Network and System Management using IEC 62351-7 in IEC 61850 Substations: Design and Implementation

Chantale Robillard

A Thesis

in

The Department

of

Concordia Institute for Information Systems Engineering (CIISE)

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Information Systems Security) at

Concordia University

Montréal, Québec, Canada

December 2018

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Chantale Robillard**

Entitled: **Network and System Management using IEC 62351-7 in IEC 61850 Substations: Design and Implementation**

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

Master of Applied Science (Information Systems Security)

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

Signed by the Final Examining Committee:

_____ Chair
Dr. Jun Yan

_____ External Examiner
Dr. Yan Liu

_____ Examiner
Dr. Chadi Assi

_____ Supervisor
Dr. Mourad Debbabi

_____ Co-supervisor
Dr. Aiman Hanna

Approved by _____
Abdessamad Ben Hamza, Director
Concordia Institute for Information Systems Engineering (CIISE)

_____ 2019

_____
Amir Asif, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Network and System Management using IEC 62351-7 in IEC 61850 Substations: Design and Implementation

Chantale Robillard

Substations are a prime target for threat agents aiming to disrupt the power grid's operation. With the advent of the smart grid, the power infrastructure is increasingly being coupled with an Information and Communication Technologies (ICT) infrastructure needed to manage it, exposing it to potential cyberattacks. In order to secure the smart grid, the IEC 62351 specifies how to provide cybersecurity to such an environment. Among its specifications, IEC 62351-7 states to use Network and System Management (NSM) to monitor and manage the operation of power systems. In this research, we aim to design, implement, and study NSM in a digital substation as per the specifications of IEC 62351-7. The substation is one that conforms to the IEC 61850 standard, which defines how to design a substation leveraging ICT. Our contributions are as follows. We contribute to the design and implementation of NSM in a smart grid security co-simulation testbed. We design a methodology to elaborate cyberattacks targeting IEC 61850 substations specifically. We elaborate detection algorithms that leverage the NSM Data Objects (NSM DOs) of IEC 62351-7 to detect the attacks designed using our method. We validate these experimentally using our testbed. From this work, we can provide an initial assessment of NSM within the context of digital substations.

Acknowledgments

I would like to thank my supervisors Dr. Mourad Debbabi and Dr. Aiman Hanna for giving me the opportunity to work on this master's degree. I have learned much about cybersecurity and academic research from them and am forever grateful. I would also like to thank Dr. Marthe Kassouf from the Hydro-Qubec Research Institute for her guidance during my research on topics such as the security monitoring of digital substations. as well as the design and implementation of NSM and IEC 62351-7.

I also want to thank everyone that helped me while working on this thesis. This includes everyone at the cybersecurity lab. Special thanks go to Mark Karanfil, Abdullah Albarakati, and Dr. Rachid Hadjidj for their help with building and using NSM in the co-simulation testbed. This work would not have been complete without it. I would like to express my gratitude to Dr. Alf Zugenmaier, as our initial discussions inspired me in elaborating the methodology I propose in this thesis, and to Suo Tan, for providing an easy-to-use template to write the thesis itself.

Finally, I would like to thank my family and my partner for their support while working on my degree. I especially wish to thank my parents, who have always encouraged me to study my passion in computer science.

Contents

List of Figures

List of Tables

List of Acronyms

ACSE Association Control Service Element

ACSI Abstract Communication Service Interface

AES Advanced Encryption Standard

AMI Advanced Metering Infrastructure

APDU Application Protocol Data Unit

ARP Address Resolution Protocol

ASDU Application Service Data Unit

ASN.1 Abstract Syntax Notation One

BER Basic Encoding Rules

CAM Content Addressable Memory

CB Circuit Breaker

CBC Cipher Block Chaining

CFS Correlation based Feature Selection

CNN Convolution Neural Network

CNN-LSTM Convolutional Neural Network-Long Short-Term Memory

CPN Colored Petri Net

CPU Central Processing Unit

CRL Certificate Revocation List

CT Current Transformer

DA Data Attribute

DCU Data Concentrator Unit

DGM Distribution Grid Management

DER Distributed Energy Resource

DES Data Encryption Standard

DES Discrete Event Simulation

DMS Distribution Management System

DNP3 Distributed Network Protocol

DNP3-SA DNP3 Secure Authentication

DNS Domain Name System

DO Data Object

DoS Denial-of-Service

DDoS Distributed Denial-of-Service

DPI Deep Packet Inspection

DHE Diffie-Hellman Ephemeral

DRDoS Distributed Reflected Denial-of-Service

DHCP Dynamic Host Configuration Protocol

EM Expectation Maximization

EOL end-of-life

EPRI Electric Power Research Institute

FTP File Transfer Protocol

GOOSE Generic Object Oriented Substation Event

GPS Global Positioning System

GRU Gated Recurrent Units

GSE Generic Substation Event

GSSE Generic Substation State Event

HMAC Hashed Message Authentication Code

HSR High-availability Seamless Redundancy

HTTP Hypertext Transfer Protocol

ICMP Internet Control Message Protocol

ICS Industrial Control System

ICT Information and Communication Technologies

ID Intrusion Detection

IDS Intrusion Detection System

IEC International Electrotechnical Commission

IED Intelligent Electronic Device

IEEE Institute of Electrical and Electronics Engineers

IoT Internet of Things

IP Internet Protocol

I-RNN Identity-Recurrent Neural Network

ISEAGE Internet-Scale Event and Attack Generation Environment

ISO International Organization for Standardization

IT Information Technology

LAN Local Area Network

LD Logical Device

LN Logical Node

LOF Local Outlier Factor

LPT Large Power Transformer

LSTM Long Short-Term Memory

MAC Media Access Control

MAC Message Authentication Code

MBR Master Boot Record

MD5 Message Digest 5

MIB Management Information Base

MitM Man-in-the-Middle

MMS Manufacturing Message Specification

MU Merging Unit

NSTB National SCADA Testbed

NERC North American Electric Reliability Corporation

NESCOR National Electric Sector Cybersecurity Organization Resource

NIST National Institute of Standards and Technology

NMS Network Management Station

NSM Network and System Management

NSM DO NSM Data Object

OCSP Online Certificate Status Protocol

OID Object Identifier

OS Operating System

OSI Open Systems Interconnection

OT Operations Technology

OWASP Open Web Application Security Project

P2P peer-to-peer

PMU Phasor Measurement Unit

P&C Protection and Control

PIDC Protocol Independent Detection and Classification

PDC Phasor Data Concentrator

PDU Protocol Data Unit

PDoS Permanent Denial-of-Service

PIM Protocol Independent Multicast

PLC Programmable Logic Controller

PN Petri Net

PRP Parallel Redundancy Protocol

PTP Precision Time Protocol

PV photovoltaic

RA Registration Authority

RBAC Role-based Access Control

RBD Reliability Block Diagram

RC4 Rivest Cipher 4

RCD Rank Correlation based Detection

RNN Recurrent Neural Network

R-GOOSE Routable GOOSE

RSA Rivest-Shamir-Adleman

RSTP Rapid Spanning Tree Protocol

RTDS Real Time Digital Simulator

RTU Remote Terminal Unit

SCADA Supervisory Control and Data Acquisition

SCC Strongly Connected Components

SCD Substation Configuration Description

SCSM Specific Communication Service Mapping

SHA Secure Hash Algorithm

SNE Substation Network Explorer

SNMP Simple Network Management Protocol

SNTP Simple Network Time Protocol

SPARKS Smart Grid Protection Against Cyber Attacks

SQL Structured Query Language

SSH Secure Shell

SSL Secure Sockets Layer

SMB Server Message Block

SMV Sampled Measured Values

SV Sampled Values

SVM Support Vector Machine

TAL time allowed to live

TC Technical Committee

TCP Transmission Control Protocol

TLS Transport Layer Security

UDP User Datagram Protocol

USB Universal Serial Bus

UML Unified Modeling Language

US DOE United States Department of Energy

USM User-based Security Module

VLAN Virtual LAN

VM Virtual Machine

VMD Virtual Manufacturing Device

VT Voltage Transformer

WAN Wide Area Network

WAMPAC Wide Area Monitoring, Protection, and Control

Chapter 1

Introduction

1.1 Motivations

The power grid is part of a nation's critical infrastructure [1]. As such, utilities in North America must comply with the standards set by the North American Electric Reliability Corporation (NERC) that aim to ensure the reliability and security of the power grid across the continent [2]. As part of improving the availability and overall quality of electrical power, utilities transition to the smart grid. This is done by adding a communication infrastructure to the existing power infrastructure. In particular, substations, which are responsible for controlling power en-route to clients [3], are increasingly being converted to digital substations that incorporate Intelligent Electronic Devices (IEDs) and Ethernet networks. The IEC 61850 standard, [4] published by the International Electrotechnical Commission (IEC), specifies how to design substations in this manner. This new dependency on Information Technology (IT) provides many benefits, but also provides new attack surfaces [5] for threat agents that want to disrupt power grid operations. Indeed, the newly added devices are vulnerable to cyber attacks. Substations are a prime target, because they are usually unmanned [6] and spread across large distances, requiring lengthy travel by the utility when maintenance is to be done. They also provide power to entire regions, making the impact of an attack against them significant. In order to address this issue, utilities must include security measures to protect their assets.

The IEC 62351 standard aims to address exactly this problem. Previously, cybersecurity was provided in power systems through security by obscurity [1]. In other words, it was assumed that

potential attackers could not possibly have the motivation and the specialized knowledge required to carry out a successful cyberattack. Nowadays, security based on secrecy is considered ineffective [1], requiring utilities to turn to truly effective security measures. The IEC 62351 standard's goal is to provide recommendations to enhance the cybersecurity of power systems, including substations, by introducing cryptographic measures, monitoring capabilities, user management, and more [7]. Due to the differences between traditional IT networks and the Operations Technology (OT) networks as used in the power grid, existing security measures typically used for IT networks must be adapted to respond to the threats targeting power networks specifically. IEC 62351 does so by applying security measures while considering the context of power systems, including the specific communication protocols previously designed by the IEC and adapted to this environment.

Among its many sections, part 7 of IEC 62351, referred to as IEC 62351-7, recommends the use of NSM [8]. NSM enables monitoring and management of the devices found in a network from a central management system, providing operators a global view of their network and an effective way to detect potential problems. It is used in many existing IT networks to detect and mitigate issues such as performance problems, equipment failures and cyberattacks. As such, NSM brings numerous benefits to an environment where reliability is high priority. To bridge the gap between existing monitoring solutions and power systems, IEC 62351-7 provides the NSM Data Objects (NSM DOs), which represent pieces of information about the health of a device or network. These NSM DOs are used to monitor power systems specifically. Because the latest edition of this standard was only published in 2017 [8], it has not yet been widely adopted by manufacturers or been the subject of much research. It is therefore essential to study the effectiveness of NSM in more depth by implementing the recommendations of IEC 62351-7 and testing the capabilities of NSM in relevant scenarios.

1.2 Contributions

The objective of this thesis is to design and implement NSM as per IEC 62351-7 in the context of an IEC 61850 substation, and study the capabilities of NSM in addressing threats against the substation. The contributions of this work are as follows:

- We contribute to the design and implementation of the specifications in IEC 62351-7 within a smart grid security testbed monitored by an NSM solution;
- We design methodologies to elaborate cyberattacks specific to IEC 61850 substations and that IEC 62351-7 is meant to address;
- We elaborate detection algorithms that leverage the capabilities of NSM to defend against cyberattacks and validate them using the testbed;
- We provide an overall assessment of the capabilities of NSM and provide recommendations to enhance them.

1.3 Thesis Organization

Chapter 2 provides background information necessary to understand the problem at hand and the remaining chapters. Chapter 3 provides a literature review of existing work on attacks against IEC 61850 substations, IEC 62351 and the use of NSM as per IEC 62351-7. Chapter 4 describes the joint implementation of the smart grid security testbed with NSM. Chapter 5 discusses and prioritizes classes of cyber attacks against the substation, defines a methodology to design new cyber attacks, and discusses the use of NSM to address them. Experiments are carried out to validate the defense mechanisms proposed and recommendations are provided to enhance NSM. Chapter 6 concludes the thesis and highlights future work.

Chapter 2

Background

This chapter gives an overview of concepts used in the rest of this thesis.

2.1 Cybersecurity Goals and Cyberattacks

The three main goals in cybersecurity are usually availability, integrity, and confidentiality. In addition to these, IEC 62351 also considers non-repudiation, authentication and authorization among its goals [7]. A definition of each of these terms and their related cyberattacks follows.

2.1.1 Authentication

Definition Corroborating parties' claimed identities and the source of messages [9].

Authentication can be divided in two aspects. Entity authentication [9] is to verify that entities are truly who or what they claim to be. A common way to achieve this is through the use of keys, passwords, certificates, or biometrics that are unique to each entity. These ensure authentication as only the legitimate entity should know and be able to present the correct credentials. Message authentication [9], on the other hand, is to verify the source of some information. To authenticate data, Message Authentication Codes (MACs) and digital signatures, like Rivest-Shamir-Adleman (RSA), are used. These techniques allow one to produce a small value for the message that can only be computed by the legitimate holder of the key. Authentication cannot be achieved if the authenticator is not reliable, as is the case for fingerprint sensors that are fooled by fake fingerprints

[10]. It also relies heavily on the security of the credentials, since an attacker that obtains them can then impersonate another entity.

2.1.2 Authorization

Definition Restrict access to only legitimate entities [9].

Authorization is to only allow actions by entities that have the required permissions. In other words, the system must allow actions to be performed by entities that have the necessary role or rights to do them, as well as prevent actions by entities that do not have those rights. Authorization is a key requirement for the other security goals, as it distinguishes the real (authorized) entities from others. Note that authentication is a prerequisite for authorization.

2.1.3 Confidentiality

Definition Ensuring that information cannot be accessed by unauthorized entities [7].

Confidentiality means hiding information except from those meant to access it. Securing confidential data can be achieved by using encryption and distributing the decryption key only to authorized entities. Encryption algorithms are designed so that encrypted information appears as useless random bits to anyone without the decryption key, while also making it very easy to recover the information for key holders. A commonly used encryption algorithm is Advanced Encryption Standard (AES). Confidentiality is essential to protect not only data, but also credentials that must be kept secret to achieve the other security goals. Attacks on confidentiality include sniffing packets traveling over a network, stealing sensitive data from a file or database on an end device, guessing encryption keys, and gathering intelligence on a system through scanning or eavesdropping.

2.1.4 Integrity

Definition Ensuring that information cannot or modified by unauthorized entities [7].

Integrity means a system or piece of information can be trusted. Whereas confidentiality aims to prevent unauthorized reading of information, integrity aims to prevent unauthorized writing of information. This is often achieved by relying on hash functions. A hash function takes data of any length and converts it into a unique, small “hash” value of a specific length. Recalculating the hash

of a specific input should always result in the same hash: if it does not, this indicates the input has been altered and cannot be trusted. Hash functions of relevance to this thesis are Message Digest 5 (MD5) and Secure Hash Algorithm (SHA)-1. MD5 is now considered insecure. SHA-1 fares slightly better than MD5, but National Institute of Standards and Technology (NIST) recommends replacing it with the superior SHA-256 [11]. A MAC or digital signature (such as RSA) blend integrity and authentication, as they are essentially hashes that require a secret key to calculate. Attacks on integrity involve falsifying data in some fashion, such as modifying the contents of files or databases. In networks, attacks on integrity include resending previously sent packets (replay), injecting forged packets in the network, or modifying existing packets after setting up a Man-in-the-Middle (MitM) attack.

2.1.5 Availability

Definition Ensuring timely and authorized access to information.

Availability means to ensure access to the system whenever it is required by authorized parties [7]. A system with high availability is ready to be accessed any time it is needed by a legitimate user. The system must also complete its tasks in an acceptable amount of time, as dictated by performance requirements. Attacks against availability include Denial-of-Service (DoS) attacks and degradation-of-service attacks. The former aims to render a system unusable by legitimate parties [12]. The latter only partially reduces availability [13], such as by significantly slowing down a system. Because degradation-of-service is often considered a subtype [13] or a synonym [14] of denial-of-service, we use the acronym DoS[1] to refer to both of them in this work.

2.1.6 Non-repudiation

Definition Preventing parties from denying their past actions or claiming actions that did not actually occur [7].

Non-repudiation ensures that parties cannot hide or falsify records of their activity. This property is important for auditing after a security event to ensure that the log archives are trustworthy. Digital

[1] Often, the expression "DoS attack" is used to refer to a packet flooding attack. We do not use this definition as it does not encompass the many other possible DoS attacks such as Permanent Denial-of-Service (PDoS).

signatures can provide a degree of non-repudiation as well as integrity and authentication for specific messages, as digital signatures can only be computed by the legitimate holder of a private key that is not shared with other entities.

2.2 Smart Grid and Potential Threats

2.2.1 Overview of the Smart Grid

A power grid is the infrastructure used to deliver electricity to clients. It has three major subsystems: generation, transmission, and distribution [3]. Generators are the source of electricity. The transmission system increases the voltage of power coming from generators, as this reduces energy loss in transit, and carries electricity to the distribution system. There, the voltage is reduced to usable levels for customers before delivery [3]. In 2009, the United States Department of Energy (US DOE) stated that the power grids in use at the time approached end-of-life (EOL) [15]. Utilities are thus motivated to switch to smart grid technologies. The smart grid is an upgrade to existing power grids that aims to provide efficiency and reliability by integrating ICT [5]. It provides many advantages, such as automatically detecting and responding to events, integrating new generation and storage devices to enable distributed generation, and improving power quality [15]. Monitoring is done using a Supervisory Control and Data Acquisition (SCADA) system that allows operators to remotely control the grid from a control center [16]. Communications in the smart grid is done through a variety of protocols, namely Distributed Network Protocol (DNP3), IEC 60870-5, and IEC 61850 [16]. As part of the development of the smart grid, next-generation substations replace old electro-mechanical devices with IEDs that can perform protection, monitoring and control functions [4]. This change led to the publication of the IEC 61850 standard that defines how these IEDs should communicate in an interoperable manner even when they are designed by different manufacturers [4].

2.2.2 Threats to the Smart Grid

Power grids are potentially vulnerable to threats such as natural disasters, errors by utility employees, and deliberate sabotage of electrical equipment. In addition to these already existing

threats, the integration of IT technologies in power grids necessarily brings with it the potential for cyberattacks against these new devices, especially if the grid's communication network is exposed to the Internet [5]. Such attacks can even be carried out remotely. News headlines already highlight several examples of cyberattacks targeting Industrial Control Systems (ICS's). Stuxnet is a sophisticated malware that targets Programmable Logic Controllers (PLCs) within ICS's. It forces them to increase and decrease their motors' speeds to reach extreme values outside their normal range, causing equipment damage [17]. Stuxnet also hides its activity to evade detection. The malware infected about 100,000 devices worldwide (most of them in Iran) through removable drives, according to a 2011 report by Symantec [17]. More recently, Ukraine's power grid was targeted on two separate occasions in 2015 [18] and in 2016 [19]. The first incident was a coordinated cyberattack that caused a power outage for over 200,000 customers. Hackers remotely took control of operators' software (possibly using the BlackEnergy trojan), opened breakers through unauthorized commands, then executed the KillDisk malware to wipe systems and stall recovery efforts [18]. By contrast, the second attack was carried out using the malware CrashOverride. This malware has many capabilities including scanning networks to gather intelligence about them, spoofing ICS commands, conducting DoS attacks to block communications or shut down devices, and wipe devices altogether [19]. CrashOverride is among the first malware that can target the IEC 61850 protocol, among other ICS protocols [16]. These examples highlight the importance of securing the smart grid and the IEC 61850 substation, given that there already exist sophisticated threats specifically designed for these environments.

