error that reaches a maximum of 3 milliseconds as time progresses. The variation in the introduced error in the case of an attack on TC is a result of the varying packet residence time at the compromised TC, and it reaches a maximum of 2.5 milliseconds when the slaves report frequency is minimum.

To better understand the advantage of early detection of cyber attacks, we have simulated the attacks on the GMC, TC and delay attack and monitored the synchronization error at affected slaves in the absence of our detection mechanism. The attacks were simulated for 10 minutes using a similar setup to that used for Fig. 5.6 and Fig. 5.7, and the collected results are presented in Fig. 5.8. The collected results show that significant errors of the order of 5 milliseconds are introduced at the slave clock on the long run. The attack on the TC is the most effective, while that on the GMC introduces a constant error at the slave side due to the nature of this attack. This error is likely to persist at slave clocks unless the ongoing attacks are detected,

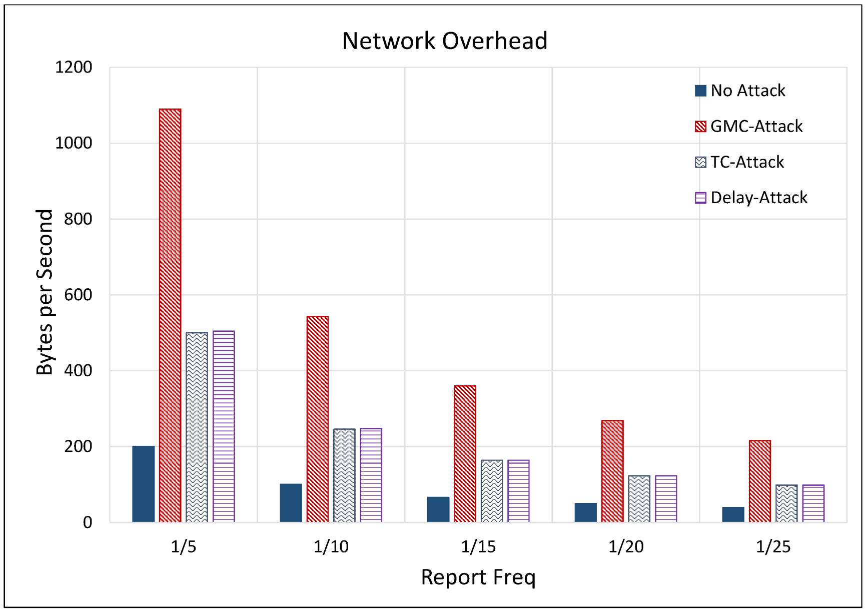

Figure 5.9: Network overhead due to the deployed extension.

and effects mitigated. Thus, there is need for our detection logic to expose cyber attacks targeting PTP.

On the other hand, the introduction of new messages to PTP presents an overhead on the network. To evaluate this overhead, we tested the protocol extension in the presence and absence of different attacks. We run each experiment on a network of 30 slaves for 100 seconds, evaluate the average number of bytes per second introduced into the network due to the detection mechanism, and we present the outcome of this experiment in Fig. 5.9. As expected, with a low frequency of reports, little traffic is introduced in the network. However, with the highest frequency of reports, we can see that the additional traffic sums up to around 1 KB per second for the worst case, and does not account for more than 200 Bytes per second in the absence of attacks. Such an addition can be easily handled by Gigabit networks as in IEC 61850 substation. Moreover, we can notice that an attack on the GMC will result in the largest overhead since the entire network will be affected and thus results in more report requests and replies traversing the network. Nevertheless, based on the collected results, we can claim that our extension is light weighted.

To conclude, experimental results validate the capability of our proposed detection mechanism in successfully detecting cyber attacks against PTP, and relates the detection speed to the amount of introduced reports as additional network traffic.

5.7 Conclusion

Time synchronization through PTP is of significant importance for the safe and stable operations of the smart grid. Nonetheless, PTP is subject to a variety of cyber attacks that threaten its deployment and impact its services. Addressing such threats, and securing PTP services is as critical as the synchronization accuracy PTP provides. PTP main weakness is its lack of awareness of the connected slaves synchronization status. The connected slaves trust all PTP traffic collected from the network and use it to adjust their clocks. Thus, in the presence of an attack, PTP slave clocks will be manipulated and the attack will go unnoticed. In this chapter, we addressed PTP security and proposed a protocol extension to enable the detection of attacks against PTP. Our presented solutions leverages slave clocks capabilities and the network design requirements to introduce a new functionality to the PTP network, and collect feedback from slave clocks that enables the detection of attacks targeting PTP time synchronization. The proposed extension is modeled using UPPAAL model checker, and formally verified using significant CTL properties. Moreover, the efficiency of the proposed extension is demonstrated through numerical simulation.

Chapter 6

Exploiting The Vulnerability of Relative Data Alignment in Phasor Data Concentrators to Time Synchronization Attacks

In the previous chapters, we addressed vulnerabilities targeting PTP as main time synchronization mechanism in the smart grid. In this chapter, we demonstrate how an attack on time synchronization can be used to affect essential smart grid components and subsystems. Using time synchronization, we formulate an attack on the relative data alignment scheme used by phasor data concentrators (PDCs) to aggregate and stream phasor measurements. This attack leverages the specification governing the PDC functionality in a wide area measurement system. Using a phasor measurement unit (PMU) as attack surface, a malicious player injects an attack vector in the PMU timing to manipulate the PDC functionality and force it to drop phasors received from benign PMUs. We expose the PDC relative data alignment scheme to this attack, formulate the attack surface and vector selection as a linear program (LP), and demonstrate the impact of this attack on system observability. We conduct experimental results on standard IEEE test systems to demonstrate the attack semantics and impact. In addition, we manifest the described attack in a case study using hardware-in-the-loop (HIL) co-simulation environment. The outcome of those tests validate the vulnerability of PDC aggregation scheme to this attack and its significance in targeting system observability.

6.1 Introduction

The power grid as we know it is witnessing a major evolution to adapt digital communication, and transform into a smart grid. This smart grid is mainly characterized by power transmission efficiency, increase in reliability, short service restoration times, usage of renewable energy, and better customer interaction among others. Such characteristics are enabled through the integration of advanced sampling technologies to monitor, protect, and control the state of the grid in real time. Thus, the traditional power system is emanating as a cyber-physical entity on top of a wide set of monitoring devices, mainly phasor measurement units (PMUs), communication networks, and system protection solutions namely WAMPAC.

Wide area monitoring, protection and control (WAMPAC) describe one category of the advanced techniques that involves the use of system wide information and the communication to a remote location to counteract the propagation of the large disturbances [31, 114]. Advanced WAMPAC technologies enable the implementation of electrical grids that realize the needs for sustainable energy delivery and enhanced power system performance [18]. However, the reliability of WAMPAC applications depends largely on the accuracy of the phasors computed by the Phasor Measurement Units (PMUs) [38], for which timing plays a critical role. Indeed, synchronized PMUs have become a reality in the control room of utilities worldwide [2]. PMU measurements are sampled synchronously across the entire power grid based on a coordinated Universal Time (UTC). This allows the presentation of a high accuracy system wide snapshot through the collection and alignment of synchronized phasor measurements.

Precise timing is essential in power systems for grid monitoring and situational awareness. The availability of time synchronized measurements from PMUs allows system operators to monitor and coordinate the operations of various grid assets, along with the protection of grid components. As outlined by the North American Synchrophasor Initiative (NASPI) [2], PMUs need access to reliable Coordinated Universal Time (UTC) to allow synchrophasor applications to time-align the voltage and current time series data for analysis and coordinated activity over a wide geographical area. Moreover, NASPI illustrates that all elements of a synchrophasor system (both PMUs and the associated phasor data concentrators) must continually access a common and accurate timing source linked to Coordinated Universal Time (UTC) [2].

However, along with the advantages accurate time synchronization brings to the smart grid, time synchronization mechanisms introduce several security concerns into the smart grid. This is due to the fact that inaccuracy in a PMU's timing adversely affects the PMU measurements, especially the estimation of phase angles of the measured quantity [2] and hence WAMPAC applications processing those measurements. Moreover, the feasibility of cyber attacks targeting timing mechanisms have been documented in the literature [62, 87]. Those attacks demonstrate the vulnerability of PMU technologies to cyber-security threats, and the candidacy for use as an attack surface to target WAMPAC applications [12].

On the other hand, a typical phasor measurements collection network comprises PMUs and phasor data concentrators (PDCs). PDCs are responsible of aggregating and relaying synchrophasor data collected from several PMUs, and arranged in a hierarchical structure through a communication infrastructure. Thus, altering the functionality of a PDC could affect a large set of data, and eventually leave a larger impact on the system functionality. This gives rise to the question, could we exploit the specifications of PMU to PDC data supply and alignment to target WAMPAC applications?

In this chapter, we demonstrate the feasibility of leveraging the specifications of phasor data alignment at the PDC level to impact WAMPAC applications namely system observability. This demonstration uses time synchronization as a stepping stone to perform an attack on data aggregation at the PDC level. The introduced attack consists of identifying a PMU as an attack surface, along with an attack vector injected at the identified PMU that results in discarding genuine measurements received from trusted PMUs at the PDC. To formulate this attack, we follow a linear programming approach. We model the PDC data alignment specifications, and identify all the attack components. Moreover, we evaluate the presented attack on standard IEEE benchmark systems.

6.1.1 Novel Contributions

The main contributions of this paper can be outlined as follows:

1. We identify the vulnerability of PDC phasor alignment to attacks through the analysis of the C37.244-2013 [7] standard, and exploiting relative data alignment dependency on the early arrivals of phasor measurements. Through an attack on time synchronization, we demonstrate the feasibility of leveraging the specifications of phasor data alignment at the PDC level to impact WAMPAC applications namely system observability.

2. We formulate the identified attack as a linear program with a special emphasis on its impact on power system observability. The formulated attack consists of identifying a PMU as an attack surface, along with an attack vector injected at the identified PMU that results in poisoning a target PDC, and eventually discarding genuine measurements received from trusted PMUs at the poisoned PDC.

3. We evaluate the defined attack on IEEE benchmark systems, and show its impact on system observability. Moreover, we demonstrate the attack semantics and impact through hardware-in-the-loop simulation.

The remainder of this chapter is structured as follows. Our system model is introduced in Section 6.2 followed by the research problem definition in Section 6.3. Section 6.4 defines our threat model. Problem formulation and mathematical model are covered in Section 6.5. A discussion of countermeasures for the described vulnerability are provided in Section 6.6. Section 6.7 portrays the experimental results. Concluding remarks are provided in Section 6.8.

6.2 System Model

The system under study is the wide area measurement, protection, and control (WAMPAC) system in its two main components: monitoring and measurement devices including phasor measurement units (PMUs) and phasor data concentrators (PDCs), and applications using those measurements for protection and control purposes mainly system observability.

The WAMPAC measurement component is arranged in a tree structure with the central PDC as the root, and PMUs as the leaves. PMUs are located at buses in the power system to collect and send readings of different system parameters (e.g.

voltage, current). Intermediate nodes constitute of local and regional PDCs that aggregate and forward measurements received from PMUs and other PDCs respectively. Accurate time synchronization is a key component of this system. The collection and alignment of measurements demands that the PMUs are accurately synchronized to a unified time source. Such a synchronization is made possible through the use of global navigation satellite systems (GNSS) or the Precision Time Protocol (PTP) [4]. Local PDCs aggregate measurements received from several PMUs into data streams before handing them to higher PDCs in the hierarchy. Those streams are prepared based on the timestamps associated with the measurements using the relative data alignment approach as shown in Fig. 6.1. Upon the arrival of a complete data frame carrying the expected data timestamp (PMU 1 in figure), the PDC starts a local timer referred to as the relative wait time. The PDC expects to receive data frames from other PMUs referring to the same timestamp within the specified waiting time. The used timer ensures that measurements are forwarded to portray the system status in real time, and thus drops any delayed measurements not arriving within the waiting time window.