2.3 IEC 61850: Standard for the Digital Substation

The IEC 61850 standard, titled "Communication networks and systems for power utility automation", defines the overall architecture and the communication protocols to be used between IEDs in digital substations [4]. The intent of the standard is to enable interoperability between IEDs from different manufacturers by defining common communication protocols and information models to be supported by all of them.

2.3.1 Substation Architecture

Primary Equipment in Substation

The purpose of a substation is to transform the voltage and current of the energy flowing on incoming power lines [3]. An example of a single-line diagram showing the primary equipment of a distribution substation[2] from IEC 61850-5 [20] is shown in Figure 2.1. The D2-1 substation, as it is referred to, contains two transformer bays (E01 and E02) and six feeder bays (K02 to K07, though some are not fully drawn) as well as one switching bay (K01). Incoming power from multiple transmission lines first passes through the transformer bays, where its voltage and current are modified by the transformers (drawn using two circles). Power then flows to the feeder bays by using the bus. In each bay, Current Transformers (CTs) (labeled I_{E01} to I_{K07}) and Voltage Transformers (VTs) (labeled U_{E01} to U_{K07}) are used to obtain measurements of the power within the bay. Note that in the case of the transformer bays, measurements are obtained from both sides of the transformer to monitor its functionality. Additionally, Circuit Breakers (CBs) (drawn as rectangles) found in each bay are used to stop the flow of electricity when required. This can be done to transfer loads, isolate parts of the substation, or mitigate faults as detected by relays monitoring the values from the CTs and VTs [3].

Communication Network and Layers

In addition to the primary equipment, the digital substation also includes IEDs interconnected using a communication network. As an example, a communication network added to the D2-1 distribution substation is found in the work of Zhang *et al.* [21] and shown in Figure 2.2. There is one Merging Unit (MU) at every location monitored by CTs and VTs, one breaker IED for every CB, and one Protection and Control (P&C) IED per bay. The MUs in a bay transmit measurements from the CTs and VTs to the P&C IED using the Sampled Values (SV) protocol [21]. The P&C IED monitors the measurements. If it detects a fault according to its configuration, it sends a trip signal using the Generic Object Oriented Substation Event (GOOSE) protocol to the relevant breaker IEDs, which then trips the CB [21]. The GOOSE and SV protocols are discussed later in Section 2.3.3.

[2]Note that IEC 61850 also considers transmission substations.

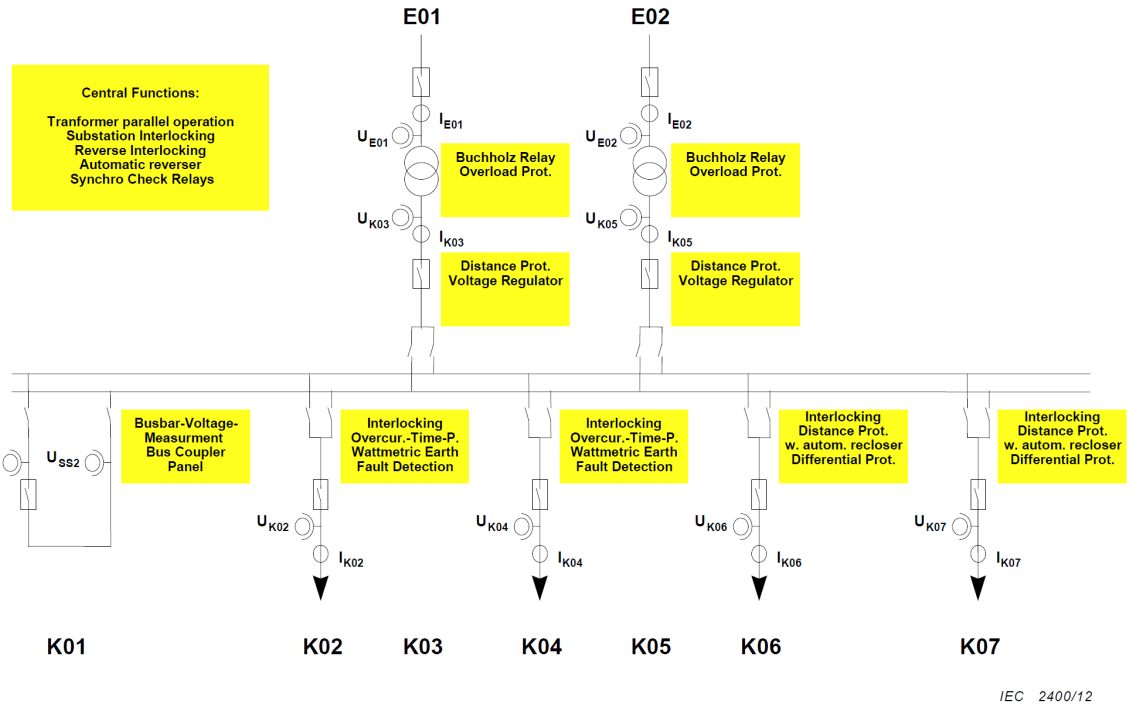

Figure 2.1: Example substation D2-1 according to IEC 61850 [20]

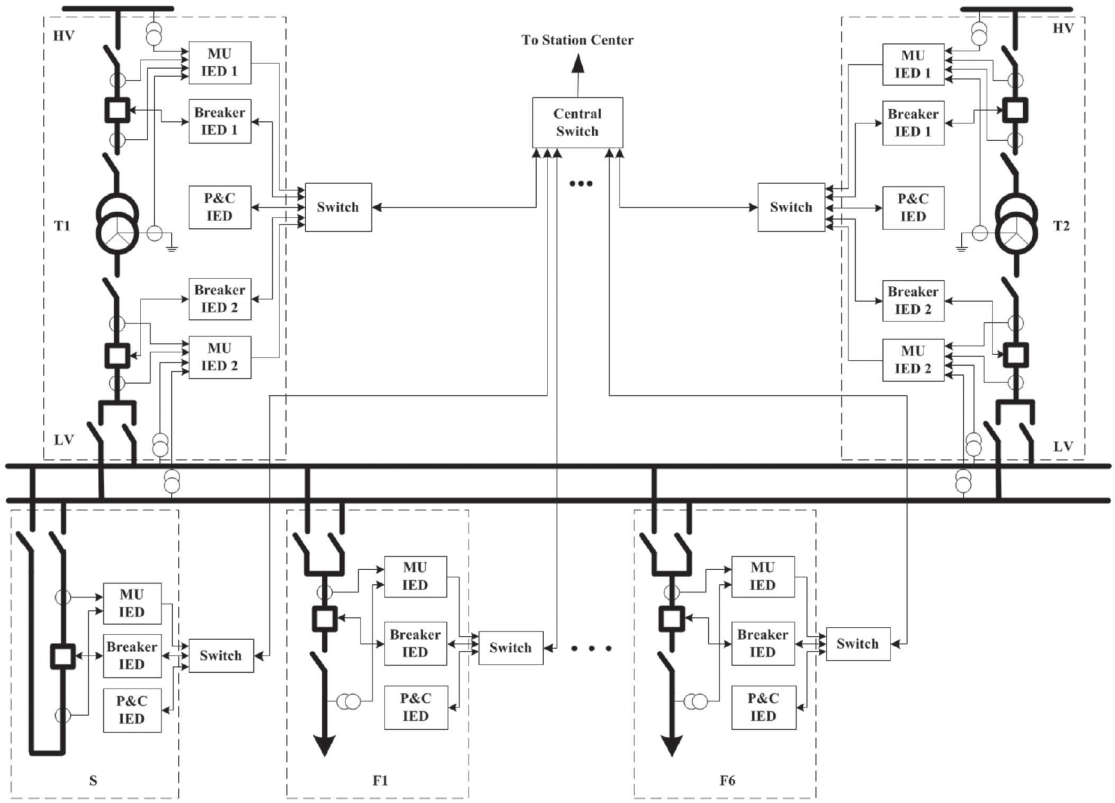

Figure 2.2: Example substation D2-1 with communication network [21]

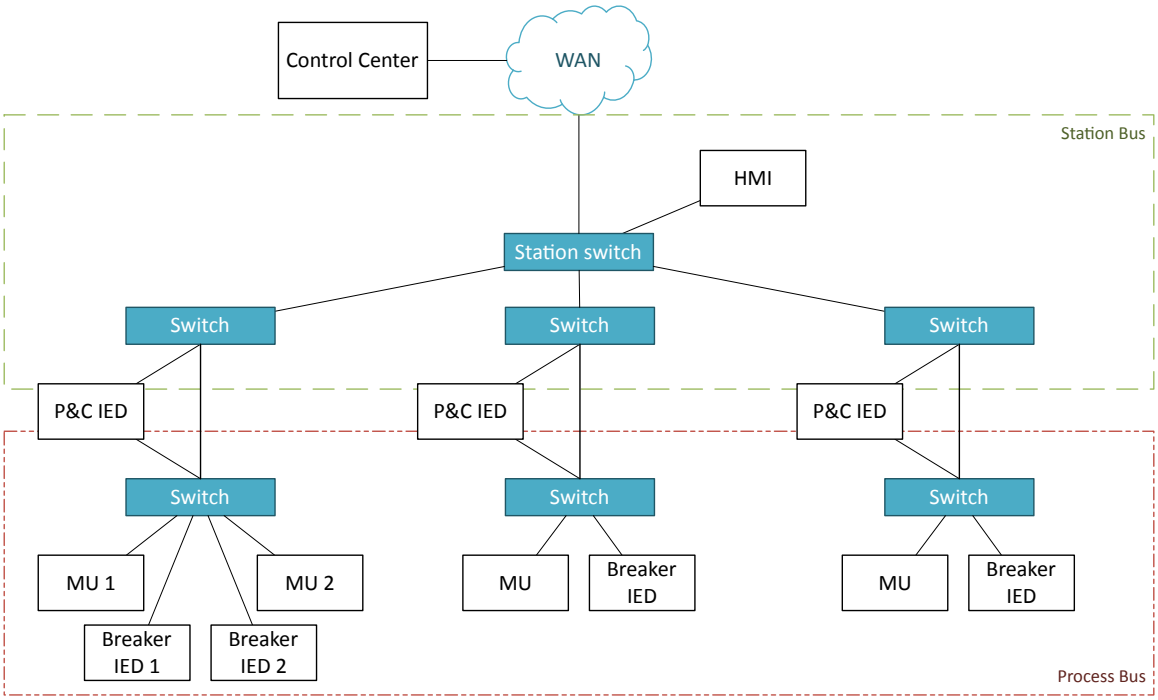

Figure 2.3: Example communication network for simplified substation D2-1 with distinct station and process buses

IEC 61850 specifies an overall architecture to be used for the communication network of a substation. It splits the network in two levels: the station bus and the process bus [22]. It is not necessary to keep them as two isolated and physically separate networks: as shown in Figure 2.2, the two buses can be one and the same. However, it is recommended to isolate them to avoid traffic congestion [22]. An example where the process bus and station bus are more distinct yet still connected is shown in Figure 2.3 (the primary equipment is not shown and only one of each kind of bay is shown). Whether or not to connect the buses depends on context. Both of the buses can use any suitable network topology, many of which are documented in IEC 61850-90-4 [22].

Station bus The station bus connects the bays together and to the gateway to allow communications outside of the substation. It carries many kinds of traffic, most notably exchanges relying on the Transmission Control Protocol (TCP) protocol such as Manufacturing Message Specification (MMS), File Transfer Protocol (FTP) or Simple Network Time Protocol (SNTP) [22]. It can also transfer GOOSE traffic [22]. Due to the nature of the traffic it carries, the station bus generally has less strict performance requirements than the process bus does.

Process bus The process bus is used to connect the MUs and related IEDs within one bay. It has much stricter performance requirements than the station bus does as it must support SV and GOOSE communications, both of which are critical to the substation [22]. It can also carry MMS [22]. The process bus can be completely isolated from the station bus if desired, meaning that only the IEDs connected to both buses can be accessed from the station bus.

2.3.2 Information Model and Abstract Communication Service Interface

Several kinds of messages are required for an IEC 61850 substation to function. They are classified into 6 messages types [20], [22] which are listed in Table 2.1. IEC 61850 defines an information model and related services to enable the use of these messages.

The information model uses an object-oriented approach. An IED contains an abstract Logical Device (LD) which holds several Logical Nodes (LNs) that each represents a function in the IED: examples include a CB, a measurement, or a transient earth fault [23]. The names given to the LNs are defined by the standard and meant to reflect each LN's function, enhancing readability. The first letter in the LN's name indicates in which of the 19 LN groups the LN belongs. For instance, the LN for a CB is named XCBR, with the letter X indicating that this LN is part of the group of switchgear functions [23]. The information in each LN is further split into Data Objects (DOs) and Data Attributes (DAs). An XCBR holds a DO named Pos (the position) that includes the DA stVal to indicate whether the breaker is open, closed or in some other state [23]. The available LNs and DOs are standardized in IEC 61850-7-2 [24] and IEC 61850-7-4 [25].

The Abstract Communication Service Interface (ACSI) provides a description of the services used to access and act on the data found in various LNs. The ACSI is abstract, as its name indicates, and must be mapped to some application protocol to be used in practice. This is done using a Specific Communication Service Mapping (SCSM) [23].

2.3.3 Application Protocols and Specific Communication Service Mapping

The IEC 61850 standard provides SCSMs to map the information model and ACSI to three protocols: GOOSE, SV and MMS [23]. Each has its own role and they are used along with existing protocols for purposes such as time synchronization. A summary of the application protocols is

Table 2.1: Message types defined in IEC 61850

Message type	Use
Type 1	Fast messages for protection functions
Type 1A	Trip messages, highest priority among Type 1
Type 2	Medium speed messages for automation functions
Type 3	Low speed messages for operator functions
Type 4	Raw data messages for continuous streams of data from IEDs
Type 5	File transfer functions
Type 6	Exchanges using access control for highest security

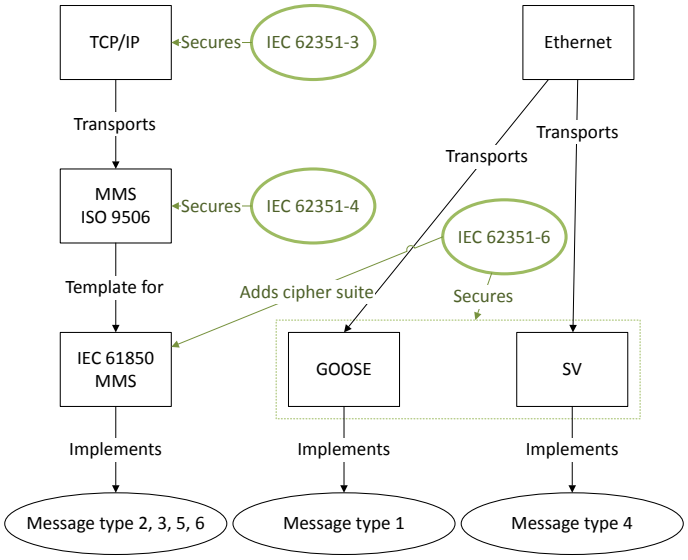

Figure 2.4: Application protocols used in IEC 61850 and their related standards

shown in Figure 2.4. The typical amounts of traffic they produce in a substation according to IEC 61850-90-4 [22] is shown in Table 2.2. Messages encoded and sent using any application protocol that implements ACSI services are referred to as Protocol Data Units (PDUs) [23]. The portion of a PDU carrying application data is referred to as Application Protocol Data Unit (APDU) [26].

Generic Object Oriented Substation Event (GOOSE)

GOOSE[3] messages are part of the Generic Substation Event (GSE) communication model that aims to quickly and reliably transmit information about events that occur in the substation [24]. They are used in IEC 61850 to transfer messages of type 1 and 1A, referring to fast messages and

[3]Not to be confused with Generic Substation State Event (GSSE), also part of GSE. GSSE resembles GOOSE, but has more limited functionality and is currently deprecated [24].

Table 2.2: Types of traffic usually found in IEC 61850 substation

Protocol	Message types	Affected buses	Approximate amount of traffic
GOOSE	1, 1A	Station and process	1 kbit/s (steady-state), 1 Mbit/s (bursts)
SV	4	Mostly process	6 Mbit/s per IED
MMS	2, 3, 5, 6	Station and process	10 kbit/s per IED
SNTP	Time synchronization	Mostly station	Unknown
PTP	Time synchronization	Mostly process	Unknown

trip messages in Table 2.1. Both of these are of high priority. GOOSE PDUs are created and sent using a publisher-subscriber model according to preconfigured conditions that indicate events and contain relevant data about the events in question [22]. They are used, for instance, to send critical trip signals to CBs in case of system instability. These PDUs have very strict performance requirements depending on the information they carry. The strictest requirement to transmit and receive a message is 4 ms [4].

Rather than rely on TCP at network layer 3, as MMS does, the GOOSE protocol works at network layer 2. In fact, it is an extension of Ethernet, meaning that IEDs in the substation address each other using Media Access Control (MAC) addresses. It leverages the multicast functions of Ethernet to enable the publisher to send GOOSE PDUs to all its subscribers at once [22].

GOOSE traffic is found at both process and substation level. A particularity of GOOSE PDUs is that they are constantly being transmitted. Even when no changes have occurred, GOOSE PDUs are re-transmitted at preconfigured intervals and carry the same information. This is meant to account for errors in the network [22] and for newly installed equipment that needs the information [24]. This re-transmission of PDUs accounts for the small amount of GOOSE traffic seen in Table 2.2 in steady-state. When a new event occurs, the amount of GOOSE traffic can greatly increase as the time between re-transmissions is temporarily made much lower than in unchanging conditions, in an attempt to ensure the subscribers receive at least one of the GOOSE PDUs on time [27].

The fields included in every GOOSE PDU are defined in IEC 61850-8-1 [27] and additional fields are defined in IEC 62351-6 [28]. They are listed in Table 2.3. Essentially, the GOOSE PDU represents the state of a DO to be communicated to subscribers, such as the status of a CB. The $allData$ field contains the values, while $datSet$ clarifies the meaning of the values by containing the name of the DO in question. The $stNum$ (state number) and $sqNum$ (sequence number) values

Table 2.3: Fields in GOOSE PDUs and their meaning

Field name	Data type	Description
$APPID$	Integer	Application identification (0x0000 to 0x3FFF for GOOSE type 1, or 0x8000 to 0xBFFF for GOOSE type 1A)
$length$	Integer	Length of this PDU, which is always (length of APDU + 8) bytes
$gocbRef$	String	GOOSE control block reference
TAL	Integer	Short for timeAllowedToLive, the maximum time (in milliseconds) the subscriber should wait for the next PDU
$datSet$	Integer	Data set, a reference to the specific DO and DAs being monitored and for which $allData$ is applicable
$goID$	String	GOOSE identification, user-defined name for this message
t	Timestamp	Timestamp, the time of the last $stNum$ change
$stNum$	Integer	State number, the number of times values in $allData$ have changed
$sqNum$	Integer	Sequence number, the number of retransmissions since the last $stNum$ change
$simulation/test$	Boolean	Test flag, if $true$, $allData$ contains simulated values for testing purposes, and if $false$, it contains real data
$confRev$	Integer	Configuration revision, the number of configuration updates to the data set named by $datSet$
$ndsCom$	Boolean	Needs commissioning flag, if $true$, this association requires maintenance
$numDatSetEntries$	String	Number of values contained in $allData$
$allData$	List	Values of the DO and DAs in the data set named by $datSet$
$AuthenticationValue$	Octet String	RSA digital signature for integrity checking (IEC 62351-6 only)

are used to distinguish new PDUs (new states) from re-transmitted ones [27]. PDUs carrying new $allData$ have a $stNum$ of (previous $stNum + 1$), and a $sqNum$ of 0. Re-transmitted PDUs have the same $allData$ and $stNum$ as preceding PDUs, and have a $sqNum$ of (previous $sqNum + 1$). As such, the recipient of the PDUs need only look at $stNum$ to find out if the values of the monitored DO have changed.

Sampled Values (SV)

As shown in Tables 2.1 and 2.2, the SV[4] protocol is used to communicate messages of type 4, meaning raw values of current and voltage in digital format, to other IEDs that require them. These values are sent from MUs that convert analog values received from field equipment into digital SV PDUs and transfer them to several receivers at once using a publisher-subscriber model [22].

In many ways, the SV protocol is reminiscent of GOOSE. It is also an extension of Ethernet,

[4]SV is referred to as Sampled Measured Values (SMV) in some publications, including IEC 62351-6 [28]. In this work, SV is used at all times for consistency.

Table 2.4: Fields in SV PDUs and their meaning

Field name	Data type	Description
$APPID$	Integer	Application identification (0x4000 to 0x7FFF for SV)
$length$	Integer	Length of this PDU, which is always (length of APDU + 8) bytes
$noASDU$	Integer	Number of ASDUs in this SV APDU
$security$	List (of 1)	Security field, holds security information; in practice it only contains $timestamp$ (IEC 62351-6 only)
$timestamp$	Timestamp	Time at which this SV APDU was sent (IEC 62351-6 only)
$asdu$	List of ASDUs	List of ASDUs (see Table 2.5)
$AuthenticationValue$	Octet String	RSA digital signature for integrity checking (IEC 62351-6 only)

Table 2.5: Fields in SV ASDUs and their meaning

Field name	Data type	Description
$svID$	String	Unique identification for the source of this ASDU
$datSet$	Object Reference	Data set, a reference to the specific DO and DAs being monitored and for which $sample$ is applicable; optional
$smpCnt$	Integer	Sample count, incremented on every message; reset to 0 on every sync pulse
$confRev$	Integer	Configuration revision, the number of configuration updates
$refrTm$	Timestamp	Refresh time of SV buffer; optional
$smpSynch$	Enumeration	Specifies the clock signal used to synchronize; 0 means no signal
$smpRate$	Integer	Sample rate, amount of PDUs sent per second; optional
$sample$	List	Values of the DO and DAs in the data set named by $datSet$

meaning it works at network layer 2 and uses MAC addresses to identify publishers and subscribers. It uses the multicast functions of Ethernet to continuously transfer PDUs to many receivers. However, SV differs from GOOSE in its much more significant and constant amount of traffic, as seen in Table 2.2. SV PDUs are transferred at rates ranging in the thousands per second, resulting in a heavier load on the network. IEC 61850-90-4 estimates that a maximum of 6 SV publishers can be connected to the same bus at 100 Mbit/s [22].

The fields for SV PDUs are defined by IEC 61850-7-2 [24], IEC 61850-9-2 [26] and additional fields are defined in IEC 62351-6 [28]. Each SV PDU holds an APDU which in turn can carry several Application Service Data Units (ASDUs). We list the fields found in these messages in Tables 2.4 and 2.5. Note that SV ASDUs do not generally carry all of the fields due to performance concerns: it is recommended to minimize the total size of the APDU by using very small names for $svID$ and not using optional fields [22].

Much like a GOOSE PDU, an SV PDU represents the values of a DO that is referred to by

$datSet$. In this case, the values of the DO are found in the $sample$ field. In addition, each SV ASDU carries a $smpCnt$ (sample count) value to identify it. This counter keeps track of the order of the SV ASDUs. Its value is (previous $smpCnt + 1$). Since $smpCnt$ is only a 16-bit value, it cannot exceed $2^{16} - 1$ (65,535), a value that would normally be reached within several seconds given the typical rate of SV PDUs in a substation. This usually does not happen as $smpCnt$ resets to 0 on every sync pulse [26], which occurs nearly every second [29]. The $smpCnt$ in typical traffic is therefore constantly increasing linearly, with periodic resets to 0.

Manufacturing Message Specification (MMS) and IEC 61850 MMS

MMS, as defined in its own standard named ISO 9506 [30], is used in several control systems applications. It is a generic communication protocol that provides a wide variety of features to model and exchange any data between client and server. Developers can use it as a template to implement new application protocols and enable communications with devices such as robots, IEDs or Remote Terminal Units (RTUs) [31]. For instance, existing derivatives of MMS are used for many applications, including inventory management, material handling, and power management [31]. Any protocol using MMS uses a model named Virtual Manufacturing Device (VMD) that specifies the objects, services and behavior of the application protocol [31]. It uses an object-oriented approach where MMS clients and servers maintain objects that represent their state and provide services for other parties to act on these objects [32]. An example of a generic service offered by MMS is the Get service to read the value of an object [32]. Possible object attributes include variables, logs, files, and more [27].

In the context of IEC 61850, the MMS protocol is leveraged to implement communications between IEDs in either the station or process bus and gateways or outside clients, such as the SCADA [22]. This specific implementation of the protocol is often referred to as IEC 61850 MMS [33]. MMS was chosen to implement these kinds of communications, as it provides many objects and services that mesh well with the abstract data model required by IEC 61850 parts 7-2 to 7-4 [27]. As is shown in Table 2.2, IEC 61850 MMS is used to communicate any message of a type not specifically covered by the other protocols. It is therefore more concerned with communications that are not as time-critical as the others, such as file transfer, operator actions, etc. It relies on the

TCP protocol to ensure the delivery of PDUs [27]. IEC 61850 MMS only requires a subset of all the functionality provided by the generic MMS protocol [27].

In this work, we distinguish the terms "MMS" from "IEC 61850 MMS" when relevant.

Time Synchronization

Time synchronization is critical to the correct operation of the substation. For this purpose, IEC 61850 recommends use of SNTP at the station bus and Precision Time Protocol (PTP) at the process bus [22]. SNTP works over User Datagram Protocol (UDP) using a client-server model to provide time synchronization. PTP is used at the process bus and instead broadcasts time using layer 2 multicast. PTP is recommended as it is considered more accurate than SNTP [22], which is needed especially for MUs using the SV protocol [29].

2.4 Simple Network Management Protocol

Simple Network Management Protocol (SNMP) is a protocol used for network management [34]. When using SNMP, devices in a network are monitored by a central Network Management Station (NMS) running an SNMP manager that provides monitoring and control features. Each managed device runs an SNMP agent that maintains information about its state as a set of objects arranged in a Management Information Base (MIB). If the device cannot run the agent itself, it can rely on a proxy to do so. The SNMP manager and the agents communicate by exchanging SNMP messages [35]. It is recommended to use two NMS's in a given network for redundancy [35].

There are three versions of SNMP. SNMPv1 is the first edition and is replaced by SNMPv2, which introduced many changes to MIB definitions, rendering these two versions incompatible with each other. SNMPv3 is essentially the same as SNMPv2, but introduces security features [35]. Because the IEC 62351-7 standard mandates the use of SNMPv3 [8], we center our discussion mostly on SNMPv2 and SNMPv3.

2.4.1 Management Information Bases and Objects

Objects in SNMP are arranged in a Management Information Base (MIB), which is a tree structure. Every node in this tree has an integer and a descriptive string associated with it. To reference a specific object in the MIB, one must use the object's Object Identifier (OID), which consists of the integers of all the parent nodes for this specific object (separated by dots). Alternatively, one can use the strings instead of the integers in the same manner [34]. For example, the MIB object representing the SNMP agent's uptime has the OID 1.3.6.1.2.1.1.3 or "iso.identified-organization.dod.internet.mgmt.mib-2.system.sysUpTime". Both notation have the same meaning, as "iso" corresponds to the integer 1, identified-organization corresponds to 3, and so on [36]. An important aspect to consider is that every OID is expected to be *globally* unique across all implementations. That is, two objects that have different meanings and that are defined by two separate organizations cannot share the same location (OID) in the MIB tree. For instance, we should not reuse the OID 1.3.6.1.2.1.1.3 to define a new MIB object. This would break compatibility with existing SNMP implementations, since it is already assigned to the sysUpTime object. To request allocation of a branch in the MIB tree, entities should register at a Registration Authority (RA) [36]. This avoids conflicts between the OIDs and ensures that all devices with SNMP use the same structure for their MIB.

In addition to an OID, MIB objects have a type and encoding. They are defined in a MIB file using Abstract Syntax Notation One (ASN.1) and encoded using Basic Encoding Rules (BER) [34]. ASN.1 provides several basic types such as Integer, Octet String, or Sequence, while additional types can be defined as needed [34]. We do not delve into the specifics of MIB definitions as we leverage existing MIB files as part of this work, foregoing the need to define MIBs ourselves. However, we note that it is not possible to update already existing MIB objects, such as by changing their types, as the new definitions would conflict with the original ones due to sharing the same OIDs. New MIB objects must be defined for such a case and the old objects must be deprecated.

2.4.2 Messages Available

The following messages can be used in SNMP [35]:

Get and GetNext The Get message allows the NMS to request the current value of a specific MIB object. The object must allow reads for this to be successful. GetNext works similarly, but requests instead the value of the object *after* the one specified in the request, according to the MIB hierarchy. GetNext is meant for traversing rows in SNMP tables.

GetBulk The GetBulk message works like a Get request, but the reply can carry an entire table as opposed to only one row. This request is designed to reduce the overall number of SNMP requests required to obtain the value of many MIB objects.

Set The Set message is used by the NMS to write to an object stored at an SNMP agent. The object must allow writes for this command to be successful. In the context of this thesis, this message goes mostly unused.

Trap Unlike the other messages, traps are sent from the SNMP agent to the NMS. Traps are used to notify the NMS of an event without requiring the NMS to poll them. The rules deciding when traps are set are not explicitly defined.

In most cases, SNMP uses UDP as its transport protocol as it is connection-less protocol and therefore more efficient, though it can also use TCP [35]. A downside of using UDP is that a sender does not know if their packet was ever received by the recipient, because UDP does not require sending a response back. In the case of SNMP, this is not an issue when sending SNMP requests as the NMS can simply resend its request if there is no answer. However, SNMP traps are problematic with UDP, as the NMS will never know if a packet gets dropped due to an error or attack. The agent is not required to confirm that the trap was received [34]. TCP does not have this problem, but this protocol tends to perform poorly when the network is congested, a situation where network monitoring is critical [34].

2.4.3 Security

SNMPv1 and SNMPv2 both rely on community strings, which are essentially passwords, for the purpose of authentication and authorization. It is generally accepted today that these versions of SNMP provide virtually no security, since all SNMP messages (including the community strings)

are sent over the network without encryption. It is trivial for an attacker to sniff packets to retrieve this sensitive information and even control the agents [37]. This is a major concern since SNMP can be used to execute Distributed Reflected Denial-of-Service (DRDoS) attacks, explained further in Section 5.1.3, in a network once the attacker knows the credentials [38]. Worse still, many manufacturers ship devices with SNMP enabled and the default community strings "public" or "private", meaning that attackers do not even need to bother finding the community string [37], [38].

SNMPv3 is intended to introduce proper security to the protocol by replacing community strings with the User-based Security Module (USM) [37]. Access control to MIBs in SNMP agents is managed using securityNames (usernames) and passphrases. Permissions to read and write specific parts of the MIB can be assigned to users to limit their privileges to what they need. In addition, authentication and confidentiality are added to SNMP communications using cryptography [37]. Specifically, SNMPv3 can work at three levels of security: none (noAuthNoPriv), with authentication only (authNoPriv), or with authentication and privacy (authPriv). Use of noAuthNoPriv is highly discouraged as it provides no security [38] and defeats the purpose of using SNMPv3. The key used for securing communications is actually a Hashed Message Authentication Code (HMAC) of the SNMP agent's unique snmpEngineID. It uses either MD5 or SHA-1 as the HMAC's algorithm and the password as the key to the HMAC [39]. The resulting key is then used to produce a keyed hash for authentication purposes. It is also used for encryption using Data Encryption Standard (DES) or AES when privacy is required. Because it is based on the SNMP agent's unique snmpEngineID, it should be different for every agent even if they share the same usernames and passwords. There are some attacks possible on this scheme [39], but they rely on the agents sharing credentials or the presence of Dynamic Host Configuration Protocol (DHCP). The security of SNMPv3 itself is outside the scope of this thesis, so we do not discuss these attacks in detail.

Overall, to secure SNMP, the latest SNMPv3 with its security features should be used. Note that some of the algorithms recommended by SNMPv3, such as DES and MD5, are considered outdated by NIST and should ideally be avoided in favor of AES and SHA-1 [11]: although SHA-1 is not ideal according to NIST, it is the most secure hash function supported in SNMPv3 at the moment. Devices that only support versions prior to SNMPv3 cannot be secured with cryptography in this

manner. The most reliable solution in that situation is to disable SNMP altogether for those devices. If use of SNMPv1 or SNMPv2 is unavoidable, one must rely on other methods like ingress filtering or access control lists to provide security [38].

2.5 IEC 62351: Standard for Cybersecurity of Power Systems

IEC 62351, entitled "Power systems management and associated information exchange - Data and communications security", is a standard published by the IEC that aims to enhance cybersecurity in the context of power system control operations [7]. Previous standards, including IEC 61850, do not include cybersecurity features in their scope, motivating the need for this new standard. It is divided into several parts. Parts 3 to 6 focus on extending existing IEC TC 57 communication protocols, such as IEC 61850 and IEC 60870-5, to improve security of communications. Part 7 of the standard introduces NSM to provide end-to-end security. The remaining parts cover other aspects of cybersecurity in the power systems context, such as key management, security architecture, etc. As our focus is on IEC 62351-7 in the IEC 61850 substation, we briefly discuss each part that is relevant to it:

- Part 1, as it discusses cyberattacks and the ones the standard is meant to address;
- Part 3, as it aims to secure TCP, used by MMS and other application protocols;
- Part 4, as it secures MMS;
- Part 6, as it secures the protocols of IEC 61850, namely GOOSE, SV and IEC 61850 MMS.

The parts we do not discuss include part 2 (the glossary), part 5 (concerned with IEC 60870-5 protocols and not IEC 61850), and parts 8 and above that discuss other security measures.

2.5.1 IEC 62351-1: Introduction to the Cybersecurity Standard

IEC 62351-1 provides an overview of the rest of the standard as well as a discussion on the attacks included in its scope.

The objectives of the IEC 62351 standard can be summarized as the following [7]:

- Authentication for entities and their actions;
- Confidentiality of messages and keys;
- Integrity by detecting tampering and preventing playback or spoofing;
- Monitoring devices and networks for availability and a "degree of intrusion detection" [7];
- Allow secured and non-secured devices on the same network for backward compatibility;
- Identity management policies (described in IEC 62351-8).

As can be seen by the above, IEC 62351 aims to fulfill the security requirements mentioned in Section 2.1. While fulfilling all requirements is desirable, some of the requirements are considered more critical than others in the context of an IEC 61850 substation. Generally, availability and integrity have the highest priority, while confidentiality has the lowest [6]. This is because the focus of the substation is to ensure that power delivery is continuous and controlled. Availability is essential to ensure that utilities can always perform the required control operations within the time constraints needed to stabilize or restore the grid [6]. Integrity, along with authentication, authorization and non-repudiation, is needed to prevent these same control operations from being abused by unauthorized parties to interfere with power delivery. This is exactly what occurred during the cyberattack targeting Ukraine in December 2015, where the attackers remotely operated CBs to deliberately cause a power outage [18]. Confidentiality is considered the least important of the requirements. This is exemplified by the specifications in IEC 62351-6, where encryption is made optional instead of mandatory due to its possible negative impact on performance (and hence availability) [28]. IEC 62351-1 states that encryption "is not considered that important" in the context of GOOSE and SV protocols. It also states that in general, authenticating commands is of higher priority than hiding data [7]. Despite this, confidentiality is still essential to prevent attackers from gathering knowledge on the layout of the substation and to protect information credentials used by operators, as this kind of knowledge can be used in further attacks once obtained.

2.5.2 IEC 62351-3: Security for TCP Using Transport Layer Security

IEC 62351-3 is named "Profiles including TCP/IP" and applies to any protocol relying on the TCP protocol for its transport layer [40]. It is intended to be referred to by other standards (with a good example being IEC 62351-4, concerned with MMS). This part of IEC 62351 aims to provide integrity, message-level authentication and confidentiality by using the Transport Layer Security (TLS) protocol, which already sees widespread use in IT networks. It also takes into account that TCP is used differently in telecontrol environments. Connections tend to be much longer (or even permanent) when compared to other contexts, this affects certificate expiration and revocation [40]. Since TLS only works when both parties exchanging messages agree on the same cipher suite, the standard specifies some TLS cipher suites that should be supported by all compliant devices to ensure interoperability between them.

2.5.3 IEC 62351-4: Security Extensions for MMS T-Profile and A-Profile

IEC 62351-4 has the title "Profiles including MMS" and aims to provide security for the MMS protocol [41] as shown in Figure 2.4. This includes any protocol that leverages MMS as its template. As explained in Section 2.3.3, MMS is used to derive the IEC 61850 MMS protocol, making IEC 62351-4 applicable to the substation context. IEC 62351-4 provides security at two network layers: the T-Profile for the transport layer, and the A-Profile for the application layer.

T-Profile

For the T-Profile over TCP [5], it refers to IEC 62351-3, described in Section 2.5.2. IEC 62351-4 specifies the parameters to be used with TLS, namely network ports, cipher suites, how to handle certificate revocation, and so on [41].

A-Profile

For the A-Profile, IEC 62351-4 specifies how to authenticate entities during creation of an initial association. Specifically, it states to add fields in the PDUs used for Association Control Service

[5]The T-Profile can use Open Systems Interconnection (OSI), but it is out of scope for IEC 62351-4 [41].

Element (ACSE) authentication: the AARQ (request) and AARE (reply). When creating an association, the MMS client must include three fields in the AARQ and AARE [41]:

(1) $SignatureCertificate$ to carry the X.509 certificate used to verify $SignedValue$;

(2) $SignedValue$ to carry a digital signature of the $time$ field;

(3) $time$ to represent the creation time of the request.

The receiver of the AARQ or AARE must then verify that the $SignedValue$ corresponds to a valid signature using the $SignatureCertificate$ and $time$ fields. The $time$ field must also have a difference of less than 10 minutes compared to the receiver's local time. The signature must be one that the subscriber has never seen before. The PDU is only accepted if all of these conditions are true. Otherwise, a P-ABORT is issued [41]. This makes it not possible for intruders to create a new association if they cannot produce the necessary $SignedValue$.

Outside of the creation of an association, there are no other PDUs that use this authentication mechanism [41]. This is presumably because the T-Profile should already be secured by TLS, providing defense against a great number of cyberattacks.

2.5.4 IEC 62351-6: Security Extensions for GOOSE and SV

IEC 62351-6, entitled "Security for IEC 61850", focuses on the protocols found in that standard, namely the IEC 61850 MMS, GOOSE and SV protocols [28]. The only new security addition to IEC 61850 MMS provided by IEC 62351-6 is an extra cipher suite for TLS. For the rest, it refers to IEC 62351-4, which provides security for MMS in general, as discussed in Section 2.5.3 [28]. For this reason, we do not explicitly include IEC 61850 MMS in the scope of IEC 62351-6 as shown in Figure 2.4. We instead focus on its security additions for GOOSE and SV.

The main contribution of IEC 62351-6 to GOOSE and SV security is twofold. The first is the addition of a new field named $AuthenticationValue$, used to check for integrity, and optional AES-128 encryption to GOOSE and SV PDUs. The second is modifications to the GOOSE and SV protocols in an effort to counter replay attacks [28]. Of note is that the standard does not mandate the use of encryption in all packets due to performance issues that arise in certain contexts, but

it recommends to use encryption whenever it does not cause problems. Using it, the contents of messages are hidden from passive attackers listening to traffic, which prevents theft of data and makes some other attacks more difficult to execute [28].