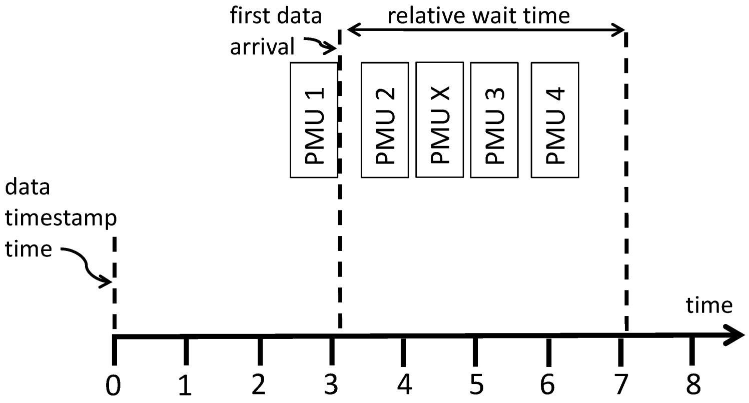

Figure 6.1: Relative data alignment as defined by C37.244-2013 [7].

The aggregation of streams received at the central PDC provides system operators with a large scale view of the system status. To ensure system observability, measurements provided by PMUs from different system buses are needed. We consider power system observability as defined in [73] as:

$$O_i = \sum_{j \in N_u} a_{ij} x_j; \quad i \in B \tag{6.1}$$

where O_i represents the observability at bus i, B is defined as the set of all system buses, N_u as the set of PMUs located in the system, x_j is a binary variable that identifies if a PMU is installed at bus j, a_{ij} is a connectivity parameter defined as:

$$a_{ij} = \begin{cases} 1 \text{ if } i = j \ or \ (i,j) \ \in \ L \\ 0 \ otherwise \end{cases}$$

where L is the set of all transmission lines in the system.

$O_i \geq 1$ implies that bus i is observable either through a PMU located on bus i, or through measurements provided by PMUs located on buses j connected to bus i with some transmission lines (i.e. $a_{ij} = 1$ for some $j \in B$). The power system is said to be observable if the observability function O_i for each bus is greater than or equal to 1:

$$O_i \geq 1, \quad \forall i \in B \tag{6.2}$$

6.3 Problem Definition

Time synchronization is a key enabling technology for WAMPAC operations. It is used by PMUs to supply timestamped measurements, and later used by PDCs to aggregate those measurements. Based on the timestamps carried by the phasor measurements, the PDC drops measurements due to the following reasons:

- *Time synchronization error:* Phasor measurements carrying an invalid time tag as judged by the PDC are not aligned with other phasor measurements into data streams, nor later forwarded to other PDCs in the hierarchy.

- *Latency:* Phasor measurements carrying a valid timestamp but do not arrive within latency expectations at the PDC. As defined by C37.244-2013 [7], latency starts when the first complete data message with a given timestamp arrives at the PDC. This latency is specified in the form of waiting time associated with the first arrival as illustrated in Fig. 6.1.

On the other hand, and as defined by C37.244-2013 [7], valid measurements received at a PDC from a PMU are handled in one of the possible ways:

1. *Add to an existing data stream:* This resembles the case when a PDC receives a timestamped frame that fits into the data stream being aggregated. The PDC aligns the new frame with the existing ones based on the frame's timestamp before forwarding the data stream to other PDCs.

2. *Setup a new data stream:* This resembles the case of first data arrival of a frame with a given timestamp at the PDC. This arrival triggers a relative wait time for the preparation of a new data stream. Thus, the PDC expects frames that align with the first arriving one to be delivered within the defined time window. The wait time is controlled by a timer that triggers the forwarding operation upon expiry.

Thus, measurements supplied by a PMU do not autonomously setup a new data stream nor are included in an existing one. Those measurements may be dropped, and as such a drop of the measurements supplied by a PMU, or group of PMUs, can result in a loss of the system observability.

The vulnerability of the system to the loss of phasor measurements opens a window for an attacker to target WAMPAC and system observability. Leveraging the importance of accurate synchronization for phasor data collection and aggregation, a time synchronization-based attack targeting PMUs can consequently result in dropping the measurements supplied by those PMUs.

Thus, we can define two attack instances:

6.3.0.1 Drop Target PMU Measurements

By introducing a synchronization error at the target PMU, the measurements supplied by this PMU are eventually dropped by the collecting PDC. The PMU timestamps the measurements at time t'_0 as opposed to the real time t_0. For $(t'_0 < t_0)$, the PDC considers those measurements as late arrivals for data streams timestamped with (t'_0), and eventually drops them. If the measurements arrive at time t where $t < t'_0$, the PDC observes the timing error, and will eventually drop those measurements.

6.3.0.2 Drop Other PMU(s) Measurements

This is a more stealthy version of an attack on time synchronization. Through this attack, the attacker leverages the system functionality, in particular relative data alignment and aggregation at the PDC, to impact data collection from other PMUs rather than the one he is directly targeting. As shown in Fig. 6.2, by targeting *PMU X* timing, the PDC handles *PMU X* data frame as a first arrival, and initiates a relative wait time interval. Upon the expiration of this interval, the PDC drops data frames received from *PMU 3* and *PMU 4*. To perform this attack, an attacker has to design and introduce an attack vector α to the time of the targeted PMU. The vector α should be formed in a way so that data frames from the targeted PMU are not dropped by the receiving PDC. Those frames will be processed and result in dropping measurements received from other PMUs, connected to the same PDC, due to latency constraints enforced by the waiting period.

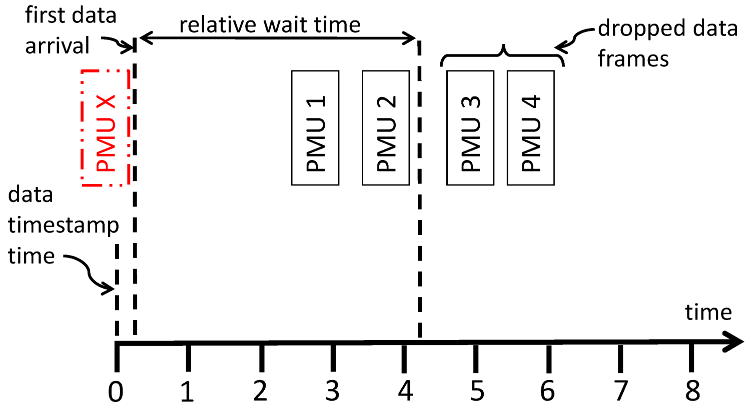

Figure 6.2: Data frames dropped using relative data alignment.

The vulnerability of smart grid to attacks on time synchronization is well documented in the literature in the form of GPS spoofing [87], and as a time synchronization attack [128]. However, we want to analyze and design attack vectors that would fulfill the stealthy attack instances. To design those vectors, we will consider the specifications of the phasor measurements collection and aggregation system.

Consider a WAMPAC system composed of N PMUs and P PDCs, where each PDC is connected to a subset of the N PMUs. Let PDC C be connected to U_i PMUs where $1 \leq i \leq N$. Consider a snapshot of the system at an instant of time t. PMUs U_1 to U_i timestamp measurements at time t before sending them to PDC C. In the absence of an attack, frames sent by the PMUs arrive at PDC C at $t + d_j$ after some variable network delay $d_j(j = 1,i)$. The first frame that reflects the system status at t initiates the assembly of a new data stream, and triggers a latency timer l. Data frames arriving before l expires are aligned according to their timestamps, and sent to the PDC receiving streams from C.

Suppose that PMU U_1 is targeted by an attacker, and is no longer synchronized to the coordinated universal time (UTC). The attacker introduces an error α into U_1 time. We can distinguish between the following cases:

- $\alpha < 0$: Frames sent by U_1 arrive at C carrying a timestamp $(t-\alpha)$. PDC C either considers those frames as redundant one representing the system state at $(t - \alpha)$, or late arrivals for the same system state, and consequently drops those measurements.

- $\alpha > 0$: In this case, frames sent by U_1 and arriving at C might be dropped if the timestamp $(t+\alpha)$ they carry is more than the system time which may be represented as $(t + d_1)$. However, for a proper selection of α, the attacker can spoof C as a first data arrival, and enforce the creation of a new frame to aggregate measurements collected at $(t+\alpha)$. Thus, with an optimal choice of the attack vector α, the attacker can trigger an early aggregation of data frames and initiate PDC C latency timer. Hence, upcoming measurements from other PMUs timestamped at $(t+\alpha)$ will be late arrivals and hence dropped by PDC C. Since it results in dropping measurements from several PMUs, such an attack vector represents a stealthy threat to system observability. Thus, determining this attack vector is a challenge for the attacker considering the system requirements and specifications.

We aim at addressing the selection of an appropriate attack vector that would threaten system observability starting with an attack on a single PMU, or multiple PMUs if needed, while taking into consideration the system requirements for phasor measurements aggregation and power system observability.

6.4 Threat Model

To carry out attacks on WAMPAC through time synchronization, we consider an active and capable attacker. We assume that the attacker is either an external entity or an insider with malicious intents. He has the expertise to perform long-term reconnaissance operations required to learn the environment, and execute a highly synchronized, multistage, multisite attack [71]. The attacker is aware of the time synchronization mechanism used by the PMUs, and is capable of targeting this mechanism to introduce the required attack vector at the target PMU. In addition to that, our attacker is aware of the WAMPAC topology. He is knowledgeable about the PMU to PDC network, and can perform the necessary estimations related to the network traffic and conditions.

Moreover, the attacker is interested in targeting WAMPAC through exploiting the PDC phasor alignment algorithms. He chooses this approach to impact the system while being undetected. If the attacker wishes to execute his attack through the introduction of additional hardware or software to the system, he has the expertise necessary to choose suitable attack locations to perform his attack, and avoid the available physical security measures.

6.5 Problem Formulation

Our problem formulation aims at leveraging the system specifications, and dependability on phasor measurements to obscure the system observability. Thus, from an attacker perspective, we aim at minimizing system observability through the definition of an attack vector, and selection of the attack surface. To achieve this target, we consider a snapshot of the WAMPAC system where phasor measurements are timestamped by PMUs at a time instance t, and define this problem through a linear program (LP) that identifies the PMU to be targeted along with the needed attack vector.

6.5.1 Nomenclature

In the problem formulation, we use the following notation:

N_u : set of available PMUs

N_c : set of available PDCs

N_u^c : set of PMUs connected to PDC c

B : set of power system buses

A : power system connectivity matrix

$\Delta = (\delta_u^c)$: matrix of estimated path delay

δ_u^c : path delay between u $\in N_u$ and $c \in N_c$

$X = (x_i)$: PMU placement vector

$$x_i = \begin{cases} \text{1 if a PMU is placed on bus i} \in \text{B} \\ \text{0 otherwise} \end{cases}$$

τ^c : Timer threshold for PDC c $\in N_c$

ϵ : small positive number

M : large positive number

We define the following decision variables:

$\alpha_u \in \mathcal{R}$: attack vector injected to timestamp of PMU $u \in N_u$

$\alpha^c \in \mathcal{R}$: α_u as seen at PDC $c \in N_c$

$\delta_{u^*}^c$: estimated path delay between attack target u^* and PDC c

$$y_u = \begin{cases} \text{1 if a PMU } u \in N_u \text{ is attack target} \\ \text{0 otherwise} \end{cases}$$

$$v_u^c = \begin{cases} \text{1 if measurements from PMU } u \text{ are aligned by PDC } c \\ \text{0 otherwise} \end{cases}$$

$$v_u = \begin{cases} \text{1 if measurements from PMU } u \in N_u \text{ are valid} \\ \text{0 otherwise} \end{cases}$$

$$b_i = \begin{cases} 1 \text{ if bus } i \in B \text{ is observable} \\ 0 \text{ otherwise} \end{cases}$$

$$z^c = \begin{cases} 1 \text{ if PDC } c \in N_c \text{ is attack target} \\ 0 \text{ otherwise} \end{cases}$$

d_u^c : time window to deliver measurements from PMU $u \in N_u^c$ to PDC $c \in N_c$

6.5.2 LP Formulation

The objective of the attacker is to affect the system observability through the introduced attack vector. This objective is presented in Eq. (6.3) where the attacker aims at minimizing the number of observed buses in the system.

$$Minimize \sum_{i \in B} b_i \tag{6.3}$$

The observability of a bus is determined through Eq. (6.1). However, the calculation of the observability variable O_i for each bus i is affected by the availability of valid measurements from PMUs positioned at bus i or neighboring buses. This is reflected through the usage of a decision variable that determines whether those measurements are valid or not as can be seen in Eq. (6.4).

$$O_i = \sum_{u \in N_u} a_{iu} x_u v_u \qquad \forall i \in B \tag{6.4}$$

The observability of each bus in the system, as defined in Eq. (6.4), is represented in Eq. (6.5) as a binary value that reflects whether a bus is observable or not. If the observability value of bus i is positive, then bus i is observable. While a zero value for observability indicates that this bus is not observable.