IEC 62351-6 introduces a total two new fields to GOOSE and SV PDUs:

(1) $AuthenticationValue$ to carry RSA digital signature: the input is a SHA-256 hash of PDU's contents;

(2) $timestamp$ to represent the creation time of the PDU (only used for SV).

Publishers must add a valid $AuthenticationValue$ to every PDU. Subscribers can then verify $AuthenticationValue$ to confirm the legitimacy of the PDU by validating the signature, since only the true publisher knows the private key required to produce it. With this change, attackers can no longer spoof or modify packets as both require producing a valid signature. The new $timestamp$ field for SV PDUs is part of the measures introduced to protect against replay attacks, which we discuss next.

Replay Protection for GOOSE

Discarding of lower $stNum$ In normal circumstances, a GOOSE message with a higher $stNum$ indicates it is more recent. As a result, IEC 62351-6 has the GOOSE subscriber record the last $stNum$ it has received from a given publisher and discard any PDU from this publisher if they contain a lower $stNum$ [28]. This behavior handles PDUs that arrive out-of-order so that the subscriber does not react to past events, potentially causing issues. It also protects against replay attacks: old recorded PDUs typically have a lower $stNum$ than the latest one, because $stNum$ values should only ever increase between successive PDUs. Recorded PDUs are therefore rejected by the subscriber when they are replayed.

Resetting of $stNum$ The $stNum$ is set back to 0 in two cases. In the first case, the 32-bit value overflows, meaning it exceeds its maximum of $(2^{32} - 1)$. This only occurs every several years if we assume a rate of 30 PDUs per second, making it a rare but not impossible event [42]. The second case is when a message timeout occurs. If the subscriber does not receive a GOOSE PDU within

the time allowed to live (TAL) specified in the TAL field of the last PDU it received, the subscriber assumes the communication is lost [27] and counts this as a message timeout[6] [28]. We refer to such an event as a TAL expiration, to match terminology used in IEC 62351-7 [8]. Presumably to receive packets once the publisher has recovered, the $stNum$ is set back to 0 [28], [42]. The exact intent behind this change is not explained.

Skew filtering Skew filtering is meant to limit the time during which a given GOOSE PDU is considered valid. The skew period, which does not exist in normal GOOSE, is a time window where the subscriber can accept a GOOSE PDU with a higher $stNum$ (i.e. that represents a change of state). The subscriber determines this time window by comparing its own local time to the value of t in the PDU it is validating. The t field represents the last time $stNum$ changed. If t is generated within the skew period, the subscriber accepts this PDU. Otherwise, it filters (discards) it [28]. The skew period is configurable and ranges from 10 to 120 seconds in the past [28]. Skew filtering limits replay attacks significantly, as PDUs older than the skew period cannot be used.

Replay Protection for SV

Many of these specifications for SV are reminiscent of the ones discussed previously for GOOSE.

Addition of $timestamp$ SV PDUs do not include a timestamp field, despite IEC 61850-7-2 mentioning that timestamps should be present [24]. This is not an issue in normal situations where the publisher and subscriber use synchronized clocks and exchange messages in real-time. However, if we consider the possibility of a delay or replay attack, it is problematic. Without the timestamp, it is virtually impossible to determine the creation time for a given SV PDU and assess its credibility. To address this, IEC 62351-6 introduces a $timestamp$ field to be contained in every SV packet [28].

Discarding of lower $smpCnt$ The $smpCnt$ value found in SV messages tracks the order of the PDUs. A PDU with a higher $smpCnt$ is usually more recent (ignoring cases where $smpCnt$ is reset to 0). For this reason, the SV subscriber should drop incoming PDUs where the $smpCnt$ is

[6] IEC 62351-6 never explicitly defines what is a "message timeout", but it is reasonable to assume it refers to a TAL expiration. This is also the interpretation used in the work of Strobel *et al.* [42].

lower than the last one it received, unless a reset occurred [28]. This is the same mechanism as the one specified for GOOSE to process the $stNum$[7].

Reset of smpCnt IEC 62351-6 specifies the two cases where the $smpCnt$ should be reset to 0 and they are essentially the same as those described for resetting the $stNum$ in the GOOSE protocol. The first case is when "there is a message timeout" [28]. It is unclear what timeout this refers to in the context of SV, as there is no such timeout described in IEC 61850. The second case is when the value of $smpCnt$ overflows [28]. There is no mention of how the $smpCnt$ is already supposed to reset on every sync pulse, as described in IEC 61850-9-2 [26].

Skew filtering Skew filtering works mostly the same way as it does for GOOSE, but with a few key differences. The skew period used for SV is specified to be 2 minutes [28], rather than configurable like the one used by GOOSE. Additionally, all SV PDUs are subjected to the filtering: to verify if an SV PDU is within the skew period, the subscriber compares the PDU's $timestamp$ (described previously) with the local time [28]. This not possible for GOOSE PDUs since the latter do not have this timestamp field: they only carry the t field, which has a different purpose.

2.6 IEC 62351-7: Network and System Management (NSM)

Among the many parts of this standard, part 7 focuses on end-to-end security monitoring of the communication infrastructure that supports the power infrastructure, referring to it as Network and System Management (NSM) [8]. This monitoring is done to detect security events or other faults in an accurate and effective way to help operators in responding appropriately. Such a feature is not usually part of typical SCADA monitoring, hence the relevance of the standard.

IEC 62351-7 specifies that security monitoring be done by using the NSM Data Objects (NSM DOs)[8] that it defines. The main advantage of these objects is how they hold health information that is specific to the power operations environment [8]. In addition, since they are standardized, it is expected

[7] IEC 62351-6 refers to $stNum$ and $sqNum$ when discussing SV, though those fields do not exist in SV messages. We assume this is an oversight and that the standard actually refers to $smpCnt$, as this value has similar behavior to $stNum$ and $sqNum$.

[8] This acronym is not used by IEC 62351-7. It defined and used in this work for simplification.

that devices from different manufacturers will use the same NSM DOs so that they can all be monitored using the same tools.

2.6.1 Objectives of IEC 62351-7

According to IEC 62351-7, the previous parts of the standard, IEC 62351-3 to IEC 62351-6, leave gaps in security even when all their specifications are implemented. They are essential as they mitigate many categories of attacks such as unauthorized modification or sniffing of communications. However, IEC 62351-7 states that the major threats they do not cover are attacks against the devices or users and DoS attacks [8]. The latter represent an especially serious gap as the availability of a power system is one of its most critical requirements. In addition to this, we also consider that there are situations where the specifications of IEC 62351-3 to IEC 62351-6 might not be applied. This is usually due to potential performance concerns in some contexts [43], [44], or because of the presence of older devices that do not support these standards. The ability to use both secure and non-secure devices is explicitly mentioned as one of the objectives of IEC 62351 [7]. It cannot be assumed that the security extensions are used, as devices might disable them to allow for backward compatibility with older equipment.

The goal of IEC 62351-7 is to provide end-to-end security [8], and address the above threats. It specifies the use of NSM to monitor the devices and the network communications. By collecting relevant NSM DOs provided by the devices themselves, NSM can assist existing Intrusion Detection Systems (IDS's) and provide a complete picture of the network and its current status. In the case of an event, such as equipment failure or an active intrusion by an attacker, NSM can use this data to narrow down the source of the problem. It can potentially detect problems before they escalate and cause significant damage, giving operators a chance to react appropriately.

Differences from SCADA

There is a distinction to be made between monitoring of the power systems as done by SCADA systems and NSM. IEC 62351-7 distinguishes two infrastructures: the power system infrastructure and the information infrastructure [8]. The former includes the power equipment and the IEDs forming the electrical network that is needed to deliver power. These are managed using IEC 61850

or SCADA protocols to monitor and control the system in terms of voltages, frequencies or currents. The information infrastructure is a layer above the power system. It is the communication network that links all this equipment together, for example by using switches or routers. NSM aims to monitor the information infrastructure: it is not concerned with the raw data about the electrical network, but rather in knowing whether or not the device handling that data is functioning correctly.

NSM aims to add monitoring capabilities that complement SCADA protocols and cover existing gaps. According to IEC 62351-7, SCADA protocols by themselves usually provide limited monitoring [8]. They can conclude that a Remote Terminal Unit (RTU) is not available if it is not responding to queries, but they cannot determine the source of the problem, whether it is a router that has crashed, an overload of the network or a problem at the RTU itself. As NSM monitors all communications equipment as well as the end devices themselves, it can determine if the problem is in the power system infrastructure or in the information infrastructure [22].

2.6.2 Differences between Editions

As of 2018, IEC 62351-7 has been published twice to update the list of NSM DOs. The first edition, published in 2010 [45], defines a set of abstract NSM DOs split into three categories to be used for power systems. It does not specify the use of a particular communication protocol when implementing them. The second edition, published in 2017 [8], replaces it entirely and defines a different set of NSM DOs and categories, accompanied by concrete mappings to the SNMP protocol [46]. As these two editions contain major differences, they are referred to as IEC 62351-7:2010 and IEC 62351-7:2017 when it is necessary to distinguish them. Since the NSM DOs defined in IEC 62351-7:2010 are not used in this work, any mention of IEC 62351-7 almost always implicitly refers to IEC 62351-7:2017.

2.6.3 NSM Data Objects Overview

The IEC 62351-7 cybersecurity standard provides NSM DOs meant to assist in monitoring and securing power systems [8]. An NSM Data Object (NSM DO) is a structure that contains attributes, each of which in turn hold a piece of information about the monitored device. As a simple example, the Name attribute of the IED agent contains the device's name. The standard defines NSM DOs

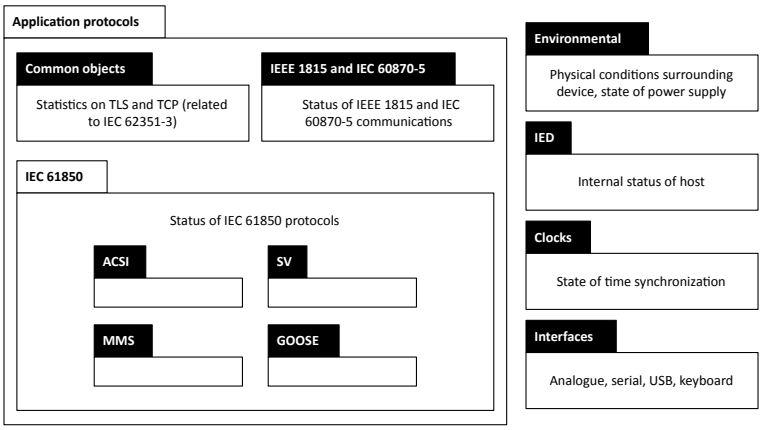

Figure 2.5: UML packages for NSM DOs defined by IEC 62351-7

using Unified Modeling Language (UML) [8]. They are abstract in order to allow mapping them to any protocol. Each NSM DO is therefore defined as a UML class or attribute. The NSM DOs are divided into multiple packages, also called agents [8]. The packages for the abstract NSM DOs are shown in Figure 2.5. Equipment manufacturers can choose to support or not support a given agent depending on how relevant it is for the equipment in question, so an individual device should not be expected to provide all of them. A description of each agent follows.

Environmental agent The NSM DOs for the environmental agents provide information about the environmental conditions of the monitored device [8]. This is includes whether there is currently physical access to the system (including the building, the perimeter, and the device itself). A sizeable amount of NSM DOs in this package focus on the conditions of the power supplies providing power to the monitored device.

IED agent Whereas the environmental agent NSM DOs represent the surrounding conditions of an IED, the NSM DOs for this agent describe the internal status of the IED itself [8]. It provides data on the health of the device, its Central Processing Units (CPUs) and storage. Some NSM DOs are specifically tailored for cybersecurity. Examples are NSM DOs concerned with the number of expired or revoked security certificates, the number of detected cyberattacks and any changes to the Role-based Access Control (RBAC) user database [8].

Clocks agent This package is concerned with monitoring clocks and detect failure or tampering with time signals [8]. Due to the importance of time synchronization in IEC 61850 substations, these NSM DOs are critical in detecting attempts at disrupting communications by attacking devices' local times. They contain information about the time synchronization protocol in use, the estimated accuracy of the clock, whether the synchronization signal is available, and detected attempts to tamper with the clock. In keeping with their intended role, none of these NSM DOs provide information in terms of timestamps like the other packages do, because their purpose is to indicate whether this signal is trustworthy in the first place.

Interfaces agent The interfaces agent's NSM DOs contain information about available Ethernet, analogue, serial, Universal Serial Bus (USB) and keyboard interfaces [8]. They report on how many of these interfaces are present on a device, how many of them are activated, and how many are failing.

Application protocols agents This package contains NSM DOs for the status of application protocols used in power systems [8]. It is split into three packages: common, IEEE 1815 and IEC 60870-5, and IEC 61850. In this work, we are interested in the common and IEC 61850 packages only as IEEE 1815 and IEC 60870-5 are out of scope. The common package are meant to be applicable to multiple application protocols [8], but in practice, its NSM DOs are mostly related to the use of TLS and TCP as described by IEC 62351-3. The IEC 61850 package is further split into four sub packages: one for each of ACSI, MMS, GSE (including GOOSE) and SV. This split is necessary as these protocols have different characteristics and, therefore, different NSM DOs.

2.6.4 NSM Data Objects as SNMP MIBs

In this work, as we implement the NSM DOs using SNMP, we leverage their definitions as SNMP MIBs for implementation purposes. These MIBs are publicly available on the IEC's website [47]. In these MIBs, the NSM DO attributes are renamed so that they have unique names, in order to conform to SNMP. This is why our previous example, the Name attribute of IED agent, is referred to as iEDName in the MIB. The mapping to SNMP also requires assigning OIDs to each

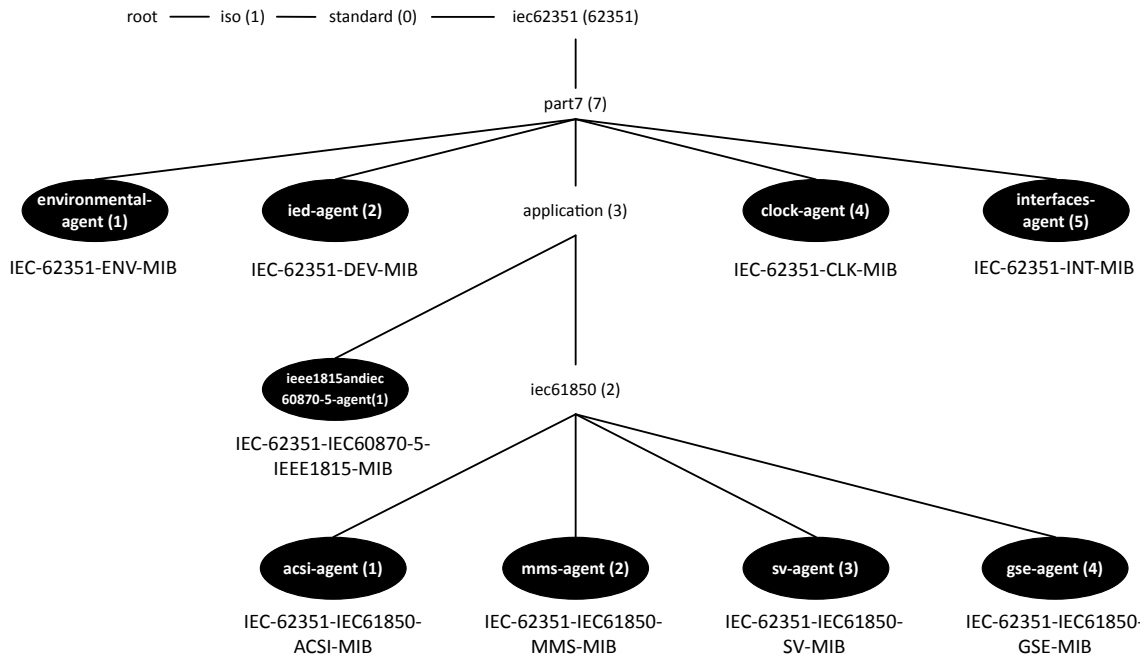

Figure 2.6: SNMP OIDs for the packages defined for IEC 62351-7 NSM DOs

of the NSM DOs. Figure 2.6[9] shows the OIDs for the packages shown in Figure 2.5. Any of the NSM DOs' attributes are defined under their respective agent's OID. The definition of the OIDs in the MIB files explicitly refer to the abstract NSM DO they represent, so it is easy to locate the OID corresponding to a specific NSM DO.

In addition to these new SNMP MIBs, IEC 62351-7 also requires the use of other existing MIB extensions for other information [8]. The MIBs named are TCP-MIB [48], TCP-ESTATS-MIB [49], SNMP-TLS-TM-MIB [50], UDP-MIB [51], IP-FORWARD-MIB [52], IP-MIB [53], and IPMCAST-MIB [54]. These provide sufficient information on TCP, UDP and Internet Protocol (IP) protocols for the standard's needs. If any other publicly available MIBs prove useful in monitoring a power system, they can also be used. MIBs are written in such a way that their OIDs do not conflict with each other, as mentioned in Section 2.4.

For the remainder of this work, we favor using the abstract names for the NSM DOs rather than their equivalents in SNMP as they are easier to read and applicable to other protocols.

[9] As a minor note, the MIB named IEC-62351-ENUM-MIB as found in the publicly released MIB files has no real location in the tree, as it only defines types and not objects, so it is not shown.

Chapter 3

Related Work

This chapter documents existing research on topics relevant to the work described in this thesis. An overview of them is shown in Figure 3.1. We focus on five topics: security assessment of IEC standards, automated protocol analysis, design and implementation of NSM, existing smart grid models and testbeds, and intrusion detection using either SNMP or ICS traffic.

As we are concerned with elaborating cyberattacks against IEC 61850 protocols, it is essential to investigate how these protocols work and if there are any known attacks against them. There are two standards of interest in this regard: IEC 61850 and IEC 62351. Note that some of the existing work only considers the specifications of IEC 61850 when assessing protocol security. These past contributions are still relevant, though, because the security measures of IEC 62351 might not be applied in some contexts for various reasons, such as backward compatibility or performance issues. Additionally, we investigate possible ways to automatically locate vulnerabilities in communication protocols, which could assist in designing attacks. In order to evaluate NSM as defined by IEC 62351-7, we are interested in setting up a smart grid testbed monitored by NSM. While there are several smart grid testbeds described in the literature that can be used to simulate attacks on power systems, few make use of NSM. In fact, much of the existing work on NSM is applicable to the 2010 edition of IEC 62351-7 rather than the one published in 2017. Since there is little work done on NSM DOs of IEC 62351-7:2017, we also look into methods to perform Intrusion Detection (ID) using similar data, most notably SNMP MIB objects and ICS protocol traffic. These techniques could be adapted to include the use of NSM DOs.