$$\begin{aligned} & b_i \geq \frac{O_i}{M} \\ & b_i \leq O_i \qquad \forall i \in B \end{aligned} \tag{6.5}$$

The attack surface is restricted to one PMU in Eq. (6.6). This limits the attacker capabilities to targeting a single PMU.

$$\sum_{u \in N_u} y_u = 1 \tag{6.6}$$

To successfully perform the attack, the timestamped measurements from the target PMU should be considered valid by the receiving PDC. Thus, those measurements should not carry a timestamp that represents future time. This is enforced through Eq. (6.7), where the introduced attack vector does not exceed the time needed by the measurements sent from PMU u to reach PDC c.

$$\alpha_u \leq \max(\delta_u^c) \qquad \forall u \in N_u, c \in N_c \tag{6.7}$$

The attack vector introduced into the timestamps of the target PMU should not be null, in contrast to other PMUs which have a null attack vector. This is ensured through Eq. (6.2) where the targeted PMU gets a positive attack vector, and the attack vector introduced at other PMUs is set to zero.

$$\begin{aligned} & \alpha_u \leq M y_u \\ & \alpha_u \geq \frac{y_u}{M} \qquad \forall u \in N_u \end{aligned} \tag{6.8}$$

To calculate the time window available for non-targeted PMUs to deliver their measurements, we need to identify the network delay between the target PMU and the receiving PDC. This is done through Eq. (6.9).

$$\delta^c_{u^*} = \sum_{u \in N^c_u} y_u \delta^c_u \qquad \forall c \in N_c \tag{6.9}$$

PDC c in the network is affected by the attack on a PMU u only if c receives measurements from the attack target u. We identify the PDCs that are affected by this attack in two steps. The first step is represented in Eq. (6.10) where we identify the attack vector implicitly available at PDC c as α^c.

$$\alpha^c = \sum_{u \in N^c_u} y_u \alpha_u \qquad \forall c \in N_c \tag{6.10}$$

The second step is presented through Eq. (6.11), where the attack vector calculated in Eq. (6.10) is reflected as a binary variable that defines whether a PDC c is indirectly under attack or not.

$$\begin{aligned} z^c &\leq \alpha^c \\ z^c &\geq \frac{\alpha^c}{M} \qquad \forall c \in N_c \end{aligned} \tag{6.11}$$

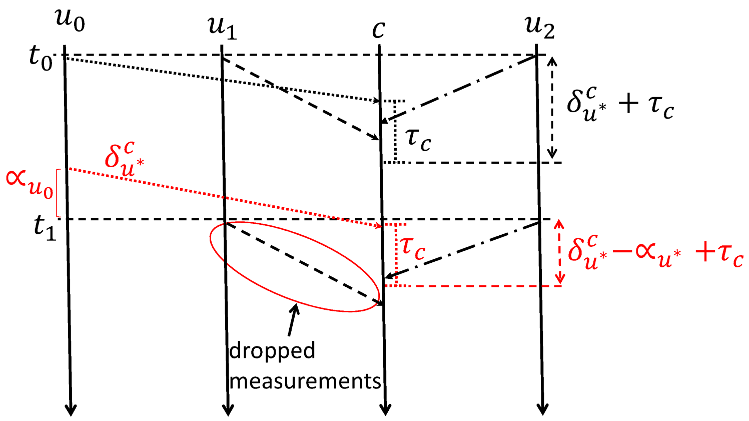

Figure 6.3: Measurement delivery restrictions in presence of attack.

We define a time window for each PMU sending measurements to a targeted PDC as per Eq. (6.12). This window represents the difference between the time available for a measurement sent by PMU u to arrive at PDC c, and that needed by the measurement to traverse the network from PMU u to PDC c. This equation is illustrated in Fig. 6.3 where the early arrival of a timestamped measurement from the PMU under attack causes a reduction in the time available for measurements from other PMUs to make it to the destination PDC.

$$d_u^c = \delta_{u^*}^c + \tau^c - \alpha^c - \delta_u^c \qquad \forall u \in N_u^c \tag{6.12}$$

The validity of measurements sent from a PMU to a PDC is decided by three factors.

1. *PMU is under attack:* Measurements received from the targeted PMU are the first arrivals corresponding to the rogue timestamp, and thus should be accepted and aligned by the receiving PDC. This is reflected through the use of y_u variable in Eq. 6.13, which validates of the measurements sent from PMU u to PDC c.

2. *PDC is under attack:* The validity of measurements at a PDC not receiving phasors from the targeted PMU is not affected by the attack. Thus, measurements arriving at such a PDC should be considered valid. This is reflected through the use of z_c variable in Eq. 6.13, which validates of the measurements sent from PMU u to PDC c. However, measurements sent to a PDC that receives phasors from the targeted PMU may be considered valid or not depending on their time of arrival as we see next.

3. *Late arrivals:* Measurements arriving at a PDC are considered late arrivals and invalid if their time window defined in Eq. 6.12 is negative. This indicates that measurements sent from such PMUs arrive after the expiration of the PDC timer.

Those three factors are combined in Eq. (6.13) which determines if measurements sent by PMU u to PDC c are valid.

$$\begin{aligned} v_u^c &\geq \frac{d_u^c(1-y_u)z^c}{M} + \epsilon \\ v_u^c &\leq 1 + \frac{d_u^c(1-y_u)z^c}{M} \qquad \forall u \in N_u^c, \forall c \in N_c \end{aligned} \tag{6.13}$$

Phasors collected by a PMU are multicast to a set of PDCs. Those measurements might be invalid at a PDC due to the attack, and valid at another which does not receive measurements from the PMU under attack. Thus, the measurements of such a PMU should be considered valid and account for in system observability. This is ensured through Eq. 6.14 which determines whether measurements from PMU u are considered valid based on the validity of these measurements at each PDC.

$$\begin{aligned} v_u &\geq v_u^c \\ v_u &\leq \sum_{c \in N_c} v_u^c \qquad \forall u \in N_u, c \in N_c \end{aligned} \tag{6.14}$$

As an outcome of this formulation, we can identify the needed attack parameters. Injecting the identified attack vector at the target PMU will result in a drop in system observability, and thus a stealthy attack on WAMPAC system.

6.6 Countermeasures

To protect WAMPAC systems against the formulated vulnerability, several strategies can be implemented. A solid protection scheme for time synchronization signals supplied to the PMUs will prevent the manipulation of this signal and eventually poisoning the PDC. Such a scheme can take the form of equipping the PMUs with multiple time sources, and thus enabling them to detect the discrepancy in time

Table 6.1: PMU to PDC Connectivity

Test System	PDC						
	1	2	3	4	5	6	7
IEEE 14-Bus	2, 6	7, 9	-	-	-	-	-
IEEE 24-Bus	2, 8, 10	3, 16	21, 23	-	-	-	-
IEEE 30-Bus	1, 2, 6, 9	10, 12, 15, 19	25, 27	-	-	-	-
IEEE 57-Bus	1, 50	4, 6, 15	20, 38, 47	32, 36, 41, 57	9, 24, 25, 28, 53	-	-
IEEE 118-Bus	3, 5, 9, 12	15, 17, 21, 23, 28, 30, 114	34, 37, 40, 45, 49	52, 56, 62, 64	68, 71, 75, 77, 80	85, 86, 91, 94	101, 105, 110

synchronization signals during attack. On the other hand, the C37.244-2013 standard provides an alternate scheme for data alignment namely alignment to absolute time reference. In this scheme, the PDC is accurately synchronized and controls the waiting time for measurements arrival rather than being triggered by early arrivals. The deployment of such a scheme avoids poisoning the PDC with early arrivals yet leaves the system vulnerable to delay attacks against PMU measurements and variations in the PMU to PDC path delay. Finally, a more advanced solution can be deployed at the PDC level where an analysis of the history of PMU measurement early and later arrivals along with a correlation to the delay experienced in the communication network, can determine a dynamic relative waiting time that enables the collection of much needed measurements and hardens the system against such attacks.

6.7 Experimental Results

Our assessment of the vulnerability of PDC data alignment to the formulated attack in wide area measurement systems is two-fold. The first set of experiments are performed using the mathematical model on standard IEEE test systems to identify the attack surface (PMU), target (PDC) and attack vector. In the second set of experiments, we consider a case study of the IEEE 24-Bus system where we perform the formulated attack using HIL co-simulation. We report on the results collected from both tests in the following sections.

6.7.1 Numerical Evaluation

To expose the vulnerability of relative data alignment for phasor measurements at PDCs, we implemented the developed model and related simulation programs using Java and IBM CPLEX concert technology. The simulations were executed on a windows machine with Intel Core i7 CPU running at 2.67GHz and equipped with 12 GB of RAM. We conducted experiments for the 14-bus, 24-bus, 30-bus, 57-Bus, and 118-Bus IEEE test systems (for details about those systems, interested readers are referred to [109, 35, 95]).

In our system setup, we consider the electric grid to be observable when all of its system states are uniquely identified [9]. System states are observed through phasor measurements received from PMUs located on selected power buses. To identify the needed number of PMUs and their respective locations for full system observability, we use the results presented by Chakrabarti et al. in [30, 32]. Based on those results, the optimal PMU number and corresponding bus locations for the IEEE test systems are indicated in Table 6.2. Moreover, to determine the number of PDCs for each system along with the set of PMUs associated with those PDCs, we adopt the results presented by Fesharaki et al. in [48, 82]. The number of PDCs for each test system along with its connected PMUs are presented in Table 6.1. We note that a hyphen in a row means that the PDC in the respective column is not available for this test system. To complete the input for the mathematical model, we need to specify the network topology and the estimated delays over the PMU-PDC communication network connections based on WAMS needs and specifications. In this respect, we make use of several results in the literature especially [130, 124, 33, 65, 129] to assign the suitable delay for the communication paths in use. The assigned PMU to PDC path delays are in the range of 40ms - 60ms for the different systems. On the other hand, taking into consideration that research outcomes available in the literature [130, 124, 33, 65, 129] define network end-to-end delay, we do not use any network topology specifications as we consider the delay to be the more decisive factor regardless of the topology.

Table 6.2: Optimal PMU number and placement for IEEE test systems

Test System	Number of PMUs	Bus Locations
IEEE 14-Bus	4	2, 6, 7, 9
IEEE 24-Bus	7	2, 3, 8, 10, 16, 21, 23
IEEE 30-Bus	10	1, 2, 6, 9, 10, 12, 15, 19, 25, 27
IEEE 57-Bus	17	1, 4, 6, 9, 15, 20, 24, 25, 28, 32, 36, 38, 41, 47, 50, 53, 57
IEEE 118-Bus	32	3, 5, 9, 12, 15, 17, 21, 23, 28, 30, 34, 37, 40, 45, 49, 52, 56, 62, 64, 68, 71, 75, 77, 80, 85, 86, 91, 94, 101, 105, 110, 114

Using the previously described setup, we run the model for the IEEE 14, 24, 30, 57 and 118 Bus test systems. In those experiments, we have fixed PMU locations, estimated communication delays, and connectivity to PDC. We have varied the PDC timer for each test, and computed the respective attack surface and vector. The variations of the PDC timer are inspired by the values used in [129], and the collected results are presented in Tables 6.3, 6.4, 6.5, 6.6, and 6.7.

As can be seen from Table 6.3, the model identifies PMU located on bus 2 as potential attack surface to target PDC-1 for different timer values. The injected attack vector varies with the timer and forces PDC-1 to drop measurements from

Table 6.3: Attack Vector and impact for 14 bus system

PDC Timer	Target(PMU,PDC)	Attack Vector	Unobserved Buses
20 ms	(2, 1)	20 ms	6, 11, 12, 13
30 ms	(2, 1)	30 ms	6, 11, 12, 13
40 ms	(2, 1)	45 ms	6, 11, 12, 13
50 ms	-	-	-
60 ms	-	-	-

other PMUs, namely PMU located on bus 6. Dropping those measurements results in the inability to observe buses 6, 11, 12 and 13 as identified by the model. The model did not identify a suitable attack vector for 50 and 60 ms timer values. This is due to the small values assigned to the network delays. Any additional delay in the network will enable the identification of a suitable attack vector.

Table 6.4: Attack Vector and impact for 24 bus system

PDC Timer	Target(PMU,PDC)	Attack Vector	Unobserved Buses
20 ms	(2, 1)	30 ms	5, 7, 8, 10, 11
30 ms	(8, 1)	30 ms	2, 4, 5, 6, 11
40 ms	(8, 1)	40 ms	2, 4, 5, 6, 11
50 ms	(10, 1)	40 ms	2, 4
60 ms	-	-	-

Table 6.5: Attack Vector and impact for 30 bus system

PDC Timer	Target (PMU,PDC)	Attack Vector	Unobserved Buses
20 ms	(19,2)	20 ms	12, 13, 14, 15 16, 17, 21, 22, 23
30 ms	(19,2)	30 ms	12, 13, 14, 15 16, 17, 21, 22, 23
40 ms	(19,2)	40 ms	12, 13, 14, 15 16, 17, 21, 22, 23
50 ms	(19,2)	40 ms	17, 21, 22, 23
60 ms	-	-	-

Similar results are collected for the IEEE 24 and 30-Bus systems and presented in Tables 6.4 and Table 6.5 respectively. We can identify the selected PMU as attack surface along with the attack vector to be injected in its timestamps. The impact of this attack is presented in both tables as a set of unobservable system buses. As can be seen from the presented results, with the increase in the timer value, more measurements can arrive on time at the PDC and this enlarges the set of observable system buses. For a timer value of 60 ms, the model could not identify an attack vector with the estimated network delays that are close to the lower bound.

Table 6.6: Attack Vector and impact for 57 bus system

PDC Timer	Target(PMU,PDC)	Attack Vector	Unobserved Buses
20 ms	(53, 5)	20 ms	9, 10, 12, 23, 24, 25, 26, 27, 28, 29, 30, 55
30 ms	(25, 5)	45 ms	9, 10, 12, 23, 26, 27, 28, 29, 52, 53, 54, 55
40 ms	(57, 4)	45 ms	31, 32, 33, 34, 35, 36, 40, 41, 42, 43
50 ms	(9, 5)	35 ms	23, 24, 25, 26, 27, 28, 29, 30
60 ms	(15, 2)	45 ms	7

Attack surface and vector for the IEEE 57, and 118-Bus systems are presented in Tables 6.6 and 6.7 respectively. With the increase in system size, the chance to identify an attack surface and formulate an attack vector increases. This formulation results in the identification of an attack target for different PDC timer values and a subsequent set of unobservable power buses. As can be concluded from the results, a short timer value representing little tolerance for delays at the PDC results in more impactful attacks and larger sets of unobservable buses.

To reflect the impact of the identified attack vector on the system observability, we plot the system observability versus the change in PDC timer for the different systems in Fig. 6.4. As illustrated in Fig. 6.4, through the exposed vulnerability in the PDC phasor alignment, it is possible to devise an attack that leaves a deep impact on system

Table 6.7: Attack Vector and impact for 118 bus system

PDC Timer	Target(PMU,PDC)	Attack Vector	Unobserved Buses
20 ms	(28, 2)	20 ms	13, 15, 17, 18, 20, 21, 22, 23, 24, 25, 26, 30, 31, 32, 113, 114, 115
30 ms	(28, 2)	30 ms	13, 15, 17, 18, 20, 21, 22, 23, 24, 25, 26, 30, 31, 32, 113, 114, 115
40 ms	(28, 2)	40 ms	13, 15, 17, 18, 20, 21, 22, 23, 24, 25, 26, 30, 31, 32, 113, 114, 115
50 ms	(56, 4)	50 ms	2, 3, 4, 5, 6, 7
60 ms	(64, 4)	55 ms	60, 62, 67

observability. For different PDC specifications, it is possible to craft an attack vector and force a decrease in system observability. The impact of this attack decreases with an increase in the PDC tolerance to delayed measurements. Nevertheless, with the increase in system size and complexity of communication network, it is possible to leverage the PDC dependency on early arrivals to initiate an attack against phasor data alignment at PDC level.

6.7.2 HIL Simulation Case Study

Using our smart grid testbed, simulating both power and communication networks, we prepared a setup composed of 7 PMUs (2 physical, 5 software) and 3 physical PDCs to demonstrate the outcome produced by the mathematical model for the IEEE 24-Bus system. The PMU to PDC connectivity, and PMU locations are configured as described in Tables 6.1 and 6.2 respectively. We experiment the case where the PDC timer is set to 50 ms, and inject the attack vector identified by the model as shown in Table 6.4. As indicated by the model, we target the PDC (PDC-1) collecting measurements from PMUs located at buses 2, 8, 10 by injecting the attack vector to

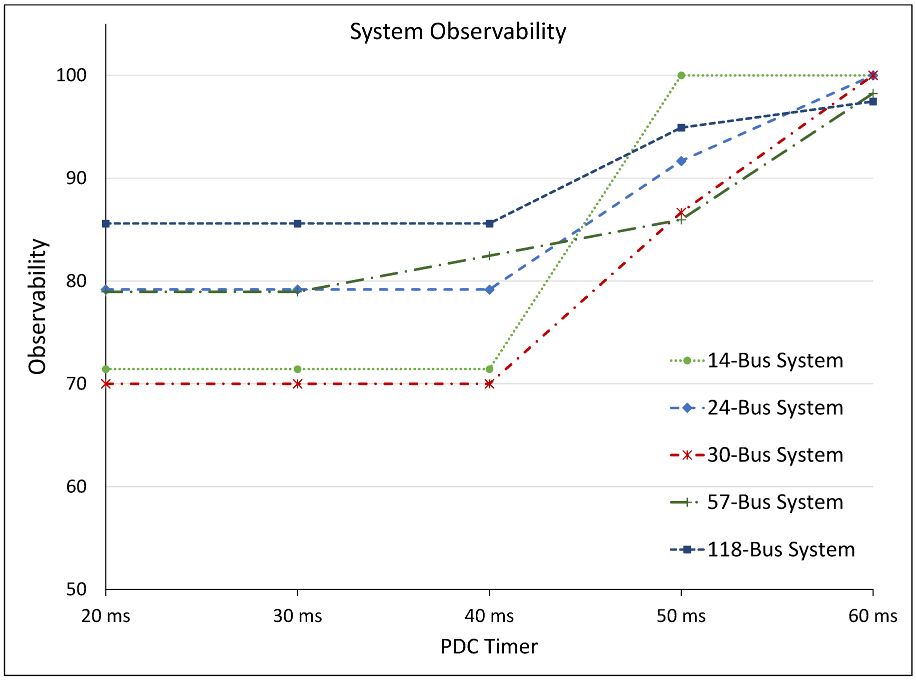

Figure 6.4: Post attack system observability.

PMU located at Bus 10. We deploy two physical PMUs (A & B) on Buses 2 and 8, and another software PMU on Bus 10. We present the measurements collected at the control center in Fig. 6.5, where target PMU is located at the upper left corner, and PMUs A & B are located at the bottom of the figure. We initiate the attack by modifying the timestamp of the software PMU at the time indicated in Fig. 6.5. As seen in Fig. 6.5, in the presence of the attack, there is a significant drop in measurements collected from PMUs A & B while measurements collected from PMU under attack are not affected. During the attack, some measurements from PMUs A & B are not dropped by the PDC. This is due to the variation in the network delay which enables the measurements sent by those PMUs to arrive at PDC-1 before its timer expires, since the model gives the least attack vector based on the network delay at an instant of time. If we increment the attack vector, we can invoke the PDC to drop more of the collected measurements. However, based on the measurements collected at the control center, it is hard to identify the attack

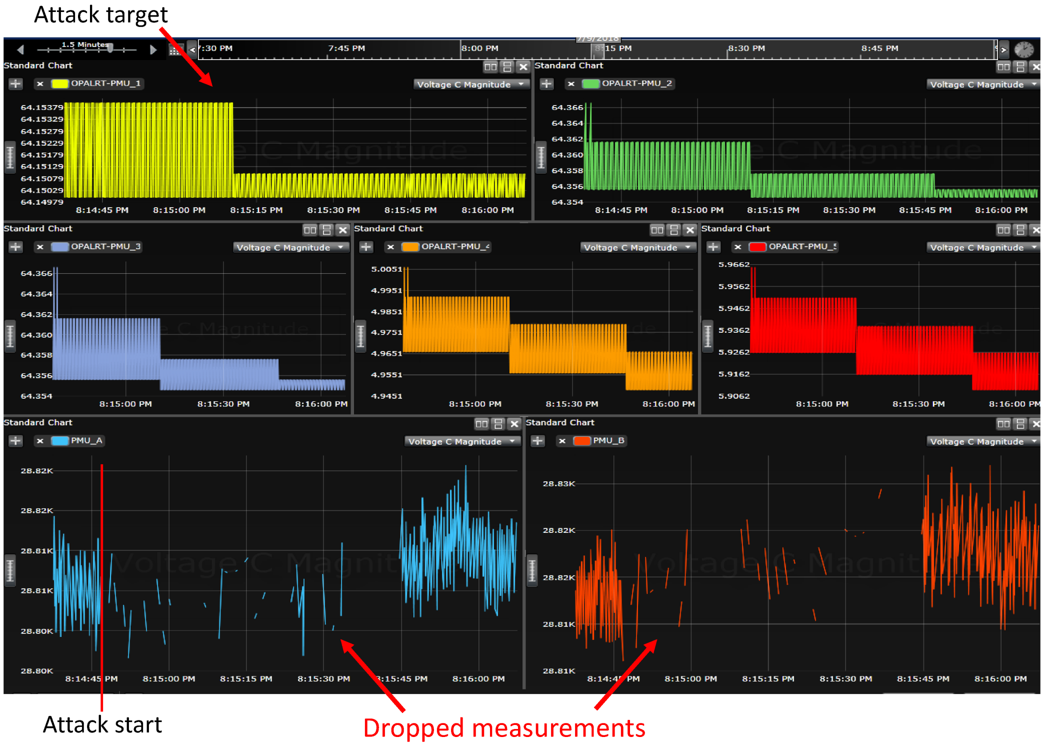

Figure 6.5: Measurements delivered to control center.

surface since the measurements from PMU under attack are aligned and delivered as in normal conditions, while the dropped measurements are from benign PMUs. The outcome of this experiment validates the vulnerability of PDC relative data alignment to time synchronization attacks.

6.8 Conclusion

In this chapter, we exposed the vulnerability of relative data alignment scheme used by PDCs in WAMS to time synchronization attacks. The formulated attack uses a PMU as an attack surface to inject a well-crafted attack vector into its time, thus generating rouge timestamped phasor measurements to poison the collecting PDC. As a direct impact of this attack, the poisoned PDC will drop benign phasor measurements transmitted by uncompromised PMUs. This impact escalates to result in a drop in the system observability thus threatening the availability and reliability of the smart grid. The proposed attack was mathematically formulated as linear program, evaluated on standard IEEE test systems, and demonstrated using HIL co-simulation. Results from the performed tests manifest the exposed vulnerability and the formulated attack.

Chapter 7

Discussion and Future Directions

7.1 Discussion

This thesis addressed several challenges and concerns associated with the inauguration of the smart grid. Mainly, it focused on threats pertaining to time synchronization being a key enabler for smart grid systems. We have considered a substation conforming to the design requirements mandated by IEC 61850, and addressed several security concerns associated with the Precision Time Protocol (PTP), the recommended mechanism for time synchronization at the substation level. At first, through Chapter 2 we presented an overview of available time synchronization mechanisms along with applications dependent on precise timing in the smart grid. Chapter 2 provided a survey of the existing literature work addressing security concerns associated with PTP. We concluded Chapter 2 with a gap analysis identifying potential research problems to be addressed in order to harden PTP security. Those gaps were later addressed in other chapters aiming at providing a security aware precision time synchronization protocol.

Motivated by the lessons learned from the literature survey, Chapter 3 addressed a gap in the authentication scheme associated with PTP through its security extension - Annex K. We highlighted the vulnerability of PTP to fake timestamp injection through a compromised component of the PTP network. Using the available network and system management solutions imposed by IEC 62351, we proposed a detection mechanism through defined SNMP data types and objects. The defined objects extend the PTP MIB and enable the collection of relevant information related to timestamps communicated by the master clock through synchronization messages. The proposed detection scheme was tested on an NSM testbed for a substation, and the collected results demonstrated the usefulness of the proposed solution.