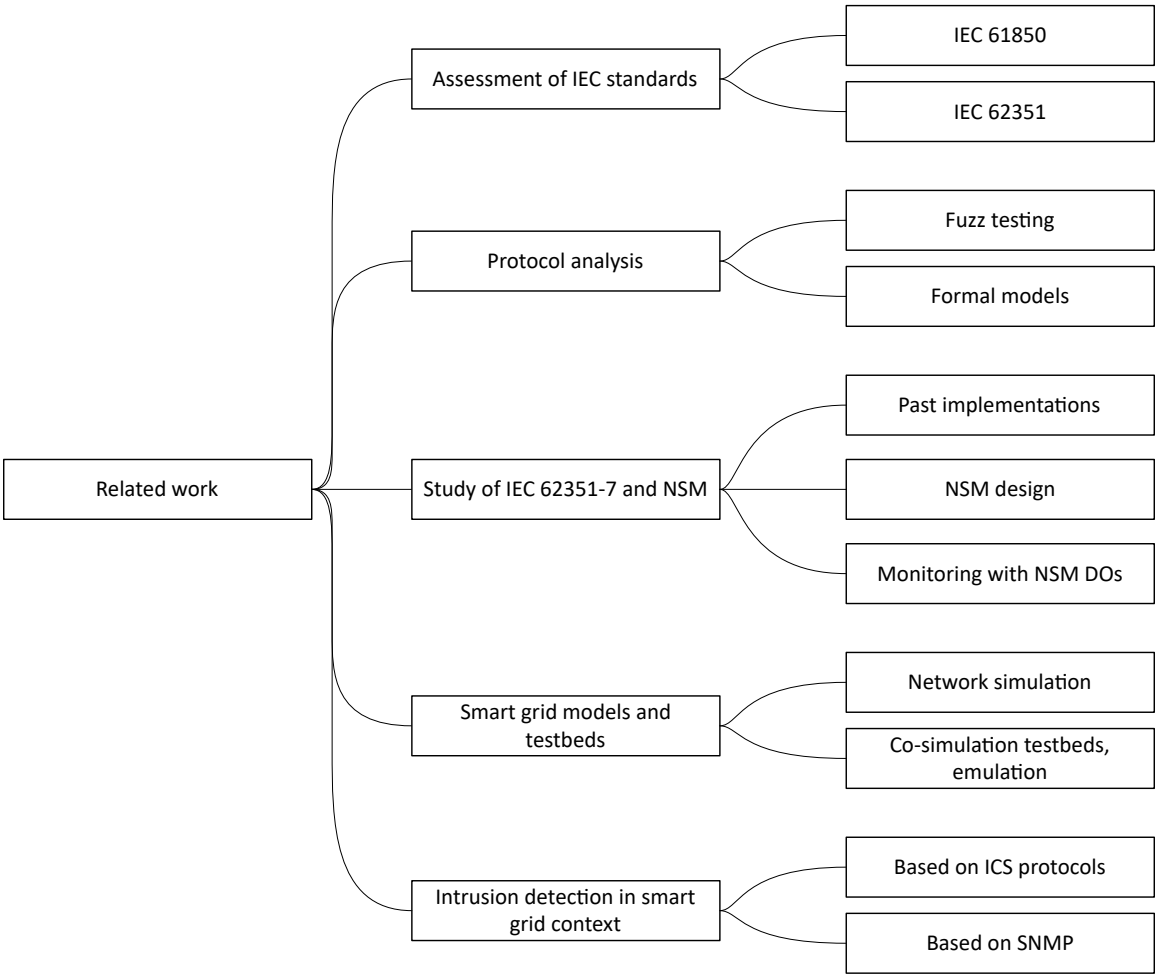

Figure 3.1: Overview of topics covered in related work

3.1 Security Assessment of IEC Standards

As this research is concerned with the cybersecurity of an IEC 61850 substation, past research on vulnerabilities found in IEC 61850 protocols, including their extensions specified in the security standard IEC 62351, is of utmost importance. NSM as per IEC 62351-7 is meant to be used alongside the other parts of IEC 62351 to mitigate threats. Thus, it is essential to know what cyberattacks are addressed by these other means in order to identify the ones that require mitigation by NSM.

As an introduction to existing threats in the smart grid context, the National Electric Sector Cybersecurity Organization Resource (NESCOR) documents a great number of failure scenarios, some of which that can caused by cyberattacks. The document is meant to assist utilities in assessing risks [55]. It covers many domains including Advanced Metering Infrastructure (AMI), Wide

Area Monitoring, Protection, and Control (WAMPAC), and Distribution Grid Management (DGM). While the NESCOR scenarios are not specific to the IEC 61850 substation, many of them can be adapted to fit the substation context. Scenarios considered more critical are covered in even more detail in a separate document [56], where they are extended with attack trees. Among these critical scenarios is the scenario DGM.11, where an attacker gains access to the Distribution Management System (DMS) and sends sequential tripping commands to critical feeders and cause a blackout. This scenario could easily be recreated by sending remote commands to IEC 61850 substations and is worth considering. Additionally, the steps undertaken by an attacker that are common to multiple failure scenarios are also documented in the form of attack trees. These are very informative as they can also be applied to attacks against the IEC 61850 substation. While the NESCOR failure scenarios and associated attack trees tend to give more high-level descriptions of possible attacks, they constitute a very useful resource for designing further cyberattacks.

Looking at vulnerabilities within the digital substation, we find much existing work. Many of the contributions do not consider the security measures offered by IEC 62351, and so the proposed attacks are often already addressed by that standard. Others focus on vulnerabilities or issues in IEC 61850 used with IEC 62351 explicitly. By doing so, they highlight some possible gaps in IEC 62351 and offer solutions to improve the standard. We summarize all these previous contributions in Table 3.1 and discuss them in detail in this section. We can see that most of the attacks discovered on IEC 61850 alone are addressed by IEC 62351. However, it is often argued that applying the latter can lead to performance problems.

3.1.1 Known Attacks on IEC 61850 Substations without IEC 62351

There is much previous work on the kinds of cyberattacks that threaten the IEC 61850 substation. Many of them are not specific to the substation or to power systems in general, while others are unique to environments where IEC 61850 protocols are found.

Rashid *et al.* have authored a survey compiling many known attacks against IEC 61850 [57]. These include both attacks against IT networks and attacks unique to IEC 61850 substations. The former are classified into three categories. DoS attacks can be carried out using TCP SYN requests, FTP or causing buffer overflows. Password cracking attacks can be carried out on active services in

Table 3.1: Vulnerabilities and issues in IEC 61850 and IEC 62351 identified in previous work

Vulnerability or issue	Addressed by IEC 62351	Proposed solution
Modification and replay on GOOSE/SV [57]	✓(RSA)	Use IEC 62351-6, but then performance is a problem [57]
MMS credential hijacking [58]	✓(TLS)	None suggested
Force GOOSE subscriptions (PIM) [58]		None suggested
MMS MitM attacks [59]	✓(TLS)	Use IEC 62351-3 and -4, if performance is not too affected [59]
Spoofing SV PDUs [60]	✓(RSA)	Algorithm pinpoints faulty IED based on SV measurements [60]
Spoofing GOOSE PDUs [61]	✓(RSA)	Use cryptography, inspect packets, apply measures to prevent attacks on Ethernet [61]
GOOSE poisoning attack [62]	✓(RSA)	None suggested
Weak cipher suites for TLS in IEC 62351-3 [63], [64]		Use secure cipher suites [63]
Performance issues if using TLS in IEC 62351-3 [64], [63], [65]		Reconsider use of TLS [64], use cipher suite with no encryption [65]
MMS MitM attacks (multi-hop scenario) [66], [67]		Use application-level secure session [66], use non-repudiation tokens [67]
Performance issues if using RSA in IEC 62351-6 [43], [44]		Use symmetric cryptography (HMACs) [43], [44], [68]
GOOSE delay and replay attack [42]		Track the last $stNum$ reset [42]
Redirecting SV PDUs [42]		Include sender MAC address to calculate $AuthenticationValue$ [42]
Weak SNTP security [42]		Use same signature mechanisms as ones used for GOOSE/SV [42]

IEDs such as FTP, Hypertext Transfer Protocol (HTTP) and Telnet. Packet sniffing attacks include Address Resolution Protocol (ARP) cache poisoning, Content Addressable Memory (CAM) table flooding or switch port stealing to allow an attacker to view Ethernet traffic meant for other hosts. The attacks that are unique to IEC 61850 target the GOOSE and SV protocols. We describe many of these attacks in more detail later in this chapter. Notably, these protocols are vulnerable to modification and replay attacks. They can also be used for flooding attacks. The attacks described in this contribution are possible against substations due to the reliance on existing protocols that have no security (FTP, HTTP and Telnet) and the lack of security on protocols unique to the substation. This work provides a good overview of the known attacks against substations.

Wright *et al.* describe two possible attacks against IEC 61850 protocols in their work [58]. This research deliberately excludes the use of IEC 62351's recommendations, as this reflects the case

where performance concerns dictate that measures like cryptography cannot be used. The first attack described is a credential hijacking attack against MMS. It is demonstrated using an automaton. By default, MMS provides an access control mechanism to establish sessions for authorized users, but it provides no protection for its PDUs, leaving them open to sniffing and alteration. The attack consists of sending a login request with the wrong credentials at the same time as a user sends their legitimate login request. The server receiving these requests responds with an error for the attacker's PDU and an acknowledgment for the user. However, the attacker can redirect the error PDU destined for her and forward it to the user. The latter is left with the impression that her credentials are wrong, and the session is hijacked. The second attack is an amplification attack based on Protocol Independent Multicast (PIM) multicast. GOOSE uses a publisher-subscriber model with Ethernet multicast [22]. The idea of this attack is to force extra LNs to subscribe to GOOSE publishers unnecessarily. The attacker does this by sending either a PIM-flood or PIM-graft packet, depending on the type of network. These requests instruct routers to add a specific interface (the target LN) as a receiver of multicast messages, even if it is not supposed to receive them. By forcing extra subscriptions, the proposed attack floods IEDs with useless traffic that they might not be able to handle. Given that the IEDs have limited resources and strict performance requirement, this can be enough for a successful DoS attack.

Kang *et al.* investigate potential attacks against IEC 61850 MMS in the context of Distributed Energy Resource (DER) [59]. This research focuses on MitM attacks against real photovoltaic (PV) inverters made possible by using ARP poisoning. In this scenario, the attacker has the capability to sniff, spoof, modify, and drop MMS PDUs. The attacks executed in this work are explicitly mapped to relevant NESCOR failure scenarios [55] specific to DER. Using MitM capabilities, the attacker alters the communications between an MMS client and an MMS server by spoofing an MMS write request to change the power limitation setting of the PV inverter to a false value. To prevent the PDUs sent in response to the malicious PDU from reaching the legitimate MMS client, the attacker drops them. The change in power limitation causes the PV inverter to go into standby mode rather than function as intended. This work highlights the security flaws of MMS and provides an example of leveraging the NESCOR scenarios to elaborate more detailed cyberattacks. However, while there is mention of IEC 62351-4 and how it improves the security of MMS, the experiments described in

this work do not apply this specification. We know from the standard [41] that doing so theoretically prevents the MitM attacks described.

Research by Valdes *et al.* documents two false data injection attacks against the SV protocol [60]. The first attack aims to insert false SV PDUs with the goal of indicating a fault when there is none. This has the potential to cause unnecessary downtime. The second works the same way, but the false PDUs instead hide actual faults. Hiding faults has the capability to cause much damage due to its potential to cause another, upstream relay to trip a CB and consequently cause a bigger outage than expected. Both attacks are good examples of how an attacker can misuse SV PDUs to cause physical damage to a substation. The work proposes an approach to detect these attacks using collaborating IEDs. Using each other's measurements and Kirchhoff's laws, they can pinpoint the source of invalid measurements. The approach is tested in MATLAB and shows that it can locate faulty measurements and IEDs. Such an approach is very different from using NSM for monitoring, as done in this thesis, since it relies on using the measurements themselves to detect cyberattacks.

Hoyos *et al.* describe a possible GOOSE spoofing attack in their contribution [61]. This attack works by sniffing existing GOOSE traffic to locate the current value of $stNum$, $sqNum$ and $allData$. Then, injecting a forged GOOSE PDU with an altered value for $allData$ causes some change of state in the system. Specifically, the attacker modifies $allData$ by flipping all the bits in it that represent Boolean values. In addition to changing $allData$, the $stNum$ must be incremented from the $stNum$ value seen in the sniffed traffic and $sqNum$ must be set to 0. These changes are necessary to ensure the malicious PDU is treated as a valid state change by the subscriber. The attacker might not know the meaning of the bits in $allData$, because this information is not found in the PDUs themselves. Despite this, the attack is likely to cause unwanted effects by changing the state of some LNs. As an example, this state could be whether a CB is open or closed. A similar idea is described by Kush *et al.* in their research presenting the GOOSE poisoning attack [62]. It is carried out the same way as the spoofing attack, with a small change. In this attack and its variants, the attacker increases the $stNum$ of malicious PDU to make it higher than what the legitimate publisher is currently using and sets $sqNum$ to 0. Note that the malicious PDU contains the same $allData$ as legitimate PDUs. Doing this allows the attacker to hijack the GOOSE communications entirely. The GOOSE poisoning attack exploits unique behavior of GOOSE where the subscriber

drops PDUs with a lower $stNum$ field than what it received last, as described in IEC 62351-6 [28] and discussed previously in Section 2.5.4. This results in an effective DoS attack as the legitimate publisher's PDUs (and thus critical commands) are dropped. These attacks show how a single malicious GOOSE PDU can have significant impact on the system. However, both rely on the lack of integrity checking in GOOSE PDUs, making it likely they can both be addressed with a similar method.

3.1.2 Security Evaluation of IEC 62351

Schlegel *et al.* provide an overview of IEC 62351 and the parts of it released up to 2015, as well as an assessment of each part when considering the conclusions of existing work [69]. It is a very useful resource to initially learn about the IEC 62351 standard and a recommended read for whoever is interested in this subject. Overall, the assessment concludes that IEC 62351 does much to provide security to power systems. It does point out a few gaps, several of which are discussed in more detail later in this section. In general, the assessment finds that potential attack vectors stem mostly from allowing configurations that enable backwards compatibility but reduce security in the process. As it was published in 2015, this assessment does not provide up-to-date information on IEC 62351-7:2017 or IEC 62351-9, as they were not published at the time.

IEC 62351-3 and IEC 62351-4: Security for TCP and MMS

Because IEC 62351-4 aims to secure MMS and refers to IEC 62351-3 to secure TCP, the transport layer of MMS, these two specific parts of the standard tend to be evaluated simultaneously in past work on secure MMS.

Khaled *et al.* study the effects on performance when applying the specifications of IEC 62351-3 and IEC 62351-4 [63]. This work provides an analysis of the 10 cipher suites specified by the standard. It recommends replacing some of them to include the use of Diffie-Hellman Ephemeral (DHE), RSA and Cipher Block Chaining (CBC), as they provide additional security benefits. To evaluate the performance of these cipher suites, experiments are done using open source implementation OpenIEC61850 and Java's Secure Sockets Layer (SSL) sockets. These can be used to support MMS communications over TLS as per the IEC standards. Several metrics are gathered including

handshake latency, request latency, CPU overhead and memory overhead that occur during an MMS Get request. The conclusion of this work is that TLS itself increases latency by 75%, but that the overall performance still suits the requirements for IEC 61850 message types 2, 3, and 5, which are the types carried by MMS. Different cipher suites affect performance to varying degrees, yet they all satisfy the requirements, so it is recommended to use the suites proposed in the study as they provide more reliable security. This is a useful study as it provides an assessment of the cipher suites recommended in IEC 62351-4 and show that there is potential in allowing use of better cipher suites. However, it is unclear if the results apply in actual substations. The authors include a mention that the model used does not accurately represent such an environment. There is also an assumption that IEDs would provide hardware support for AES like the CPUs used in the study.

Wright *et al.* provide an analysis of IEC 62351-3 and point out some of its flaws, mainly concerning key management and the use of weaker cryptography [64]. The standard specifies the use of TLS for protocols using TCP. Use of TLS necessary implies the use of certificates and everything they involve, such as checking for certificates that are expired or revoked and certificate authorities. There is no mention of how these requirements are to be implemented in actual networks, as key management is instead left to IEC 62351-9, which was not published at the time. This work explains that whether or not the trust architecture used relies on Certificate Revocation Lists (CRLs) or the Online Certificate Status Protocol (OCSP) protocol, there exist potential attacks against these architectures that must be accounted for. Additionally, IEC 62351-3 allows the use of weak cipher suites to allow backward compatibility [40], such as ones using Rivest Cipher 4 (RC4) or MD5. According to this research, using a downgrade attack, an attacker could trick a device into using one of those weak cipher suites to execute known attacks against them [64]. This work by Wright *et al.* sheds light on potential improvements to IEC 62351-3.

Chowdhury *et al.* attempt to implement the use of TLS in MMS as recommended by IEC 62351-4 in legacy environments [65]. Since that part of the standard refers to IEC 62351-3 to secure the transport layer [41], their work also implicitly evaluates IEC 62351-3. One of the objectives of this research is to measure the performance of MMS with TLS in a low-resource embedded systems, as would be expected in a power system. The concern is that TLS involves costly operations to handle certificates, sessions, digital signatures and encryption that might not be feasible for such

systems. The implementation of TLS in MMS in this work relies on OpenSSL and the real-time Operating System (OS) VxWorks. The results indicate that using encryption in TLS increases the memory usage by 85% and greatly increased duration of read and write operations. Performance is improved if using a cipher suite that only includes MAC and foregoes encryption, considered less critical in the power system context. The SSL handshake could also take up to 3 seconds, which could affect certain applications. This work provides insight into the use of TLS in a more realistic power system context and the potential ramifications on performance when using encryption. It does not cover the security of the A-Profile as specified by IEC 62351-4 [41] as it is out of its scope.

Fries *et al.* [66] and Ruland *et al.* [67] discuss a weakness in the security of the MMS A-Profile as specified by IEC 62351-4 . In typical situations, the T-Profile and A-Profile security are both used for MMS communications between two end devices, providing authentication, confidentiality and integrity of PDUs [41]. However, both studies point out that there are several use cases involving a multi-hop connection. In other words, the communicating parties do not communicate directly with each other, instead requiring an intermediate proxy to forward their traffic. The security of the T-Profile, which relies on TLS, can only be provided for direct communications between the end devices and the proxy. The T-Profile alone cannot secure the communication between the end devices in this scenario. This leaves only the A-Profile security to protect communications between the two endpoints. As explained in the work by Fries *et al.* [66], the A-Profile security only provides authentication during connection establishment and not for subsequent messages. It additionally does not provide integrity. In the multi-hop scenario, if the proxy is malicious, it can easily carry a MitM attack and modify the PDUs it is supposed to forward. To remedy this, both works suggest several methods of establishing a cryptographic session during the connection establishment: the creation of a session key to provide application-level authentication and integrity of subsequent messages [66], or the use of tokens in PDUs to provide non-repudiation [67] . These works provide an argument to secure the T-Profile and A-Profile independently, instead of assuming T-Profile security is present, and discusses several use cases showing why this is needed in the context of the smart grid.

IEC 62351-6: Security for GOOSE and SV

Past research has criticized IEC 62351-6 in particular, as applying it in practice results in failing to meet the performance requirements of IEC 61850. As discussed in Chapter 2, part 6 requires that every SV and GOOSE PDU be signed using RSA. Fuloria *et al.* [43] and Hohlbaum *et al.* [44] investigate whether it is possible to use RSA signatures on every PDU while meeting the strict 4 ms requirement of certain GOOSE applications. The conclusion is that during 2010, prohibitively expensive hardware was needed to be barely able to compute a 1024-bit RSA signature in time. Because NIST recommends using at least 2048 bits for RSA signatures as of 2018 [11], the requirement is not realistic. Looking at these results, the implication is that unless IEC 62351-6 is updated, it cannot be expected to be fully applied in practice. If one decides to implement the other recommendations of IEC 62351-6 without the RSA signature, more problems arise. The recommendations designed to prevent replay attacks leave the system open to the GOOSE spoofing attack described by Hoyos *et al.* [61] and the GOOSE poisoning attack described by Kush *et al.* [62]. Without the integrity check on PDU contents without the RSA signature, an attacker can spoof or modify GOOSE PDUs easily. As a potential solution, experiments done by Weerathunga [68] demonstrate that using HMACs rather than RSA digital signatures for integrity checking results in satisfactory performance. The main drawback of this proposition is that HMACs cannot provide non-repudiation, because they require that the sender and receiver share the same key. The latter requirement also renders key management more complicated.