Next, Chapter 4 of this thesis addressed one of the well-known threats to message exchange-based time synchronization protocols, the delay attack. By exploiting the design requirements for an IEC 61850 substation, we devised an approach to detect the occurrence of such attack at the substation level. Using PRISM formal model checker, we developed a model resembling time synchronization under PTP and the proposed detection approach. The prepared model enabled us to evaluate quantitative and qualitative properties related to PTP security. Moreover, we introduced a mechanism to mitigate the impact of the delay attack at slave clocks. The mitigation mechanism relies on maintaining a history of recent synchronization records at connected devices, then use this history to alleviate the attack impact and maintain clock synchronization in the presence of the attack. The proposed mechanisms were validated using formal model checking, and demonstrated using an implementation of PTP on a physical setup.

Next, to address threats associated with the PTP attack surface as a whole and motivated by the outcome of Chapter 4, we proposed an extension to PTP that allows the collection of feedback from slave devices, and thus monitor their synchronization status. The proposed extension was presented in Chapter 5 in the form of new functionality in the PTP network and additional message exchanges. This extension leveraged slave capabilities to send and receive timestamped messages, and the presence of accurate back-up time sources, to periodically collect and analyze timestamped messages from slave clocks and assess their synchronization status. We theoretically demonstrated the ability to detect attacks on PTP using the proposed extension. Moreover, we modeled the proposed extension using timed automata, and evaluated liveness and correctness properties on the developed model. The evaluated properties assured that the extension does not affect PTP functionality, and allows to detect attack on PTP. A final evaluation for the presented extension was done using Omnet++ simulation, where the timeliness of attack detection and introduced network overhead were estimated. The collected results supported the usefulness of the approach, and its efficiency in detecting cyber attacks against PTP.

On the other hand, the solutions presented in Chapters 3, 4, and 5 provide a valuable opportunity to develop and assemble a time synchronization security solution for the smart grid especially at the substation level. Indeed, the PTP extension presented in 5 can be merged with the mitigation scheme presented in 4 to enable the detection and mitigation of different attacks detected through the NTR functionality. Moreover, an additional layer of security is enabled through the introduced SNMP MIBs, thus protecting PTP against insider threats through a compromised network component. However, the realization of this approach is challenged by the ability to test it and evaluate its outcome in a substation environment, while using a real setup that mimics the substation and its components. Although we have a Hardware-In-the-Loop (HIL) testbed for smart grid security at the Security Research Center, this challenge persists due to the inability to modify existing PTP implementations on the available hardware due to restrictions imposed by the manufacturers. Moreover, applying the proposed solutions to existing deployment of PTP remains a major challenge due to the difficulty of incorporating such changes to devices installed in the field. Yet, those modifications can be taken into consideration for upcoming versions of PTP. Nevertheless, the solutions proposed can be adapted as guidelines to develop a security metric that reflects the security posture of time synchronization through PTP at the substation level. Indeed, we consider such a metric as part of our future research goals.

Finally, driven by the use of time synchronization in wide area monitoring networks, we addressed the vulnerability of such systems to time synchronization based attacks. We identified an attack that exploits relative data alignment scheme used by phasor data concentrators. The defined attack uses time synchronization as an attack surface to manipulate the timing of a selected phasor measurement unit, and later poison the PDC collecting measurements from this PMU. The identified vulnerability is formulated using a linear program that identifies the PMU and PDC to be targeted, along with the suitable attack vector while considering the overall system observability. As a direct impact of this attack, the poisoned PDC will invalidate and drop measurements collected from other PMUs. The dropped measurements will consequently result in a drop in system observability, and hence put the system under risk. The formulated problem was evaluated on different IEEE benchmark systems, and validated using HIL simulation of the smart grid. The collected results demonstrate the PDC relative data alignment scheme to the identified vulnerability.

7.2 Future Work

Over the past two decades, the research community has been actively setting the path for the grid of the future, a smart, green, failure and attack resilient, and self-healing grid. We have witnessed the birth of advanced technologies and applications that enable the migration towards the smart grid. Those applications and technologies are driven by one pulse, accurate time synchronization.

In this thesis, we have addressed several challenges associated with a prominent time synchronization mechanism for the smart grid. The final section of this thesis highlights potential challenges and directions for future research.

7.2.1 PTP Usage in Wide Area

Although PTP is recommended for use at the substation level, and GNSS is more favorable for synchronization in wide area networks, securing the use of PTP in wide area networks improves the grid resiliency and avoids having GNSS as a single point of failure. PTP use in wide area networks is faced by many challenges, mainly the inability to ensure a symmetric path and thus accurate time synchronization. In addition to that, PTP will be faced by the need to authenticate devices available in its network especially with the illustrated vulnerability of PTP to fake timestamp injection through a compromised clock. This calls for a revised PTP authentication scheme, and innovative solutions that protect the PTP network from compromise through delay attacks.

On the other hand, the presence of PTP in wide area networks along with GNSS provides an excellent opportunity to incorporate data collected from both systems into cyber attacks prevention, detection, and mitigation mechanisms. This is mainly driven by the ubiquitous nature of GNSS, and the ability of PTP to use existing communication deployments rather than dedicated cabling.

7.2.2 Characterization of Attacks Impact on Power Systems

The impact of attacks on time synchronization varies based on the application processing measurements that are affected by the attacks. There is a need to characterize this impact to devise suitable detection and mitigation strategies. This characterization can be achieved by experimenting with different attack models and power applications through HIL simulation, and later an analysis of the observed system behavior and response to the different attacks.

7.2.3 Time Synchronization Data Analysis

Analyzing the data generated from time synchronization whether through GNSS or PTP is one of the threads worth exploring. Time synchronization is a very frequent activity that generates a lot of data depending on the receiving clock quality, and the communication medium status. This data is affected by attacks, and through developing an environment aware approach that can relate the collected data with the network and system status at real time, we might be able to detect the occurrence of attacks. Moreover, by learning the synchronization data variations as imposed by network and system conditions, we can create models for the expected synchronization behavior and detect any anomalies that violate those expectations. Such a model, when linked with models developed for other system behavior, will aid in developing a defense mechanism for cyber attacks on time synchronization, and make way for a secure time synchronization mechanism.

Bibliography

[1] BeiDou Navigation Satellite System. `http://en.beidou.gov.cn/`.

[2] North American Synchrophasor Initiative. `https://www.naspi.org/`.

[3] IEEE Standard for Synchrophasors for Power Systems. *IEEE Std C37.118-2005 (Revision of IEEE Std 1344-1995)*, pages 1–65, 2006.

[4] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. *IEEE Std 1588-2008 (Revision of IEEE Std 1588-2002)*, pages c1–269, July 2008.

[5] IEEE Standard Profile for Use of IEEE 1588 Precision Time Protocol in Power System Applications. *IEEE Std C37.238-2011*, pages 1–66, July 2011.

[6] IEEE Standard Profile for Use of IEEE 1588 Precision Time Protocol in Power System Applications. *IEEE Std C37.238-2011*, July 2011.

[7] IEEE Guide for Phasor Data Concentrator Requirements for Power System Protection, Control, andMonitoring. *IEEE Std C37.244-2013*, pages 1–65, May 2013.

[8] Precision Time Protocol Daemon - ptpd. `https://sourceforge.net/projects/ptpd/`, 2016.

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

[10] S. Achanta, S. T. Watt, and E. Sagen. Mitigating GPS Vulnerabilities. `https://www.selinc.com/WorkArea/DownloadAsset.aspx?id=104197`, 2015.

[11] M. Akbarzadeh and M. A. Azgomi. A framework for probabilistic model checking of security protocols using coloured stochastic activity networks and pdetool. In *Telecommunications (IST), 2010 5th International Symposium on*, pages 210–215. IEEE, 2010.

[12] M. S. Almas, L. Vanfretti, R. S. Singh, and G. M. Jonsdottir. Vulnerability of synchrophasor-based WAMPAC applications' to time synchronization spoofing. *IEEE Transactions on Smart Grid*, 2017.

[13] F. Aminifar, M. Fotuhi-Firuzabad, A. Safdarian, A. Davoudi, and M. Shahidehpour. Synchrophasor Measurement Technology in Power Systems: Panorama and State-of-the-Art. *Access, IEEE*, 2:1607–1628, December 2014.

[14] Arbiter Systems, Inc. IRIG-B Time Code Accuracy and Connection Requirements with comments on IED and system design considerations. `http://www.arbiter.com/files/product-attachments/irig_accuracy_and_connection_requirements.pdf`.

[15] Arbiter Systems, Inc. Time in the Power Industry: How and Why We Use It. `http://www.arbiter.com/files/product-attachments/TimeInThePowerIndustry.pdf`, 2009.

[16] Arbiter Systems, Inc. TIMING SIGNALS, IRIG-B AND PULSES. `http://www.arbiter.com/files/product-attachments/pd0043200_timing_signals_overview.pdf`, 2013.

[17] R. Atkinson and S. Kent. Security Architecture for the Internet Protocol. *RFC 2401*, 1998.

[18] S. Bahramirad, J. Svachula, and J. Juna. Trusting the data: ComEd's Journey to embrace analytics. *IEEE power and energy magazine*, 12(2):107–111, 2014.

[19] S. Barreto, A. Suresh, and J.-Y. Le Boudec. Cyber-attack on packet-based time synchronization protocols: The undetectable Delay Box. In *Instrumentation and Measurement Technology Conference Proceedings (I2MTC), 2016 IEEE International*, pages 1–6. IEEE, 2016.

[20] S. Basagiannis, P. Katsaros, A. Pombortsis, and N. Alexiou. Probabilistic model checking for the quantification of DoS security threats. *Computers & Security*, 28(6):450–465, 2009.

[21] S. Basagiannis, S. Petridou, N. Alexiou, G. Papadimitriou, and P. Katsaros. Quantitative analysis of a certified e-mail protocol in mobile environments: A probabilistic model checking approach. *Computers & Security*, 30(4):257–272, 2011.

[22] A. Basu, S. Bensalem, M. Bozga, B. Delahaye, and A. Legay. Statistical abstraction and model-checking of large heterogeneous systems. *International Journal on Software Tools for Technology Transfer*, 14(1):53–72, 2012.

[23] B. Baumgartner, C. Riesch, and M. Rudigier. IEEE 1588/PTP: The Future of Time Synchronization in the Electric Power Industry. In *PAC World Conference*, 2012.

[24] P. Behr. Outage on Quebec power grid traced to airborne attacker. `http://www.eenews.net/stories/1060020352`, 2015.

[25] G. Behrmann, A. David, and K. Larsen. A tutorial on uppaal. *Formal methods for the design of real-time systems*, pages 33–35, 2004.

[26] C. Bonebrake and L. O'Neil. Attacks on GPS Time Reliability. *Security Privacy, IEEE*, 12(3):82–84, May 2014.

[27] D. Broman, P. Derler, A. Desai, J. C. Eidson, and S. A. Seshia. Endlessly circulating messages in IEEE 1588-2008 systems. In *2014 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, pages 7–12. IEEE, 2014.

[28] D. Broman, P. Derler, and J. Eidson. Temporal issues in cyber-physical systems. *Journal of the Indian Institute of Science*, 93(3):389–402, 2013.

[29] A. Bykhovsky and J. H. Chow. Power system disturbance identification from recorded dynamic data at the Northfield substation. *International journal of electrical power & energy systems*, 25(10):787–795, 2003.

[30] S. Chakrabarti and E. Kyriakides. Optimal Placement of Phasor Measurement Units for Power System Observability. *Power Systems, IEEE Transactions on*, 23(3):1433–1440, Aug 2008.

[31] S. Chakrabarti, E. Kyriakides, T. Bi, D. Cai, and V. Terzija. Measurements get together. *IEEE Power and Energy Magazine*, 7(1), 2009.

[32] S. Chakrabarti, E. Kyriakides, and D. G. Eliades. Placement of synchronized measurements for power system observability. *IEEE Transactions on Power Delivery*, 24(1):12–19, 2009.

[33] M. Chenine, E. Karam, and L. Nordstrom. Modeling and simulation of wide area monitoring and control systems in IP-based networks. In *Power & Energy Society General Meeting, 2009. PES'09. IEEE*, pages 1–8. IEEE, 2009.

[34] Y. Chompoobutrgool, L. Vanfretti, and M. Ghandhari. Survey on power system stabilizers control and their prospective applications for power system damping using Synchrophasor-based wide-area systems. *European transactions on electrical power*, 21(8):2098–2111, 2011.

[35] R. Christie. Power system test archive. *University of Washington*, 1999.

[36] J. E. Dagle. Data management issues associated with the august 14, 2003 blackout investigation. In *Power Engineering Society General Meeting, 2004. IEEE*, pages 1680–1684. IEEE, 2004.