By analyzing the specification, Strobel *et al.* [42] uncover three weaknesses, two of them found in IEC 62351-6. The first is a replay attack on the GOOSE protocol. It allows an attacker to recreate the same effect as a GOOSE poisoning attack [62] without the requirement of spoofing PDUs. This is important because spoofing is unavailable to the attacker when the integrity measures of IEC 62351-6 are applied, yet replay remains possible. To execute this attack, the attacker replays a legitimate message with a high $stNum$ closely after the $stNum$ is set back to 0 by normal conditions. The $stNum$ counter can overflow and return to 0 on its own as it is only 32 bits long. It also resets if there is absence of GOOSE traffic for longer than the TAL parameter of the last PDU. The attacker can therefore delay GOOSE PDUs to deliberately cause a $stNum$ reset, then replay

a PDU captured shortly before the reset. This cause the subscriber's $stNum$ to become higher than the publisher's, exactly like in a GOOSE poisoning attack. The second vulnerability involves replaying an SV PDU meant for one subscriber to a different subscriber. This is possible if the two subscribers are subscribed to the same dataset coming from a logical node but expect a different data rate. In a regular SV PDU, there is no information that specifies the subscriber that should receive the PDU, It therefore is possible to replay a PDU with a higher $smpCnt$ to a subscriber that is currently at a lower $smpCnt$ due to having a lower data rate. The last weakness described by the work of Strobel *et al.* concerns the SNTP protocol, used in IEC 61850 for time synchronization. The authentication schema of SNTP is considered weak as it relies on the insecure DES algorithm for encryption. It also uses a shared key, meaning it is leaked if any device using it is compromised.

3.2 Automated Protocol Analysis

Part of our work involves analyzing the GOOSE and SV protocols to derive potential cyberattacks. Several approaches have been suggested in the literature in order to automatically analyze communication protocols, but we find that these approaches do not fit our specific use case. Existing implementations [70], [62] and models of IEC 61850 protocols tend to not apply all specifications of IEC 61850 and IEC 62351, which we are interested in. An analysis of them could therefore be incomplete. However, we do note that the approaches can be used to complement our own, and as such we document them here.

3.2.1 Fuzz Testing

According to Open Web Application Security Project (OWASP), fuzzing involves automatically injecting malformed data into an application (treated as a black box) in an attempt to locate bugs [71]. This technique is very useful given that it does not require knowledge of the application's internals or source code. According to Synopsys, ICS protocols such as IEC 61850 MMS, implementations of IEC 60870-5-104 and Modbus are among the most vulnerable of all protocols when subjected to their fuzzing tool [72]. Additionally, work by Yang *et al.* [73] details the results of fuzz testing against various IEDs in a IEC 61850 testbed. The fuzzing simulator connects to the IEDs

as an IEC 61850 client, records some of the normal traffic, and then sends malformed MMS PDUs based on the ones recorded previously. For each malicious PDU, the simulator inspects the state of the IED to determine if it reacts negatively to the input (crashing, loss of communication, etc.). Overall, this work finds that fuzzing detects vulnerabilities in most of the IEDs, which supports the results of Synopsys. However, fuzzing can detect both errors that are innate to the protocol and errors that are exclusive to a specific implementation. This makes it difficult for us to determine whether these results reflect bad ICS protocol design or simply implementation errors. We can still conclude that ICS protocols tend to contain several vulnerabilities in practice, and thus can benefit from fuzzing techniques and further analysis. In our case, we cannot apply fuzz testing. Existing implementations tend to omit the specifications of the cybersecurity standard IEC 62351-6, as verified by the experiments of El Hariri *et al.* [70]. We therefore do not have a "correct" implementation of these protocols to speak of, and as such, we cannot use techniques that rely on one. We must rely on the text from the standards until such an implementation is found.

3.2.2 Formal Methods

A different technique used to find vulnerabilities is using formal methods. These differ from fuzz testing, as they do not rely on an implementation acting as a black box. Instead, formal models aim to replicate the behavior of a protocol and can be subjected to formal analysis to verify for correctness. Such a model is found in the work of Amoah *et al.* [74], where the DNP3 Secure Authentication (DNP3-SA) protocol is modeled using Colored Petri Nets (CPNs). DNP3-SA can be described as the DNP3 protocol (a derivative of IEC 60870-104-5) that applies the recommendations of IEC 62351-5 [75]. Note that neither of these standards are part of the scope of this thesis. The formal model described in this work reconstructs the non-agressive challenge-response feature of DNP3-SA (in contrast to DNP3-SA's aggressive mode) using CPN features such as places, transitions and arcs. Then, analysis of the model can be done using its state space and Strongly Connected Components (SCC) graph to test the validity of the model including reachability, lack of loops, and lack of dead transitions. By analyzing the model when considering a regular scenario and an attack scenario, this work discovered a potential DoS attack against DNP3-SA. The attack requires the presence of an attacker that can modify requests to change them from critical to non-critical.

A further work by Amoah *et al.* [76] extends this model to also represent DNP3-SA's aggressive mode. This leads to discovery of yet another flaw that can be exploited when an attacker switches between the non-aggressive and aggressive modes. However, the attack described is stated to be non-applicable in a later work by Cremers *et al.* [77]. Formal models provide many benefits when designing and analyzing protocols as they eliminate ambiguity and allow for rigorous analysis that is very pertinent for critical applications that cannot afford to contain defects, such as aviation [78]. However, when it comes to IEC 61850 protocols, the existing models using Petri Nets (PNs) [79] and timed automata [80] do not include the IEC 62351 specifications. This renders them insufficient for analyzing these improved versions of the protocols. Without an existing model, it is challenging to elaborate a suitable one and validate that it behaves as intended, given the potential complexity of formal models for SCADA. As stated by Amoah *et al.*, the model created for DNP3-SA contains hundreds of nodes and over a thousand arcs [74].

3.3 Study of Network and System Management and IEC 62351-7

3.3.1 Design of Network and System Management Solution

The IEC 62351-7 standard [8] itself provides some guidelines of how NSM is to be added to existing communication networks for the purposes of security monitoring and ID, but it does not give many details. Its recommendations are shown in a figure entitled "Example of a power system SCADA architecture extended with NSM Data Objects" [8], which we show here in Figure 3.2. Some expected elements of the architecture can be gleaned from this example. For instance, there are three separate network segments: the control center, the substation and the Wide Area Network (WAN). A firewall and an IDS are located at every point where these segments are connected, in order to monitor traffic coming in and out of the segments. Additionally, nearly every device shown supports serving NSM DOs so that they can be monitored in a similar manner. It is unclear which device is meant to collect NSM DOs from other devices. Of note is that the security client is situated outside of the substation, while the security server is inside it.

In a similar vein to the above, work by Obregon [81] provides an overview of security monitoring guidelines according to relevant standards. It considers four aspects: network segmentation,

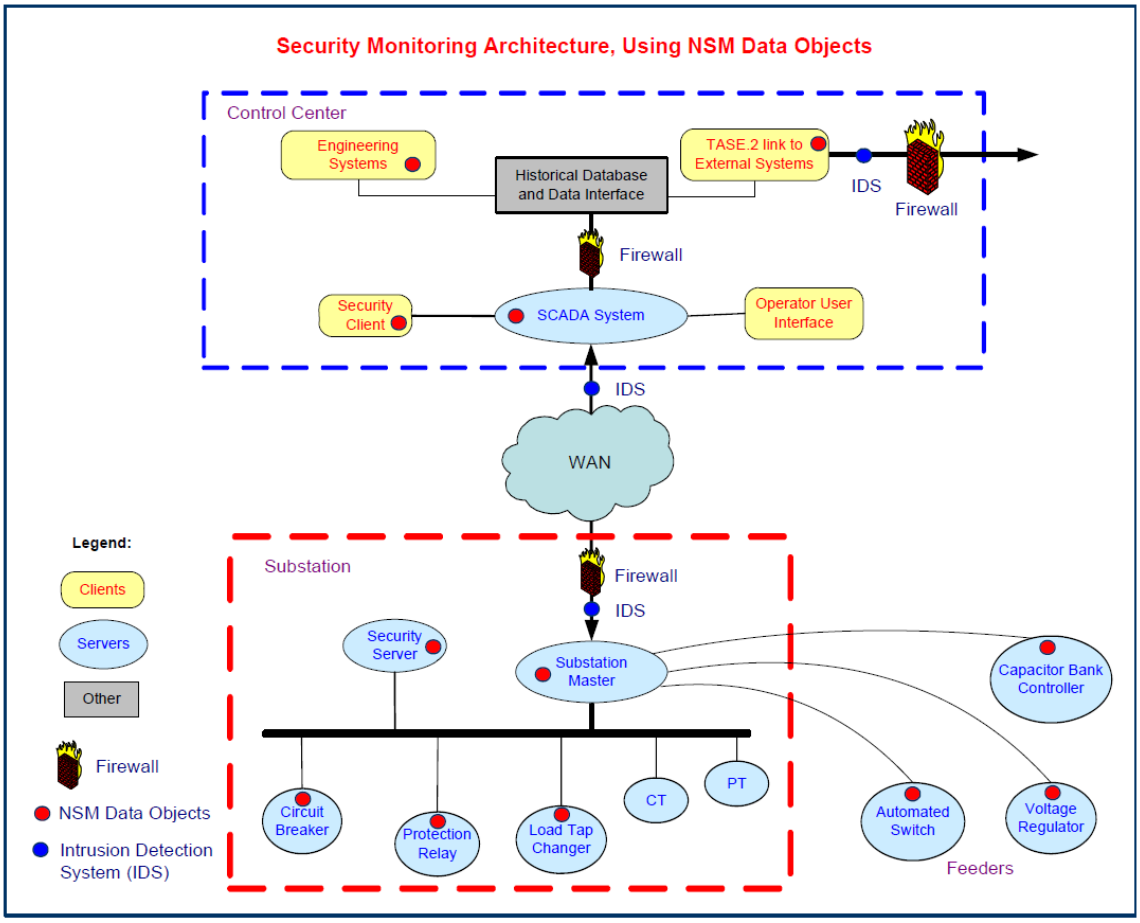

Figure 3.2: Security monitoring architecture with NSM according to IEC 62351-7 [8]

security event logging, ID and intrusion prevention, and packet capturing. While this research pertains to networks in general and not specifically to power systems, the guidelines remain applicable to this context. Much like IEC 62351-7, it recommends the use of network segments with security controls between them as well as the use of firewalls and IDS's between them for inter-zone monitoring. However, the recommendations found in this work differ on other aspects. It reserves a network segment, the management zone, specifically for the hosts performing the monitoring. It also mentions placing an IDS or packet sniffer within network segments as well, in order to monitor traffic traveling within the segment. This is pertinent as attacks can occur from within, but it might not be easily achieved in all substation communication networks, given that they can have various topologies [22]. In fact, some topologies do not include a central switch where all traffic

can be recorded. In that case, multiple IDS sensors or packet sniffers would be required to maximize visibility. Log management is also mentioned in Obregon's work. It is split into three tiers: log generation from monitored devices, log collection, and log monitoring. Due to the similarities between NSM DO and log collection, this structure might be applicable to the NSM architecture.

3.3.2 Implementations and Applications of Network and System Management

There exists previous work in implementing and applying NSM as per IEC 62351-7, but a large gap remains. As explained in Section 2.6.2, IEC 62351-7:2010 is rendered obsolete by the newer IEC 62351-7:2017. As of 2018, most of the previous work pertains to IEC 62351-7:2010. We show this in Figure 3.3, which presents an overview of the previous work on NSM. The new NSM DOs and their corresponding SNMP MIB definitions defined by IEC 62351-7:2017 have yet to be researched in detail.

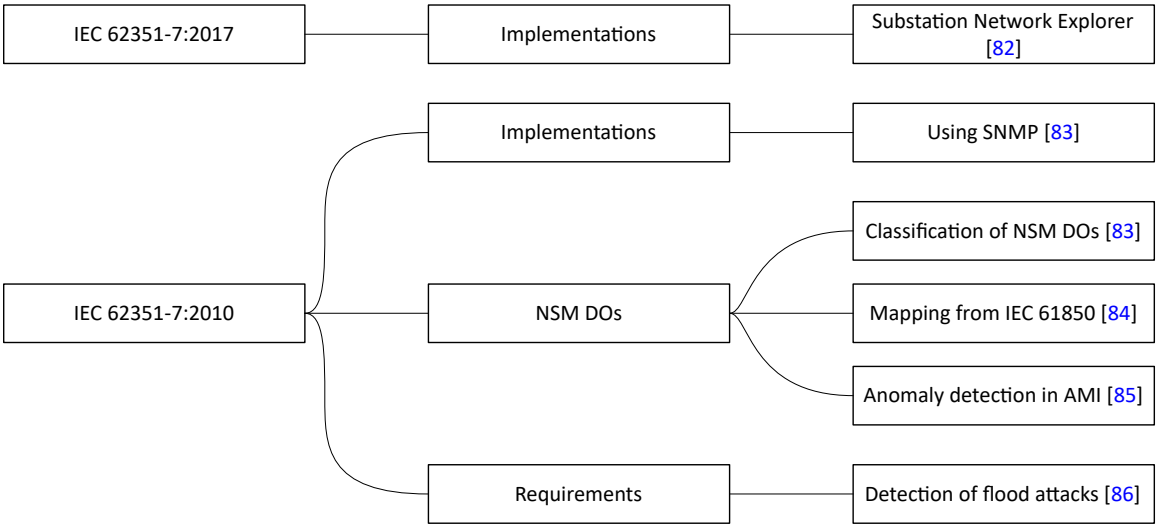

Figure 3.3: Overview of existing work done on NSM and IEC 62351-7

A 2014 report by the Electric Power Research Institute (EPRI) describes an implementation of NSM as per IEC 62351-7 in a system named Substation Network Explorer (SNE) [82]. The report differs from other previous work by using very similar NSM DOs to the ones defined by IEC 62351-7:2017 despite the latter not being published at the time. This is because the report is part of the development of IEC 62351-7:2017. The SNE is a web application for network monitoring, traffic analysis and visualization of a substation topology based on Substation Configuration Description

(SCD) files. It monitors a simulated substation network using SNMP and the MIBs of IEC 62351-7. However, because no devices on the market support the standard at the time of the report [82], an intermediate system named an SNMP proxy is used to convert proprietary MIBs and data from non-SNMP protocols into the standard's MIBs. The SNE is able to detect several problems in the substation, including resource exhaustion, power supply failures, lost or unexpected GOOSE PDUs, clock errors, and network storms. An important fact mentioned by the report is that IEDs at the time do not support IEC 62351-7 and do not provide the information necessary to create the NSM DOs [82]. This remains the case in 2018, based on our experience.

Kwon *et al.* analyze IEC 62351-7:2010 and briefly give details about their implementation of NSM, which uses SNMP [83]. This contribution classifies the 136 NSM DOs of IEC 62351-7:2010 into seven type classifications to make the standard easier to understand. As this classification is meant for the obsolete NSM DOs, details are omitted here. The implementation of NSM includes a few NSM DOs and uses the open-source NetSNMP library running on the Ubuntu operating system. The work mentions that specific NSM DOs can be implemented by leveraging values found in already standardized SNMP MIBs, such as TCP-MIB [48] and UDP-MIB [51]. For example, the abstract NSM DO ConnCnt could be populated using the value tcpMaxConn from TCP-MIB. Coincidentally, the revised IEC 62351-7:2017 requires the use of TCP-MIB and UDP-MIB themselves [45].

Kim *et al.* provide mappings to implement the NSM DOs of IEC 62351-7:2010 by extracting existing information from IEC 61850 protocols [84]. This extraction can be performed internally by the monitored device itself or externally by a separate one, though the former is preferred for accuracy. By using only the information and services provided by IEC 61850, this work documents mappings to create 21% of the NSM DOs of IEC 62351-7:2010 without the use of additional hardware. Given the major differences between the editions of the standard, it is unlikely that the mappings found still work with the NSM DOs of IEC 62351-7:2017. Despite this, the method used to create NSM DOs from an already existing protocol is very relevant when implementing NSM DOs in IEDs or proxies.

Ju *et al.* investigate the possibility of using the NSM DOs of IEC 62351-7:2010 for anomaly detection in AMI [85]. This research starts by defining 33 NSM DOs considered as needed to properly

monitor the AMI environment and removing 10 NSM DOs from the standard due to redundancy. In total, the number of NSM DOs is increased from 136 to 159. With these new objects, the work describes how to leverage the NSM DOs in rule-based detection to detect two attacks. The first attack involves a Data Concentrator Unit (DCU) infected with malware that can collect data from smart meters and send disconnect commands. The second attack is one where hardware is attached to a smart meter, allowing for a later remote disconnection. The results of this work demonstrate the potential of NSM DOs in enabling detection of cyber attacks, though it is unclear how to elaborate the detection logic to do so. The NSM DOs proposed in this work for AMI are likely not applicable today due to the change of NSM DOs that occurred between IEC 62351-7:2010 and IEC 62351-7:2017.

Choi *et al.* [86] use ID to detect two types of DoS attacks explicitly mentioned by IEC 62351-7: a TCP SYN flood attack and a buffer overflow attack. The work selects relevant data attributes for each attack, collects data for these attributes, and uses it with data mining algorithms to find the best one for ID purposes. While this work refers to NSM described in IEC 62351-7 and claims to analyze the NSM DOs, it only uses the standard as a reference for the types of attacks to be handled by NSM. It therefore does not provide input on the use of NSM DOs and instead provides an example of how to fulfill the ID requirements of IEC 62351-7. Additionally, it is mentioned that both DoS attacks are executed using the GOOSE protocol. As explained in Section 2.3.3, GOOSE runs over Ethernet, not TCP/IP, so it is not clear how this can be done. There exists a protocol named Routable GOOSE (R-GOOSE) that allows GOOSE to work over IP [87], but it uses UDP instead of TCP. As such, it is not clear if the DoS attacks shown in this work are realistic. However, SYN flood attacks are definitely possible by abusing other existing protocols such as MMS or FTP.

3.4 Smart Grid Models and Testbeds

To evaluate the cybersecurity provided by NSM, we perform tests using a co-simulation smart grid security testbed. Previous literature contains many approaches that aim to either simulate or emulate the behavior of substations.

3.4.1 Network Simulation Tools

Past research has used OPNET Modeler[1] [88] to simulate the substation communication network. This software is a Local Area Network (LAN) simulation tool that can be used to evaluate network design and performance. Zhang *et al.* [21] use it to analyze the data flow within a IEC 61850 substation's Virtual LAN (VLAN)-based communication network. The substation used for the analysis is the D2-1 substation from IEC 61850-5 [20], shown earlier in Figure 2.1, and also includes a communication network, which we show as an example in Figure 2.2. Using this tool, three theoretical data flow models are elaborated: cyclic data flow (regular traffic like the streams of SV PDUs and of GOOSE PDUs in steady-state), stochastic data flow (event-driven messages like GOOSE trip signals or MMS PDUs) and burst data flow (GOOSE PDUs on state changes). The models are then used to evaluate the performance of the communication network given the traffic expected in an IEC 61850 substation. The research finds that VLANs improve performance by limiting overall data flows when the VLANs are configured correctly. Additionally, events such as system faults lead to an increase in network traffic (and as such, lowered performance). The results also suggest that among the network topologies outlined in IEC 61850-90-4 [22], a substation communication network using a ring topology performs better overall than one with a star topology.

Similarly, Ali *et al.* [89] use OPNET Modeler to validate their proposed process bus architecture for IEC 61850-9-2 [26] (concerned with SV). The latter is evaluated using Reliability Block Diagrams (RBDs) in addition to the simulation. It aims to improve reliability and delay performance by including dual-homing and Ethernet switches configured to use VLANs and Rapid Spanning Tree Protocol (RSTP). The proposed architecture differs from IEC 61850 guidelines by not including the use of Parallel Redundancy Protocol (PRP) or High-availability Seamless Redundancy (HSR) [22]. These are used to provide redundancy and increased reliability to ring networks, but the work considers them costly and complex [89].

Golshani *et al.* [90] use the Discrete Event Simulation (DES) tool OMNeT++ [91] with the INET framework to simulate a substation communication network. The substation in question uses a star topology without redundancy that includes one transformer bay and two feeder bays. These contain devices such as MUs, P&C IEDs, breaker IEDs, and Phasor Measurement Units (PMUs). The

[1] OPNET Modeler has since been renamed to Riverbed Modeler.

main contribution of this research is evaluating the end-to-end delays of SV and GOOSE to validate that they suit the performance requirements for substation communications. Additionally, this work compares PMU traffic that relies on UDP or IEC 61850 (Ethernet) protocols. PMU communications are found to have less delay when relying on IEC 61850 rather than the typical UDP used for this purpose, though this result is mentioned to be preliminary. The conclusion reached in this study is otherwise similar to the other works presented in this section, while relying on a different simulation tool.

The work presented previously is very pertinent in understanding the typical network activity in a digital substation. In this thesis, we are less concerned with the choice of network topology to use for the communication network. It should be noted that the approaches suggested previously do not include the use of actual IEDs or power systems. This leads to some limitations, such as making it difficult to study the physical impact of a given cyberattack.

3.4.2 Co-simulation Testbeds

The work of Hahn *et al.* [92] provides a good overview of existing SCADA co-simulation testbeds. These aim to combine SCADA with emulation and simulation tools to create realistic power systems for testing. The testbed described in this work is the PowerCyber testbed of Iowa State University. It includes a control center that connects to various substations (real and virtual) over a WAN through their RTUs using the DNP3 protocol. The substations themselves rely on GOOSE and MMS for communications between themselves and with RTUs. The IP network carrying this traffic is emulated using the Internet-Scale Event and Attack Generation Environment (ISEAGE) testbed, found at the same university. The testbed can be connected to two power system simulators, namely PowerFactory by DIgSILENT [93] and Real Time Digital Simulator (RTDS) [94]. The main difference between them is that the former does not work in real-time, but allows for simulating larger systems and has better analysis capabilities according to Hahn *et al.* [92]. PowerCyber is used to test some attacks that have an impact on the physical equipment, including coordinated attacks that aim to disrupt voltage stability. This research mentions several other testbeds that rely on different technologies. The National SCADA Testbed (NSTB) recreates a physical system out of real grid components and software, which proves effective but also costly.

Other co-simulation testbeds like PowerCyber also integrate emulation and simulation tools, though they rely on other technologies. These are the PowerWorld power system simulator [95] and OPNET System-in-the-Loop, the latter of which enables OPNET Modeler to connect to live devices. They are used to conduct tests in different smart grid contexts, rather than just the substation. The work of Yang *et al.* [73], mentioned previously in Section 3.2.1 on fuzz testing, also relies on a testbed that makes use of RTDS. The exact technologies used to create the network in this work are not stated explicitly.

Lo *et al.* [96] discuss the possibility of virtualizing IEC 61850 networks to take advantage of the benefits of cloud technologies. Rather than use physical IEDs, they consider the idea of using Virtual Machines (VMs) to implement the Logical Devices and functions in IEC 61850, removing the coupling between the physical hardware and the required functions. A cloud solution allows for easy configuration and management of the infrastructure and for the allocation of VMs as needed. To test the idea, a case study is done to replicate an existing power system and replace it with a solution based in VMware. The conclusion is that it is possible to map all required functionality of IEC 61850 to a virtual infrastructure. Testing is still required to evaluate the performance and potential issues that could arise with such a change. In our work, we make use of cloud technologies, as we rely on OpenStack and its VMs to implement the network along with several components in our testbed (namely the NSM agent proxies described in Chapter 4). However, we do not make all components virtual. Of note is how the infrastructure proposed by Lo *et al.* [96] makes use of UDP for all network communications rather than the expected IEC 61850 TCP/IP stack. Similarly, in our own setup, we converted some of the communication protocols based on Ethernet to UDP to ensure compatibility with our emulated network.

A more recent work by Wjtowicz *et al.* [97] describes a single physical server that uses VMware to perform various tasks of IEC 61850 by leveraging 13 VMs and 3 virtual switches. Doing so avoids the need to deploy dozens of IEDs within a substation. The system is used to simulate connections between an IEC 61850 client and server (using the Hammer and Anvil applications respectively) and for testing time synchronization using SNTP. The experiments reveal that there are some issues with reception of GOOSE PDUs and potential memory problems. These might be caused by the Hammer and Anvil software. Outside of these issues, GOOSE communications work within the

performance requirements of IEC 61850. Time synchronization is done within the physical server using SNTP, which does not match the recommendation of IEC 61850 to use the more precise PTP for time synchronization at the process bus [22].

3.5 Intrusion Detection Techniques

In this thesis, we are interested in the security provided by the implementation of NSM DOs of IEC 62351-7 using SNMP. To our knowledge, there is little to no past work on detecting cyberattacks using NSM and the NSM DOs of IEC 62351-7:2017 specifically. Nonetheless, several detection techniques that are proposed in existing literature could be potentially adapted to the context of an IEC 61850 substation monitored by NSM. An overview of them is shown in Table 3.2. Overall, detection techniques have been successfully used with SNMP traffic and with IEC 61850 traffic. Making use of the NSM DOs, in the form of SNMP MIBs, would represent a combination of these ideas.

3.5.1 Detection Using Simple Network Management Protocol

One of the recommended protocols to implement NSM DOs is SNMP [8]. When using this protocol, the NSM DOs are implemented as SNMP MIBs, much like other existing MIBs such as TCP-MIB [48]. In the literature, we can find research that proposes attack detection mechanisms based on data from SNMP MIBs. None of them apply these techniques to the smart grid context or rely on the NSM DOs of IEC 62351-7. However, the approaches proposed and the choices of MIBs variables selected for analysis can potentially be integrated with the NSM DOs.

Yu *et al.* propose a mechanism to detect flooding attacks based on data collected using SNMP and machine learning with Support Vector Machine (SVM) [98]. This work suggests that IDS's can benefit from information available in SNMP MIBs, but that there is insufficient integration between the two to enable this to happen. In an attempt to rectify this, the proposed mechanism works by gathering SNMP data and using select objects found within SNMP MIBs to classify flooding attacks using SVM. The polling rate of each device is optimized using an algorithm to ensure that the polling occurs slightly after the MIB values have been updated by the SNMP agent. On average,

Table 3.2: Comparison of previous work on detection techniques

Approach	Input	Strengths	Gaps
SVM [98]	SNMP MIBs	Can detect, classify 3 flooding attacks	No real-time detection, only tested with flooding attacks
C4.5 [99]	SNMP MIBs	Can detect, classify 3 flooding attacks in real-time	Only tested with flooding attacks
PIDC [100]	SNMP MIBs	Can detect, classify 3 DRDoS attacks	Only tested with DRDoS
P2P network and distributed data clustering using k-means [101]	SNMP MIBs	Improved availability. Can detect, classify 2 DoS, 1 DDoS, and 1 brute-force attack	Unknown if applicable to substation context
Whitelist, stateful analysis, anomaly detection [102]	IEC 61850 traffic	Can detect known, unknown attacks. Tested in substation environments [103]	
Snort signatures [104]	ARP, FTP, HTTP, ICMP, telnet traffic	Password cracking, DoS with ICMP, ARP sniffing	Not specific to substations
Signature-based, host and network detection [105]	IEC 61850 traffic, logs	Can monitor IEC 61850 packet contents. Can detect coordinated attacks	Network-based detection cannot detect unknown attacks
SVM [106]	MMS and GOOSE traffic	Can detect unknown attacks	Not tested against attacks
Signature-based and LSTM [107]	Modbus traffic	Can detect known, unknown attacks.	Not IEC 61850
Deep learning algorithms [108]	Power quality disturbances	Compares several machine learning algorithms incl. LSTM, CNN, GRU	Not IEC 61850, not tested against deliberate attacks

the polling rate in the experiments presented is 15 seconds. The 13 specific objects used for the analysis are selected using Correlation based Feature Selection (CFS). This approach is tested and yields positive results against three types of flooding attacks conducted in a testbed network: TCP-SYN flooding, UDP flooding, and Internet Control Message Protocol (ICMP) flooding. It should be noted that the quality of the results is influenced by the choice of the objects to use within the MIBs. Hence, care must be taken when selecting the relevant features for detection purposes. This work demonstrates the potential of using NSM DOs for attack detection, given that they can be implemented as SNMP MIBs, though it remains to be seen whether it can apply to attacks other than flooding. While the work does state how the SNMP data was collected and the polling frequency,

the analysis of the MIB data is only done after-the-fact rather than in real-time, making it difficult to use as-is for real-time detection. In a further work by Yu *et al.* [99], an improved system is proposed where the C4.5 algorithm is used instead of SVM. The system must be trained offline first, but can then be used for real-time detection. Additionally, using association rule mining (also done offline), this system can extract the rules used to classify flooding attacks into the three different types. This is a useful approach to understand the differences between the attacks and derive useful rules automatically.

In a similar vein to the above, Priya *et al.* propose a Protocol Independent Detection and Classification (PIDC) system to detect DRDoS attacks using SNMP MIB data [100]. The intention is to classify TCP and Domain Name System (DNS) DRDoS attacks respectively. The proposed algorithm collects information from SNMP MIBs and select 13 objects[2] related to TCP and DNS. With these objects, the Rank Correlation based Detection (RCD) algorithm is used to compare the traffic flows to distinguish normal traffic from potential attack traffic. Once an attack is suspected, it is classified using the C4.5 classification algorithm to identify whether it is a TCP or DNS flooding attack. Much like the work by Yu *et al.* [98], this research only attempts to detect one kind of attack, namely DRDoS.

A different approach for detection using SNMP is proposed by Cerroni *et al.* [101]. Like other work presented in this section, this detection mechanism uses SNMP data as its input. It differs mainly in that it uses a decentralized peer-to-peer (P2P) network and unsupervised distributed data clustering, unlike the other solutions that expect the presence of a central NMS. Monitoring stations are spread in the network and gather SNMP data for a section of the network. The stations then collectively analyze this data with a distributed data mining algorithm, allowing all peers to learn about the entire network. Similarly to the work by Yu *et al.* [98], CFS is used to select the most relevant MIB variables, resulting in a set of 14 variables. The data clustering algorithm used is based on the k-means algorithm. In the network configurations used for the tests, detection accuracy varies based on parameters such as the connectivity of the network, the number of attack classes known by each monitoring station, and the value of k, i.e. the number of clusters used. The algorithm

[2]The 13 objects used in [100] are different from the 13 used in [98]: the total number of variables being the same is a coincidence.

is simulated using the Java data mining framework WEKA [109] and tested against four attacks. These consist of a DoS attack, a Distributed Denial-of-Service (DDoS) attack, a DoS attack on Secure Shell (SSH) and a brute-force attack on SSH. Overall, the detection rates were comparable or better than other (centralized) detection methods. The work states that a decentralized solution offers better availability by avoiding single points of failure in the network. It is not known if this approach is applicable or beneficial to the substation context specifically, as it constitutes a relatively small part of the smart grid. The availability and robustness offered by this solution is certainly of interest to the smart grid context in general.

3.5.2 Detection Using IEC 61850 or Industrial Control Systems Traffic

Much of the literature focuses on ID within the IEC 61850. The approaches suggested differ from ones described previously, where SNMP MIBs are used as the input to the detection mechanism. Instead, the approaches discussed in this section rely on the IEC 61850 traffic and its characteristics. NSM is compatible with IDS's as the standard states that the NSM DOs are intended to support them [8], though it is not stated exactly how they do so.

A report by McLaughlin and the SPARKS project provides a summary of research on IDS in SCADA systems and proposes an architecture for a multi-layer IDS for IEC 61850 systems [102]. It should be noted this report is related to another work presented previously [59]. The system makes use of three techniques: a whitelist to only allow specific communications, a stateful-analysis approach and and one based on anomaly detection. These methods are intended to complement each other. Whitelisting is effective in SCADA environments as the networks tend to be very static, and it limits the options available to an attacker. Traffic that is allowed according to the whitelist is subjected to a stateful analysis based on rules, in the form of profiles, that aims to compare traffic to the normal behavior profile to detect anomalies. Anomaly detection is applied after all these checks to detect anomalies. This kind of detection must be trained beforehand and can be retrained regularly to allow it to adapt to new threats. The architecture described is meant to provide means to detect both known and unknown attacks in SCADA systems. An implementation of this architecture is described by Yang *et al.* [103]. This IDS in particular is claimed to have been tested within actual substation environments.

Premaratne *et al.* propose an anomaly detection using Snort [110] for IEC 61850 substations [104]. In this work, snort signatures are elaborated for three attack scenarios: password cracking, DoS using ICMP and ARP sniffing. The attacks presented are not specific to the substation as they are based on common communication protocols like ARP or ICMP. Since this thesis is more interested in threats unique to the substation, we consider this contribution a bit less applicable than other work, which leverages features more specific to the IEC 61850 protocols.

A signature-based anomaly detection method for GOOSE and SV is proposed by Hong *et al.* [105]. This work combines host-based anomaly detection with network-based detection. The former relies on information about hosts such as the number of login attempts, configuration changes, etc. The data to enable this is expected to be provided by the hosts in the form of logs. Network-based detection relies on GOOSE and SV traffic monitored using port mirroring. It applies a set of specific rules to determine whether the PDUs represent an anomaly or an attack. For example, the system monitors the count of PDUs sent and received and signals an anomaly if it is outside of the expected minimum and maximum thresholds. In some cases, it also monitors some specific fields in the PDUs, such as GOOSE's $sqNum$, as these are usually affected by known attacks on GOOSE. Multiple attacks are carried out on a testbed to validate the approach. The main downsides of this approach is that it relies on signatures, rendering it unable to detect unknown attacks that do not fit any signature, and its reliance on the hosts producing logs.

Yoo *et al.* [106] propose an anomaly detection mechanism based on the GOOSE and MMS traffic of a substation. This approach has the advantage of potentially detecting unknown attacks as it relies on anomaly detection rather than signatures. Due to the regular traffic expected to be found when using IEC 61850, it is expected that anomaly detection can perform better than in other contexts, where the false-positive rate can be very high [106]. In this work, MMS and GOOSE PDUs are initially collected in normal conditions and used as input to a 3-phase packet filtering module where they are processed. The module is responsible for filtering traffic other than the two protocols of interest, as well as extracting relevant features from the PDUs (by themselves and in groups). The extracted data is then subjected to an algorithm to perform outlier removal, such as Expectation Maximization (EM) and Local Outlier Factor (LOF), to remove data caused by noise or errors. Training is then performed using one-class SVM in order to learn the normal behavior

of the system. This system is tested with data from normal operation to test the false positive rate, found to be 97.8294% for MMS and 94.1239% for GOOSE. This work's limitations are that it does not conduct attacks, meaning data for false negative rates is unavailable, and it mentions that the false positive rate should be improved. Given the amount of GOOSE traffic that can be expected in a IEC 61850 substation, a lower false positive rate would indeed be a definite improvement.

Feng *et al.* [107] propose an anomaly detection method for ICS's that combines a signature-based approach with the use of a Long Short-Term Memory (LSTM) classifier. According to this research, the network configuration of an ICS tends to be sufficiently stable that one can prepare a signature database (in the form of a bloom filter) that represents all potential network packets that can be found in non-attack scenarios. Any packet with a signature that is not in this database is classified as an anomaly with this approach. The reliance on a bloom filter implies that this form of detection can have false positives, due to potential collisions in the hash functions used by the bloom filter, but not false negatives. In other words, it is not possible for a normal packet to be classified as anomalous. In addition to the signature-based detection, an LSTM neural network is used to detect attacks that depend on state (e.g. the previous packets). LSTM networks are used when dealing with multivariate time-series, as in this case, and aim to predict the behavior of the system assuming no anomalies occur. In this work, the LSTM predicts the signature of the next packet rather than the characteristics of the packet itself. To reduce the amount of false positives, noise is added to the training data. The proposed approach is tested using Modbus data from a SCADA gas pipeline system in both normal and attack scenarios. It shows better performance in most aspects than other existing approaches due to handling complex data better. The average time required to classify a packet is about 0.03 ms, which is acceptable performance in the context. The anomaly detection suggested in this work is pertinent to our work despite not being applied to IEC 61850 substations. The combination of signature-based detection with anomaly detection allows for better accuracy than either by itself.

Mohan *et al.* [108] apply various deep learning algorithms using TensorFlow [111] and Keras [112] to classify power quality disturbances in the smart grid. The algorithms considered are Convolution Neural Network (CNN), Recurrent Neural Network (RNN), Identity-Recurrent Neural Network (I-RNN), LSTM, Gated Recurrent Units (GRU), and Convolutional Neural Network-Long

Short-Term Memory (CNN-LSTM). The latter is a hybrid architecture proposed by this work and combines CNN with LSTM to offer better overall performance (with an accuracy of 0.984) than the other algorithms. While this work is concerned with unintentional disturbances rather than deliberate cyberattacks, the algorithms discussed could still be useful to our work.

Chapter 4

Network and System Management in the Digital Substation

In this chapter, we provide details on our design and implementation of Network and System Management (NSM), as per the IEC 62351-7 cybersecurity standard, in a substation conforming to IEC 61850.

4.1 Overview of Network and System Management and IEC 62351-7

In this section, we outline the objectives of NSM and the capabilities it provides to accomplish these objectives.

4.1.1 Objectives of IEC 62351-7

The main goal of IEC 62351-7 and NSM is to monitor the state of the communication infrastructure that supports the power infrastructure [8]. The standard states that the communication network tends to not be monitored sufficiently well in traditional power systems. This has historically led to major consequences, such as blackouts [8]. Yet, power systems increasingly rely on communication networks for their operation. Hence, it is imperative that operators have the capability of detecting network problems quickly enough to address them before they propagate [8]. There are existing solutions to enable monitoring of typical IT networks, such as the SNMP protocol. However, these

solutions must be adapted to the power system context to work well in that environment. The IEC 62351-7 standard exists to bridge this gap by defining the necessary NSM Data Objects (NSM DOs) to ensure adequate monitoring [8].

IEC 62351-7 defines the four functions of NSM as follows:

- "Monitoring the status of software applications, hardware equipment, and communications. [...]
- Monitoring the performance of systems and communications. [...]
- Intrusion detection. [...]
- Configuration management. [...]" [8]

NSM aims to address not only problems that are caused by accidents, but also ones caused by malicious actors [8]. The main capability NSM provides against such deliberate attacks is detection. This is in contrast with other security measures in the rest of the IEC 62351 standard, which focuses on prevention of attacks. Detection complements prevention, as it provides a warning when attackers somehow circumvent preventative security measures and enables operators to respond to the threat [8]. The intrusion detection capabilities of NSM are required to include the following [8]:

- Various DoS attacks, such as resource exhaustion, buffer overflows, and device shutdowns;
- Unauthorized attempts to access systems and networks;
- Integrity attacks, such as tampering with PDUs;
- Coordinated attacks.

4.1.2 Capabilities in IEC 61850 Substation

In order to fulfill the requirements outlined previously, NSM provides many capabilities in the form of its NSM Data Objects (NSM DOs) [8], which are designed with the functions of NSM in mind. A NSM DO is a data structure carrying a piece of information about the current state of a host. An example is the ConfigurationVersion object (from the IED package), which is a readable string of characters describing the "[v]ersion of the last uploaded configuration" [8]. Devices in

the network that support NSM DOs can therefore communicate valuable information about their current status. Events in the network can potentially cause changes in the state of the monitored hosts, resulting in corresponding changes in specific NSM DOs. By retrieving and analyzing the values in these NSM DOs, a monitoring system has an overview of the current state of the network.