[37] P. Daly, P. N. Misra, B. Parkinson, and J. Spilker. GPS and global navigation satellite system (Glonass). *Global Positioning System: Theory and applications.*, 2:243–272, 1996.

[38] J. De La Ree, V. Centeno, J. S. Thorp, and A. G. Phadke. Synchronized phasor measurement applications in power systems. *IEEE Transactions on smart grid*, 1(1):20–27, 2010.

[39] A. Desai, D. Broman, J. Eidson, S. Qadeer, S. A. Seshia, P. Derler, P. Garg, P. Madhusudan, V. Gupta, and E. Jackson. Approximate Synchrony: An Abstraction for Distributed Time-Synchronized Systems. Technical report, University of California, Berkeley, 2014.

[40] A. Desai and S. Seshia. Modelling and Analysis Of IEEE 1588, Nov. 2013.

[41] T. Dierks and E. Rescorla. The transport layer security (tls) protocol. In *IETF RFC 4346*. Citeseer, 2006.

[42] A. S. Dobakhshari and A. M. Ranjbar. A Wide-Area Scheme for Power System Fault Location Incorporating Bad Data Detection. *Power Delivery, IEEE Transactions on*, 30(2):800–808, 2015.

[43] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208, 1983.

[44] A. dos Santos, B. Soares, C. Fan, M. Kuipers, S. Sabino, A. M. Grilo, P. R. B. Pereira, M. S. Nunes, and A. Casaca. Characterization of substation process bus network delays. *IEEE Transactions on Industrial Informatics*, 14(5):2085–2094, 2018.

[45] M. J. Dworkin. Sha-3 standard: Permutation-based hash and extendable-output functions. Technical report, 2015.

[46] L. Fang, Y. Yamagata, and Y. Oiwa. Evaluation of A Resilience Embedded System Using Probabilistic Model-Checking. *arXiv preprint arXiv:1405.1703*, 2014.

[47] V. Fernández, M.-J. García-Martínez, L. Hernández-Encinas, and A. Martín. Formal verification of the security of a free-space quantum key distribution system. In *Proceedings of the International Conference on Security and Management (SAM)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (World-Comp), 2011.

[48] F. H. Fesharaki, R. A. Hooshmand, and A. Khodabakhshian. A new method for simultaneous optimal placement of PMUs and PDCs for maximizing data transmission reliability along with providing the power system observability. *Electric Power Systems Research*, 100:43–54, 2013.

[49] K. Fodero, C. Huntley, and D. Whitehead. Secure, Wide-Area Time Synchronization. In *proceedings of the 12th Annual Western Power Delivery Automation Conference, Spokane, WA*, 2010.

[50] G. Gaderer, A. Treytl, and T. Sauter. Security aspects for IEEE 1588 based clock synchronization protocols. In *IEEE International Workshop on Factory Communication Systems (WFCS06), Torino, Italy*, pages 247–250. Citeseer, 2006.

[51] E. Ghahremani and I. Kamwa. Dynamic State Estimation in Power System by Applying the Extended Kalman Filter With Unknown Inputs to Phasor Measurements. *Power Systems, IEEE Transactions on*, 26(4):2556–2566, Nov 2011.

[52] J. L. Gutiérrez-Rivas, J. López-Jiménez, E. Ros, and J. Díaz. White rabbit hsr: a seamless sub-nanosecond redundant timing system with low-latency data capabilities for smart grid. *IEEE Transactions on Industrial Informatics*, 2017.

[53] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A tool for automatic verification of probabilistic systems. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 441–444. Springer, 2006.

[54] J. Hong et al. Cyber-Physical Security Test Bed: A Platform for Enabling Collaborative Cyber Defense Methods. In *PACWorld Americas Conference*, 2015.

[55] IEC-TC57. IEC 61850-5: Communication networks and systems for power utility automation - Part 5: Communication requirements for functions and device models. *International Electro technical Commission Std*, 2010.

[56] IEC-TC57. IEC 61850-90-4: Communication networks and systems for power utility automation - Part 90-4: Network engineering guidelines. *International Electro technical Commission Std*, 2010.

[57] IEC-TC57. IEC 61850: Communication networks and systems for power utility automation. *International Electro technical Commission Std*, 2010.

[58] IEC-TC57. IEC 62351-7: Data and Communication Security – Security Through Network and System Management. *International Electro technical Commission Std*, 2016.

[59] D. M. Ingram, F. Steinhauser, C. Marinescu, R. R. Taylor, P. Schaub, D. Campbell, et al. Direct evaluation of IEC 61850-9-2 process bus network performance. *Smart Grid, IEEE Transactions on*, 3(4):1853–1854, 2012.

[60] E. Itkin and A. Wool. A security analysis and revised security extension for the precision time protocol. *IEEE Transactions on Dependable and Secure Computing*, 2017.

[61] Q. Jiang, X. Li, B. Wang, and H. Wang. PMU-based fault location using voltage measurements in large transmission networks. *Power Delivery, IEEE Transactions on*, 27(3):1644–1652, 2012.

[62] X. Jiang, J. Zhang, B. Harding, J. Makela, and A. Dominguez-Garcia. Spoofing GPS Receiver Clock Offset of Phasor Measurement Units. *Power Systems, IEEE Transactions on*, 28(3):3253–3262, Aug 2013.

[63] I. Kamwa, S. Samantaray, and G. Joos. Development of rule-based classifiers for rapid stability assessment of wide-area post-disturbance records. *Power Systems, IEEE Transactions on*, 24(1):258–270, 2009.

[64] M. Kassouf, L. Dupont, J. Béland, and A. Fadlallah. Performance of the Precision Time Protocol for clock synchronisation in smart grid applications. *Transactions on Emerging Telecommunications Technologies*, 24(5):476–485, 2013.

[65] R. Kateb, P. Akaber, M. Tushar, A.-B. Abdullah, M. Debbabi, and C. Assi. Enhancing WAMS Communication Network Against Delay Attacks. *IEEE Transactions on Smart Grid*, 2018.

[66] V. Kekatos and G. Giannakis. Distributed Robust Power System State Estimation. *Power Systems, IEEE Transactions on*, 28(2):1617–1626, May 2013.

[67] M. Kezunovic and B. Perunicic. Synchronized sampling improves fault location. *Computer Applications in Power, IEEE*, 8(2):30–33, 1995.

[68] N. Kinnock. Galileo: involving Europe in a new generation of satellite navigation services. *Air & Space Europe*, 1(2):3, 1999.

[69] M. Kwiatkowska, G. Norman, and D. Parker. PRISM: probabilistic model checking for performance and reliability analysis. *ACM SIGMETRICS Performance Evaluation Review*, 36(4):40–45, 2009.

[70] K. G. Larsen, P. Pettersson, and W. Yi. Uppaal in a nutshell. *International Journal on Software Tools for Technology Transfer (STTT)*, 1(1):134–152, 1997.

[71] R. M. Lee, M. J. Assante, and T. Conway. Analysis of the Cyber Attack on the Ukrainian Power Grid. *SANS ICS Report*, 2016.

[72] W. Li, M. Ferdowsi, M. Stevic, A. Monti, and F. Ponci. Cosimulation for smart grid communications. *IEEE Transactions on Industrial Informatics*, 10(4):2374–2384, 2014.

[73] H. Lin, C. Chen, J. Wang, J. Qi, D. Jin, Z. Kalbarczyk, and R. K. Iyer. Self-healing attack-resilient PMU network for power system operation. *IEEE Transactions on Smart Grid*, 2016.

[74] B. Liscouski and W. Elliot. Final report on the august 14, 2003 blackout in the united states and canada: Causes and recommendations. *A report to US Department of Energy*, 40(4), 2004.

[75] T. Logsdon. The Navstar Global Positioning System. *The Navstar Global Positioning System, New York, NY (USA), 1992, 272 p.*, 1, 1992.

[76] M. Majidi, M. Etezadi-Amoli, and M. Fadali. A Sparse-Data-Driven Approach for Fault Location in Transmission Networks. *Smart Grid, IEEE Transactions on*, PP(99):1–9, 2015.

[77] J. McGhee and M. Goraj. Smart High Voltage Substation Based on IEC 61850 Process Bus and IEEE 1588 Time Synchronization. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 489–494, Oct 2010.

[78] D. Mills. Network Time Protocol (Version 3) specification, implementation and analysis. 1992.

[79] B. Milosevic and M. Begović. Voltage-stability protection and control using a wide-area network of phasor measurements. *Power Systems, IEEE Transactions on*, 18(1):121–127, 2003.

[80] T. Mizrahi. Time synchronization security using IPsec and MACsec. In *Precision Clock Synchronization for Measurement Control and Communication (ISPCS), 2011 International IEEE Symposium on*, pages 38–43. IEEE, 2011.

[81] T. Mizrahi. A game theoretic analysis of delay attacks against time synchronization protocols. In *Precision Clock Synchronization for Measurement Control and Communication (ISPCS), 2012 International IEEE Symposium on*, pages 1–6. IEEE, 2012.

[82] M. B. Mohammadi, R.-A. Hooshmand, and F. H. Fesharaki. A new approach for optimal placement of PMUs and their required communication infrastructure in order to minimize the cost of the WAMS. *IEEE Transactions on Smart Grid*, 7(1):84–93, 2016.

[83] N. Moreira, A. Astarloa, and U. Kretzschmar. SHA-3 based Message Authentication Codes to secure IEEE 1588 synchronization systems. In *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, pages 2323–2328. IEEE, 2013.

[84] N. Moreira, A. Astarloa, U. Kretzschmar, J. Lazaro, and E. Molina. Securing IEEE 1588 messages with message authentication codes based on the KECCAK cryptographic algorithm implemented in FPGAs. In *Industrial Electronics (ISIE), 2014 IEEE 23rd International Symposium on*, pages 1899–1904. IEEE, 2014.

[85] N. Moreira, J. Lázaro, J. Jimenez, M. Idirin, and A. Astarloa. Security mechanisms to protect ieee 1588 synchronization: State of the art and trends. In *Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS), 2015 IEEE International Symposium on*, pages 115–120. IEEE, 2015.

[86] S. S. Mousavi-Seyedi, F. Aminifar, and S. Afsharnia. Parameter Estimation of Multiterminal Transmission Lines Using Joint PMU and SCADA Data. *Power Delivery, IEEE Transactions on*, 30(3):1077, 2015.

[87] B. Moussa, M. Debbabi, and C. Assi. Security Assessment of Time Synchronization Mechanisms for the Smart Grid. *IEEE Communications Surveys Tutorials*, 18(3):1952–1973, thirdquarter 2016.

[88] B. Moussa, M. Debbabi, and C. Assi. A detection and mitigation model for PTP delay attack in an IEC 61850 substation. *IEEE Transactions on Smart Grid*, 9(5):3954–3965, 2018.

[89] L. Narula and T. Humphreys. Requirements for secure clock synchronization. *IEEE Journal of Selected Topics in Signal Processing*, 2018.

[90] L. Narula et al. Requirements for Secure Clock Synchronization. *IEEE Journal of Selected Topics in Signal Processing*, pages 1–1, 2018.

[91] D. Novosel, D. G. Hart, E. Udren, and M. M. Saha. Fault location using digital relay data. *Computer Applications in Power, IEEE*, 8(3):45–50, 1995.

[92] R. Nuqui. *State estimation and voltage security monitoring using synchronized phasor measurements*. PhD thesis, Citeseer, 2001.

[93] C. Onal and H. Kirrmann. Security improvements for IEEE 1588 Annex K: Implementation and comparison of authentication codes. In *Precision Clock Synchronization for Measurement Control and Communication (ISPCS), 2012 International IEEE Symposium on*, pages 1–6. IEEE, 2012.

[94] C. Outra, S. Zimath, H. Rachade, and L. de Oliveira. Substation time synchronization in today and future architectures. In *Developments in Power System Protection (DPSP 2014), 12th IET International Conference on*, pages 1–6. IET, 2014.

[95] A. Pai. *Energy function analysis for power system stability*. Springer Science & Business Media, 2012.

[96] M. Patel, S. Aivaliotis, E. Ellen, et al. Real-time application of synchrophasors for improving reliability. *NERC Report, Oct*, 2010.

[97] Y. Pathan, A. Dalvi, A. Pillai, D. Patil, and D. Reed. Analysis of selective packet delay attack on IEEE 1588 Precision Time Protocol. *University of Colorado at Boulder*, 2014.

[98] A. G. Phadke and J. S. Thorp. *Synchronized phasor measurements and their applications.* Springer Science & Business Media, 2008.

[99] F. Ramos, J. L. Gutiérrez-Rivas, J. López-Jiménez, B. Caracuel, and J. Díaz. Accurate timing networks for dependable smart grid applications. *IEEE Transactions on Industrial Informatics*, 14(5):2076–2084, 2018.

[100] I. Saha and D. Mukhopadhyay. Quantitative Analysis of a Probabilistic Non-repudiation Protocol through Model Checking. In *Information Systems Security*, pages 292–300. Springer, 2009.

[101] E. Schweitzer, D. Whitehead, S. Achanta, and V. Skendzic. Implementing Robust Time Solutions for Modern Power Systems. `https://www.selinc.com/WorkArea/DownloadAsset.aspx?id=99360`, 2012.

[102] U. Serizawa, M. Myoujin, K. Kitamura, N. Sugaya, M. Hori, A. Takeuchi, I. Shuto, and M. Inukai. Wide-area current differential backup protection employing broadband communications and time transfer systems. *Power Delivery, IEEE Transactions on*, 13(4):1046–1052, 1998.

[103] V. Shmatikov. Probabilistic Analysis of Anonymity. In *Proc. 15th IEEE Computer Security Foundations Workshop (CSFW'02)*, pages 119–128. IEEE Computer Society Press, June 2002.

[104] W. Stallings. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Addison-Wesley Longman Publishing Co., Inc., 1998.

[105] I. Standard. 200-04–IRIG Serial Time Code Formats–Sept04, Timing Committee, Telecommunications and Timing Group, Range Commanders Council. *US Army White Sands Missile Range, NM*, 2004.

[106] I. Standard. Network Engineering Guideline for Communication Networks and Systems in Substations. Technical report, IEC 61850-90-4, 2013.

[107] S. E. Stanton, C. Slivinsky, K. Martin, and J. Nordstrom. Application of phasor measurements and partial energy analysis in stabilizing large disturbances. *Power Systems, IEEE Transactions on*, 10(1):297–306, 1995.

[108] F. Steinhauser, C. Riesch, and M. Rudigier. IEEE 1588 for time synchronization of devices in the electric power industry. In *Precision Clock Synchronization for Measurement Control and Communication (ISPCS), 2010 International IEEE Symposium on. IEEE*, pages 1–6, 2010.

[109] P. Subcommittee. IEEE reliability test system. *IEEE Transactions on Power Apparatus and Systems*, 1979.

[110] Symmetriccom. Profile for the Use of the Precision Time Protocol in Power Systems. `http://www.arsitec.com.br/whitepaper.pdf`, 2013.

[111] Symmetricom. Mitigating GPS Vulnerabilities and protecting power utility network timing. `https://www.aventasinc.com/whitepapers/WP_Power_Utilities.pdf`, 2013.

[112] Tekron. Best Practices in Substation Time Synchronization - Isolation. `http://tekron.com/sites/default/files/time_sync_white_paper.pdf`, 2014.

[113] Terrorism Research & Analysis Consortium. Attacks on Electrical Grids. http://www.trackingterrorism.org/article/attacks-electrical-grids, 2013.

[114] V. Terzija, G. Valverde, D. Cai, P. Regulski, V. Madani, J. Fitch, S. Skok, M. M. Begovic, and A. Phadke. Wide-area monitoring, protection, and control of future electric power networks. *Proceedings of the IEEE*, 99(1):80–93, 2011.

[115] J.-C. Tournier and O. Goerlitz. Strategies to secure the IEEE 1588 protocol in digital substation automation. In *Critical Infrastructures, 2009. CRIS 2009. Fourth International Conference on*, pages 1–8. IEEE, 2009.

[116] A. Treytl, G. Gaderer, B. Hirschler, and R. Cohen. Traps and pitfalls in secure clock synchronization. In *Precision Clock Synchronization for Measurement, Control and Communication, 2007. ISPCS 2007. IEEE International Symposium on*, pages 18–24. IEEE, 2007.

[117] A. Treytl and B. Hirschler. Security flaws and workarounds for IEEE 1588 (transparent) clocks. In *IEEE International Symposium on Precision Clock Synchronization*, pages 1–6. Citeseer, 2009.

[118] A. Treytl and B. Hirschler. Securing IEEE 1588 by IPsec tunnels-an analysis. In *Precision Clock Synchronization for Measurement Control and Communication (ISPCS), 2010 International IEEE Symposium on*, pages 83–90. IEEE, 2010.

[119] J. Tsang and K. Beznosov. A security analysis of the precise time protocol (short paper). In *Information and Communications Security*, pages 50–59. Springer, 2006.

[120] M. Ullmann and M. Vogeler. Delay attacks - Implication on NTP and PTP time synchronization. In *Precision Clock Synchronization for Measurement, Control*

and Communication, 2009. ISPCS 2009. International Symposium on, pages 1–6. IEEE, 2009.

[121] A. Varga et al. The omnet++ discrete event simulation system. In *ESM'2001*, volume 9, page 65. sn, 2001.

[122] W. Wallner. Simulation of the IEEE 1588 Precision Time Protocol in OMNeT++. *arXiv preprint arXiv:1609.06771*, 2016.

[123] F. L. Walls. Precision oscillators: dependence of frequency on temperature, humidity and pressure. 1992.

[124] J. Wei and D. Kundur. Goalie: goal-seeking obstacle and collision evasion for resilient multicast routing in smart grid. *IEEE Transactions on Smart Grid*, 7(2):567–579, 2016.

[125] Q. Yang, D. An, and W. Yu. On time desynchronization attack against IEEE 1588 protocol in power grid systems. In *Energytech, 2013 IEEE*, pages 1–5. IEEE, 2013.

[126] J. Zhang and D. Chen. On the application of Phasor Measurement Units to power system stability monitoring and analysis. In *Power and Energy Conference at Illinois (PECI), 2012 IEEE*, pages 1–6. IEEE, 2012.

[127] R. Zhang, Y. Xu, Z. Y. Dong, and K. P. Wong. Post-disturbance transient stability assessment of power systems by a self-adaptive intelligent system. *IET Generation, Transmission & Distribution*, 9(3):296–305, 2015.

[128] Z. Zhang, S. Gong, A. D. Dimitrovski, and H. Li. Time synchronization attack in smart grid: Impact and analysis. *Smart Grid, IEEE Transactions on*, 4(1):87–98, 2013.

[129] K. Zhu, M. Chenine, and L. Nordstrom. ICT architecture impact on wide area monitoring and control systems' reliability. *IEEE transactions on power delivery*, 26(4):2801–2808, 2011.

[130] K. Zhu, M. Chenine, L. Nordstrom, S. Holmstrom, and G. Ericsson. Design requirements of wide-area damping systems—Using empirical data from a utility IP network. *IEEE Transactions on Smart Grid*, 5(2):829–838, 2014.

[131] Zhu, J. and Lubkeman, D. and Girgis, A. . Automated fault location and diagnosis on electric power distribution feeders. *Power Delivery, IEEE Transactions on*, 12(2):801–809, 1997.

[132] G. Ziegler. *Numerical differential protection: principles and applications.* John Wiley & Sons, 2012.

Appendix A

New SNMP Objects

```
PTPClockTimestamp ::= TEXTUAL CONVENTION
STATUS          current
DESCRIPTION     ”Timestamp in Sync message:
seconds (6 byte integer)
followed by nanoseconds
(4 byte integer).”
SYNTAX          OCTET STRING(SIZE(10))

PTPSyncSeqNumber ::= TEXTUAL CONVENTION
STATUS          current
DESCRIPTION     ”The sequence number of
received Sync message.”
SYNTAX          Unsigned32(0..65535)

PTPTimestampHash ::= TEXTUAL CONVENTION
STATUS          current
```

```
DESCRIPTION     ”Hash of the timestamp
values.”
SYNTAX          OCTET STRING(SIZE(32))

PTPSeqBitmask   ::= TEXTUAL CONVENTION
STATUS          current
DESCRIPTION     ”The bitmask of received
sequence numbers used in
the hash (up to 16).”
SYNTAX          BITS {
        bit0(0),
        bit1(1),
        bit2(2),
        bit3(3),
        bit4(4),
        bit5(5),
        bit6(6),
        bit7(7),
        bit8(8),
        bit9(9),
        bit10(10),
        bit11(11),
        bit12(12),
        bit13(13),
        bit14(14),
        bit15(15)}
```

```
ptpTimestampTable  OBJECT TYPE
SYNTAX           SEQUENCE OF ptpTimestampEntry
MAX ACCESS       not accessible
STATUS           current
DESCRIPTION      ”Table containing GMC’s
timestamp information.”
::={ieeeC37238Objects 8}

ptpTimestampEntry OBJECT TYPE
SYNTAX           PTPTimestampEntry
MAX ACCESS       not accessible
STATUS           current
DESCRIPTION      ”Information about sequence
number and timestamp of a
single Sync message.”
INDEX            {ptpTimestampSeqNum}
::= {ptpTimestampTable 1}

PTPTimestampEntry          ::= SEQUENCE{
          ptpTimestampSeqNum     PTPSyncSeqNumber,
          ptpTimestampValue      PTPClockTimestamp}

ptpTimestampSeqNum OBJECT TYPE
SYNTAX             PTPSyncSeqNumber
MAX ACCESS         read only
```

```
STATUS          current
DESCRIPTION     ”Sync message
sequence number.”
::= {ptpTimestampEntry 1}

ptpTimestampValue OBJECT TYPE
SYNTAX             PTPClockTimestamp
MAX ACCESS         read only
STATUS             current
DESCRIPTION        ”Sync message timestamp.”
::= {ptpTimestampEntry 2}

ptpTSHashTable      OBJECT TYPE
SYNTAX          SEQUENCE OF ptpTSHashEntry
MAX ACCESS      not accessible
STATUS          current
DESCRIPTION     ”Table containing PTP slave's
hashes for received timestamps.”
::= {ieeeC37238Objects 9}

ptpTSHashEntry     OBJECT TYPE
SYNTAX          PTPTSHashEntry
MAX ACCESS      not accessible
STATUS          current
DESCRIPTION     ”Information about the
received timestamps.”
```

```
INDEX          {ptpTSHashFirstSeqNum}
::= {ptpTSHashTable 1}

PTPTSHashEntry          ::= SEQUENCE{
        ptpTSHashFirstSeqNum        PTPSyncSeqNumber,
        ptpTSHashBitmask      PTPSeqBitmask,
        ptpTSHashValue        PTPTimestampHash}

ptpTSHashFirstSeqNum     OBJECT TYPE
SYNTAX           PTPSyncSeqNumber
MAX ACCESS       read only
STATUS           current
DESCRIPTION      "Sequence Number of
first message in block."
::= {ptpTSHashEntry 1}

ptpTSHashBitmask   OBJECT TYPE
SYNTAX           PTPSeqBitmask
MAX ACCESS       read only
STATUS           current
DESCRIPTION      "Bit mask of the sequence
numbers in block."
::= {ptpTSHashEntry 2}

ptpTSHashValue   OBJECT TYPE
SYNTAX           PTPTimestampHash
```

```
MAX ACCESS       read only
STATUS           current
DESCRIPTION      ”Hash of reported
timestamps.”
::= {ptpTSHashEntry 3}
```

Appendix B

PTP Extension Validation

B.0.1 Verification Environment

To verify the soundness of our approach, we followed a formal modeling and verification approach. We built a model of the extended PTP protocol using timed automata, then simulated the model, and formally verified most relevant properties using model checking. Timed Automata are finite state machines augmented with clocks and time constraints. A system modeled using Timed Automata consists of a network of processes. A process is a timed automaton where states are called locations. Transitions between locations define how the system behaves. The simulation step consists of running the system interactively to check that it works as intended [25]. However, this step is not conclusive as it does not allow to cover all possible behaviors of the system. Then we can a model checker to check some reachability, liveness and safety properties. Model-checking is basically an exhaustive search that covers all possible dynamic behaviors of the system [25].

The modeling and verification platform we used is the publicly available tool, UPPAAL [70]. UPPAAL is a tool box for modeling and verification (via automatic model-checking) of real-time systems [25]. A system in UPPAAL is a collection of processes, each represented by an automaton. Transitions, controlled by conditions (guards), are used to change state. Control on transitions is performed by guards and synchronization. The synchronization mechanism in UPPAAL is a hand-shaking synchronization: two processes take a transition at the same time, one will have an a! and the other an a?, with a being the synchronization channel [25]. With transitions, updates on variables or reset of clocks are possible.

B.0.2 PTP Model

The proposed solution model has been implemented in UPPAAL as a network of timed automata that synchronize over communication channels. The model is composed of the following parts:

1. Declarations.
2. The System Clock model.
3. The Slave model.
4. The Master model.
5. The Network/Attacker model.
6. The NTR model.
7. Systems declarations.

We will detail each of the mentioned model parts.

Declarations The declaration part is where all global variables, constants, clocks and channels are listed. Constants are used to configure parameters of the model such as the number of slaves, the path delay interval, etc. Channels are the means

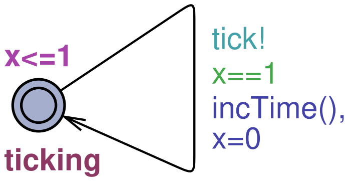

Figure B.1: The system clock component.

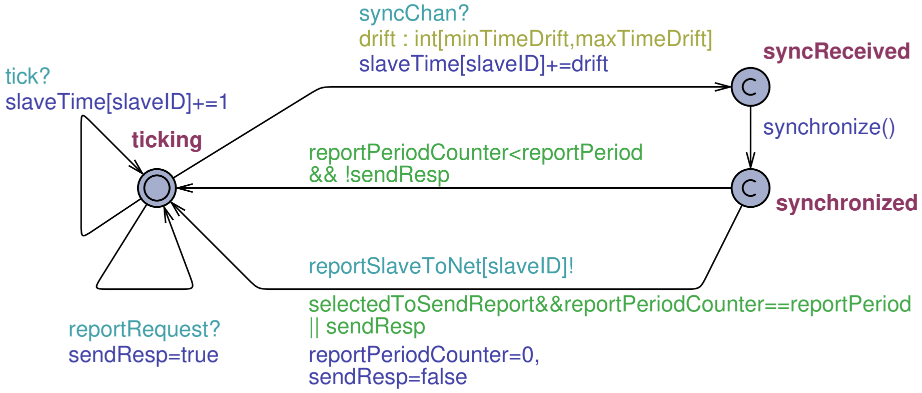
syncChan?
drift : int[minTimeDrift,maxTimeDrift]
slaveTime[slaveID]+=drift
syncReceived
tick?
slaveTime[slaveID]+=1
ticking
synchronize()
reportPeriodCounter<reportPeriod
&& !sendResp
synchronized
reportSlaveToNet[slaveID]!
selectedToSendReport&&reportPeriodCounter==reportPeriod
|| sendResp
reportRequest?
sendResp=true
reportPeriodCounter=0,
sendResp=false

Figure B.2: The slave component.

of synchronization between the different components of the system. Variables store quantities visible to the different processes.

The System Clock model The system clock model presented in Fig. B.1 is at the heart of our model. The system clock ticks to measure absolute time, and other system components that are time aware tick with it. Thus, the ticks are broadcast via the channel 'tick', and the other components (including slaves) synchronize via this channel. However, each of these slaves may drift at a specific pace as we will see later. In Fig. B.1, the initial state is depicted with a double circle. The cyan expression is the synchronization channel name. The green expressions are guards that have to be true in order for the corresponding transition to be enabled. The blue expressions are assignments to variables and execution of pre-defined functions, that are executed when the corresponding transition is taken.

The Slave model The slave model presented in Fig. B.2 represents the slave behavior in our system. This model is used as a template to instantiate multiple slaves in the system. At the instantiation of a slave, parameters representing the slave identitiy (slaveID), assignment to send reports to the NTR (selectedToSendReport), frequency of sending reports(reportPeriod), etc are specified.

The slave model updates its local timing over the 'tick' channel, receives the master *Sync* messages over the 'syncChan' and transits between states to synchronize its clock. When in a committed state, state containing the letter 'C', the only possible transition is always the one going out of the committed state. The committed state has to be left immediately as opposed to other normal states. Using such states allows the representation of actions that are accomplished before the elapse of any time as in the case of synchronizing the slave clock using *Sync* messages. Upon synchronization, and based on local guards, the model decides the next transition to take place. Such transitions include issuing a *Report* to the NTR upon fulfillment of predefined frequency, or respond to NTR report request processed in earlier state.

The Master model The master model is presented in Fig. B.3. The master ticks with the clock, and broadcasts a *Sync* message over the network at a constant pace specified by the syncPeriod constant. The timestamp associated with this message is saved in a global variable that may be accessed by all devices receiving this message. Moreover, this timestamp can be attacked when we need to model an attack on the master clock.

The Network model To model the asynchronous communication that happens over the network, and the impact of attackers on it, we introduce different components for different network connections. The network connection between the master and each slave, as well as the connection between the NTR and each slave are modeled in

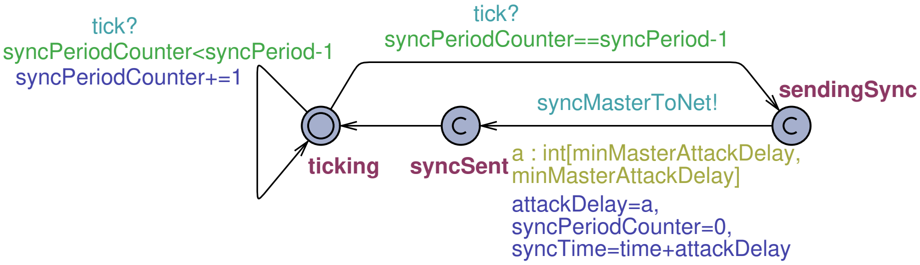

Figure B.3: The master component.

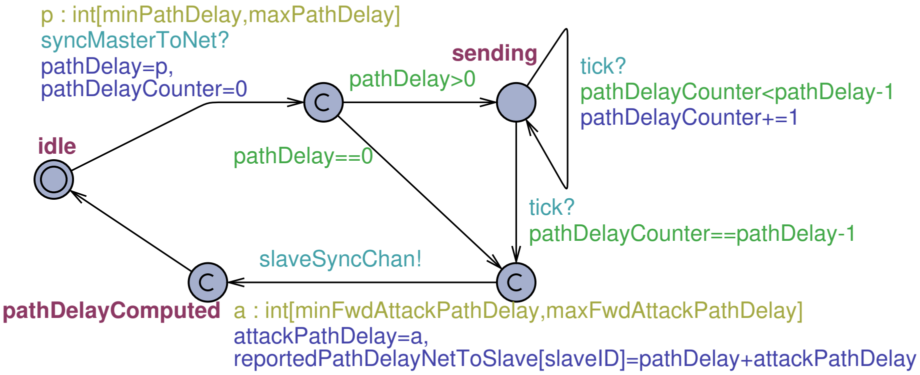
p : int[minPathDelay,maxPathDelay]
syncMasterToNet?
pathDelay=p,
pathDelayCounter=0
sending
pathDelay>0
tick?
pathDelayCounter<pathDelay-1
pathDelayCounter+=1
idle
pathDelay==0
tick?
pathDelayCounter==pathDelay-1
slaveSyncChan!
pathDelayComputed
a : int[minFwdAttackPathDelay,maxFwdAttackPathDelay]
attackPathDelay=a,
reportedPathDelayNetToSlave[slaveID]=pathDelay+attackPathDelay

Figure B.4: The master to slave network component.

a similar manner, yet using distinct components.

The master to slave network model is presented in Fig. B.4. It resembles the way the network handles *Sync* messages broadcast by the master. This component takes the ID of the slave it is connecting to (slaveID), and the channel to synchronize with the Master and receive the *Sync* message as input. The model selects a non deterministic path delay from a pre-defined interval, and accordingly delivers the *Sync* message after the elapse of this time over the network while in the state 'sending'. Once the system is built, an instance of this component is created for each slave in the network. This allows to have different path delays in the delivery of *Sync* messages to different slaves. Morover, using this network model, the attack delay can be performed through the attackPathDelay variable which is non deterministically selected from a predefined interval.

The network connecting the slaves to the NTR is identical to that in Fig. B.4,

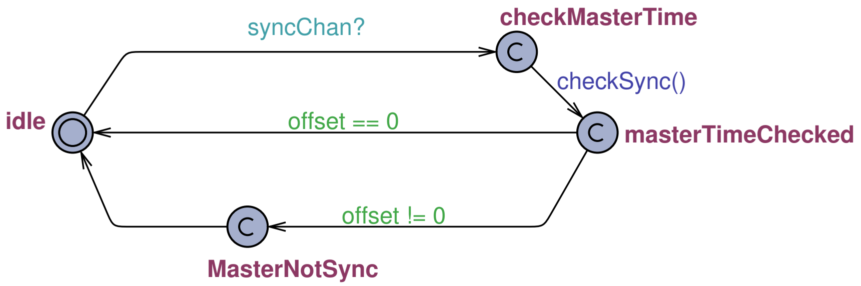

Figure B.5: NTR model - master component.

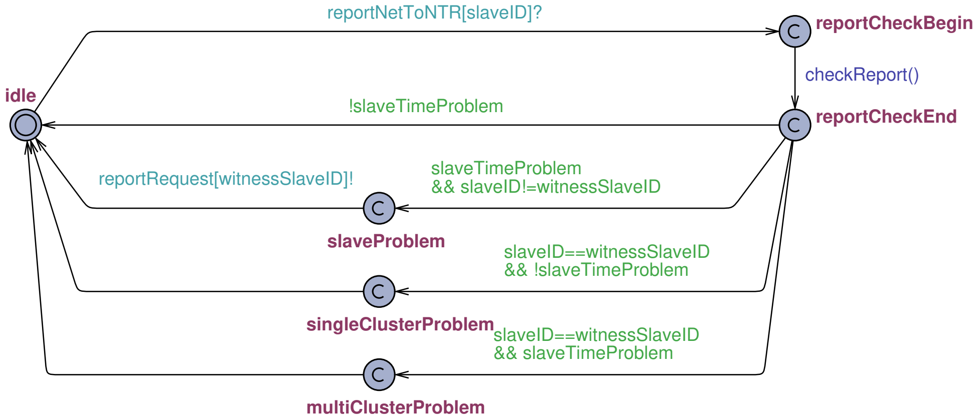
reportNetToNTR[slaveID]?
reportCheckBegin
checkReport()
idle
!slaveTimeProblem
reportCheckEnd
reportRequest[witnessSlaveID]!
slaveTimeProblem && slaveID!=witnessSlaveID
slaveProblem
slaveID==witnessSlaveID && !slaveTimeProblem
singleClusterProblem
slaveID==witnessSlaveID && slaveTimeProblem
multiClusterProblem

Figure B.6: NTR model - slave component.

however the synchronization channel is used to synchronize the delivery of messages from the slaves to the NTR.

The NTR model The NTR is modeled in two parts. The part that monitors the master timing through *Sync* messages is shown in Fig. B.5, while the second component receives and processes reports from slaves and is shown in Fig. B.6.

The NTR component assigned to the master functionality receives *Sync* messages sent by the master over the 'syncChan'. Using the timestamp and the path delay associated with this message, it computes an offset in the 'checkSync()' method. Based on the computed offset, the model takes the suitable transition. This component goes into the state 'MasterNotSync' when the offset computation indicates an error in the master time.

On the other hand, the NTR-slave component periodically receives reports from selected slaves in the network over the associated network synchronization channel. Each Time the NTR receives a report, it checks it for anomalies in the ‘checkReport() procedure. If an anomaly is detected, the NTR requests a report from a witness slave to check if the attack is affecting more than one cluster or not. This request is sent over the reportRequest[ ]! channel, associated response is received later through the network channel, and processed in a similar way to that of the report. The NTR identifies this as a response if it is sent by the ‘witnessSlave’. Based on the outcome of this communication, the NTR-slave component transits into different states each indicating the presence of a timing problem at different levels. Those states, as can be seen in Fig. B.6, are ‘slaveProblem’, ‘singleClusterProblem’, and ‘multiClusterProblem’.

System Declarations In this part of the model, the configuration of the system is set by creating instances of all participating components and connecting them accordingly.

B.0.3 Verified Properties

We verified two types of properties on the built model.

1. A[] p: this property validates that the proposition ‘p’ always holds for all paths in the system.

2. p → q: this property validates that whenever propositions ‘p’ holds, proposition ‘q’ will eventually hold.