Our focus is on aspects of the system unique to IEC 61850. In this context specifically, multiple packages defined by IEC 62351-7 contain NSM DOs that can be of use to monitor IEC 61850's application protocols. The ACSI package provides information on aspects of the IEC 61850 stack [8], without referring to a specific protocol. Upon closer examination, the NSM DOs for ACSI pertain mainly to the configuration of the substation's communication network. They mostly report the number of associations, subscriptions and publications in use. Packages GSE (including GOOSE) and SV pertain to their respective protocols, and the two share a great number of NSM DOs due to the similarities of the two protocols. To monitor MMS, two packages are of interest. The first is the common package that theoretically applies to all protocols [8]. In practice, it pertains mostly to statistics concerning the use of TCP, TLS and IEC 62351-3 (discussed in Section 2.5.2). The second is the MMS package, which contains the most NSM DOs among the IEC 61850 protocols. A discussion of the NSM DOs in all of these packages follows.

NSM DOs for GOOSE and SV The GOOSE and SV protocols are sufficiently similar that many NSM DOs are defined for both of them. For both protocols, there are different tables for publishers and subscribers that list all associations on the monitored device They are named GSEandSVPublisherAssociation and GSEandSVSubscriberAssociation. These NSM DOs compile statistics on the rate of PDUs, the number of buffer overflows or underflows, and the number of failures involving security keys for the association. In addition to these, several NSM DOs are provided to reflect unique aspects of GSE. Many of them refer implicitly to fields found in GOOSE messages, described earlier in Table 2.3, although does not always name them explicitly. Specifically, the NSM DOs track the current state of the $confRev$ and $ndsCom$ fields in PDUs, the number of TAL expirations that occurred, the number of received PDUs that are in error (meaning the PDUs are malformed, have bad parity or have a configuration mismatch [8]), and the number of unexpected PDUs received. The SV protocol has much fewer NSM DOs than other IEC 61850 protocols. In addition to the

NSM DOs it shares with GSE, it provides one additional NSM DO to track the number of PDUs received that did not have the correct length. However, this is stated to only apply to IP communications. This limits the capabilities of NSM in monitoring SV communications when compared to GSE. We summarize the sets of NSM DOs used to monitor GOOSE in Table 4.1.

Table 4.1: NSM capabilities for monitoring GOOSE

NSM capability	NSM DOs
Rate of PDUs sent or received per second	TxPduPerSecond, RxPduPerSecond
Count of integrity check or decryption failures	MessageIntegrityFailCnt, DecryptFailCnt
Whether $confRev$ in received PDU is unexpected	ConfRevMis[1]
Value of $ndsCom$ in received PDU	NdsComm[2]
Count of TAL expirations	TalExpCnt
Count of received PDUs in error	InErrCnt
Count of unexpectedly received multicast PDUs	InUnexpectedMulticast

We show the NSM DOs used to monitor SV in Table 4.2. Many of the objects are shared with those used for GOOSE.

Table 4.2: NSM capabilities for monitoring SV

NSM capability	NSM DOs
Rate of PDUs sent or received per second	TxPduPerSecond, RxPduPerSecond
Count of integrity check or decryption failures	MessageIntegrityFailCnt, DecryptFailCnt
Count of PDUs received with length error (IP only)	PDUSizeFail

NSM DOs for MMS The common package contains many NSM DOs to monitor the performance of TCP, which is the transport protocol used by MMS, and the optional TLS, introduced by IEC 62351-3. The MMS package is the largest package among the IEC 61850 protocols and provides information about the PDUs sent and received by the host. Overall, there is a great variety of NSM DOs available to monitor the MMS protocol: we show some of them in Table 4.3. NSM can track the various types of PDU being exchanged as well as reports, and monitor the use of security keys. Because IEC 62351-7's scope goes beyond IEC 61850, it should be noted that there are no NSM DOs related to features unique to IEC 61850 MMS.

In addition to the above, we should note that, assuming the use of SNMP when implementing NSM, existing MIBs can also be leveraged alongside the NSM DOs [8].

Table 4.3: NSM capabilities for monitoring MMS

NSM capability	NSM DOs
Whether IEC 62351-3's security (TLS) is used	IEC62351part3
Duration of TCP handshake	TCPHndShTime
Duration of TLS handshake, renegotiation and resumption	TLSHndTime, TLSRenegotiationTime, TLSResumptionTime
Which IEC 62351-4 security profile (TLS) is used	SecurityProfile
Rate of reports received	ReportPer100Seconds
Time between reports	RptReceptionDelay
Count of unacknowledged requests	MisCmdAckCnt
Count of PDUs sent and received, by type	ErrorRxCnt, ErrorTxCnt, RejectRxCnt, RejectTxCnt, ReqRxCnt, ReqTxCnt, RespRxCnt, RespTxCnt
Count of failed key negotiations	SessKeyFailCnt, TProfileSessKeyFailCnt, UpKeyFailCnt
Count of decryption failures (A- and T-Profiles)	AProfileDecryptFailCnt, DecryptFailCnt, TProfileDecryptFailCnt
Count and rate of session reestablishments	SessionEstablishmentRate, SessionRestartCnt

4.2 Design of Network and System Management

This design is a joint work with Mr. Mark Karanfil.

In this section, we describe our design used to include NSM in an IEC 61850 substation and how to leverage its capabilities to detect cyberattacks.

4.2.1 Protocol Selection

IEC 62351-7 states that NSM DOs can be implemented in any suitable protocol, one of which is SNMP. The standard provides working MIB definitions that map NSM DOs to SNMP MIBs, found on the IEC website [47]. Accordingly, this design uses SNMP as the protocol for NSM DOs and leverages these MIB definitions.

4.2.2 Addition of Components

We show the design of our NSM solution in Figure 4.1. To include NSM in an already existing substation, two key components are required: a set of NSM agents and the NSM manager.

NSM agent The NSM agent is responsible for producing the NSM DOs of IEC 62351-7, responding to queries sent by the NSM manager, and sending notifications to the NSM manager. The NSM

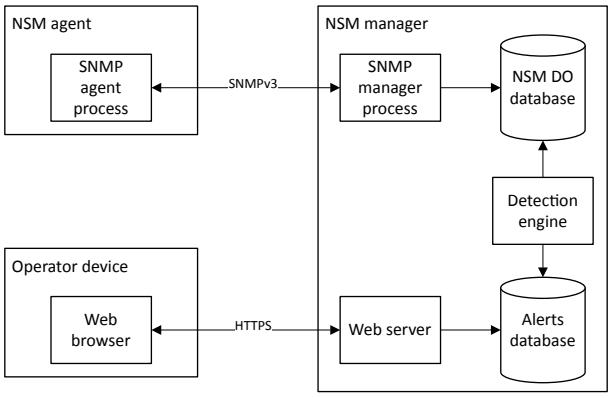

Figure 4.1: Overall design of NSM

agent can be a simple application running on a monitored host, a set of applications, or it can be a full-fledged device itself. This is the case when the monitored host cannot support the NSM DOs. We refer to it as a proxy in those instances. In Figure 4.1, the NSM agent is its own device and runs an SNMP agent process to communicate using SNMPv3. Note that we only show one NSM agent in the figure, but there are actually several of them.

NSM manager The NSM manager is the central location where NSM DOs and notifications from all the NSM agents are collected and analyzed. It runs an SNMP manager process to communicate with all NSM agents and receive their SNMP traps. The raw NSM DOs are placed in an NSM DO database. They must then be retrieved by a detection engine capable of analyzing them. If the engine finds an anomaly, it generates and saves an alert that is accessible to operators through a web application. We opt for web technologies because they do not require installation of a specific client. Users can connect to the application using any compatible web browser.

We show in Figure 4.2 how one could add the above components to the example substation shown previously in Figure 2.3. As can be seen, the NSM agents, represented by small squares with a red border, must be added to all the nodes within the substation communication network itself, while the NSM manager is expected to be in a separate security domain or zone. Using its communication link to the substation, the NSM manager is able to communicate to all NSM agents with SNMP. All other types of network traffic can be filtered using a firewall to reduce the exposure of the NSM manager itself to attackers. It is expected that the NSM manager queries the agents

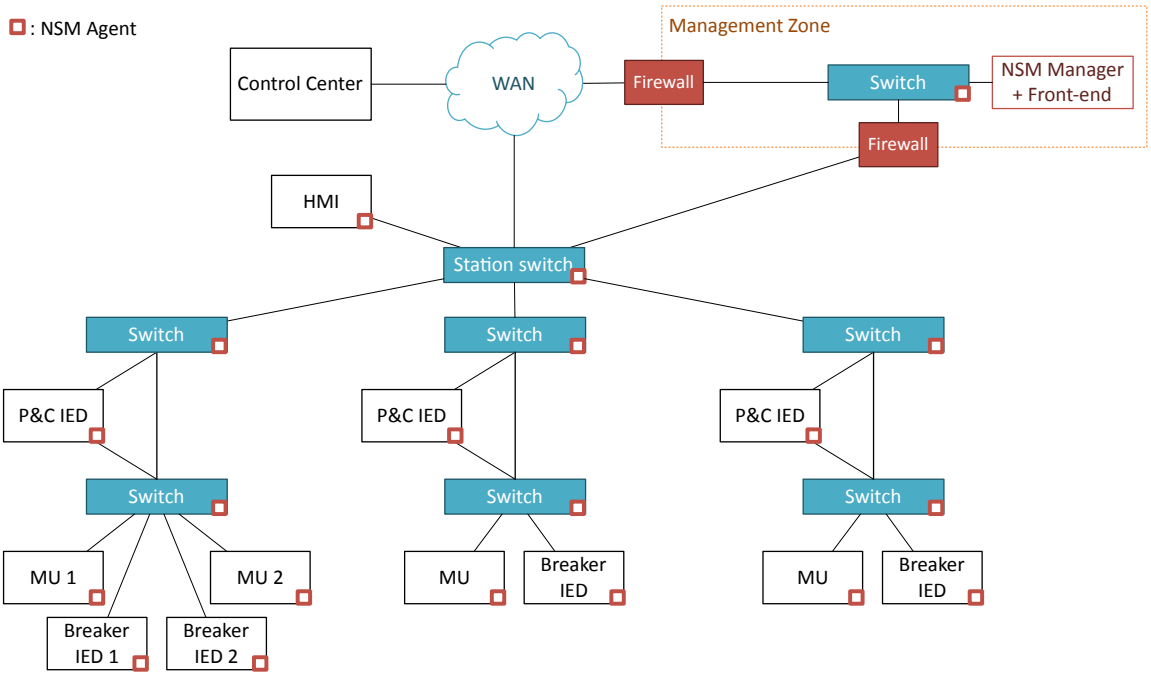

Figure 4.2: Example communication network for simplified substation D2-1 with NSM components

periodically to monitor their status and analyzes this data in real-time to detect any issues.

Methodology to Detect Attacks

IEC 62351-7 defines the NSM DOs, but does not specify how to react to changes in these objects. This is considered an implementation-specific issue [8]. Yet, in order to detect threats and mitigate them using NSM, it is not sufficient to inspect the values found in the attributes of NSM DOs. They represent the state of the system, rather than only reporting anomalous behavior. Distinguishing between the normal state of the substation and its state when under attack requires a more thorough analysis of relevant NSM DOs. We propose a general approach to detect attacks using the NSM DOs and their capabilities.

Consider a network with several devices forming the substation communication infrastructure, along with one host acting as the NSM manager. Devices to be monitored are NSM agents. Each agent i in the network monitored through NSM maintains its own set of NSM DOs M_i. The first agent maintains the set O_1, the second maintains O_2, and so on. The exact NSM DOs found in O_1 and O_2 might differ. Agents update their NSM DOs with new values for their attributes to

reflect the current state of the agent. If any event occurs, it is possible that a subset of attributes of the NSM DOs in $O_{i,t}$ are affected as a result. Identifying this subset can be done through manual analysis or by recreating the event and detecting the NSM DOs that were affected by it.

When agents update their NSM DOs, the previous values in their attributes are lost. The agents therefore are unable to track changes in their own state over time. To keep such a record, the NSM manager periodically polls agents. On every poll, the manager saves the set of NSM DOs it receives and notes the time at which it was obtained. As a result, the NSM manager saves multiple sets per agent that differ based on their values and their timestamp t. The set for agent i at time t can be expressed as $O_{i,t}$. At $t = 0$, the manager collects the sets $O_{1,0}$ and $O_{2,0}$ by agents 1 and 2 respectively, and repeats this for increasing values of t. This process is repeated continuously as the system operates. This provides the manager with information about the state of all agents, while also considering the time dimension.

The NSM manager can perform many operations on the sets of NSM DOs it collects. In some cases, it can detect attack signatures by comparing specific attributes in the NSM DOs to predetermined rules and thresholds. When this method is not effective, the manager can instead use the sets as input to more complex algorithms: the NSM DOs in a given set $O_{i,t}$ can be treated as a vector F where each attribute name is assigned to a row a in the vector and the value for the attribute is the element v in that attribute's row such that $F_a = v$. Additionally, the vectors for the same value of t can be combined into a single vector that represents the state of the entire substation at time t. With the correct algorithms, the manager can detect that a specific event occurred using the information from the NSM DOs. Algorithms such as Support Vector Machine (SVM), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) are examples of ones used in previous works, as described previously in Chapter 3.

4.3 Implementation in Co-simulation Testbed

This implementation is a joint work with Mr. Abdullah Albarakati and Mr. Mark Karanfil.

In order to test NSM in realistic scenarios, we implemented its components within a co-simulation security testbed. A discussion of this implementation follows.

4.3.1 Co-simulation Smart Grid Security Testbed

The power model we use is modeled using the HYPERSIM real-time digital power system simulator. HYPERSIM [113] allows for modeling and simulating at a microsecond level a transmission system in real-time. It supports the IEC 61850 protocols GOOSE and SV. The transmission system used in our testbed is shown in Figure 4.3.

Our focus is on part of this system, which is highlighted in Figure 4.4. It receives power from several generators and serves the load marked Load_7. There are 3 P&C IEDs of note, Relay_2, Relay_4, and Relay_5, as well as their 3 associated CBs, named CB2, CB4, and CB5. Each relay is responsible for sending GOOSE messages to the CB with the same number in case the breaker needs to be opened in case of a fault. The relays receive measurements from local MUs (not shown in the figure) as SV PDUs, which is what is used to determine whether a trip signal should be sent. All of these IEDs are simulated by the HYPERSIM hardware.

In addition to the power model, a communication network is required to enable communications between MUs, relays and CBs using GOOSE and SV. Though this is not made explicit in Figures 4.3 and 4.4, Relay_2, CB2, Relay_4 and CB4 are not connected directly, but instead connect to an OpenStack network situated outside of HYPERSIM. This way, IEDs cannot communicate directly with other IEDs through HYPERSIM itself and must instead send messages through OpenStack, where they are eventually routed to their destination. This gives us the opportunity to inspect the traffic and even modify it en route to simulate various scenarios. Relay_2 and Relay_4 therefore send trip signals to CB2 and CB4 respectively in the form of GOOSE PDUs traveling on the OpenStack network, while MU2 sends SV PDUs to its corresponding Relay_2. In steady state, we can expect to consistently see 4 GOOSE PDUs per second per relay. This number increases slightly on state change due to the burst of PDUs that occurs then. As for SV traffic, it remains constant at 4,800 PDUs a second.

The OpenStack network itself is shown in Figure 4.5 and is fairly simple. The HYPERSIM hardware, two real PMUs and a Global Positioning System (GPS) clock are connected to a physical switch that is connected in turn with the substation switch, a virtual switch within OpenStack. Any communications sent from the IEDs in HYPERSIM go through the physical switch to OpenStack,

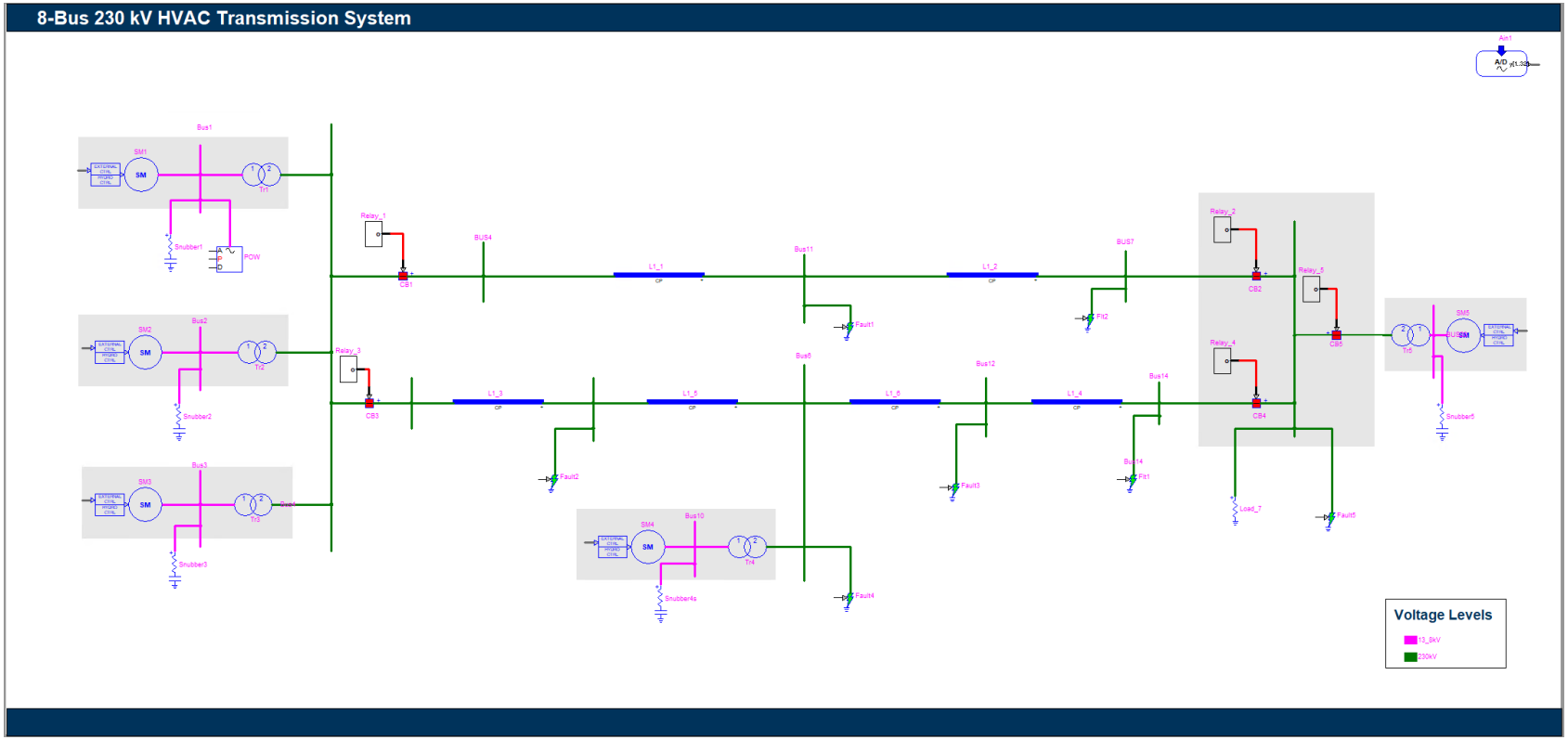

Figure 4.3: Transmission system line diagram as shown in HYPERSIM

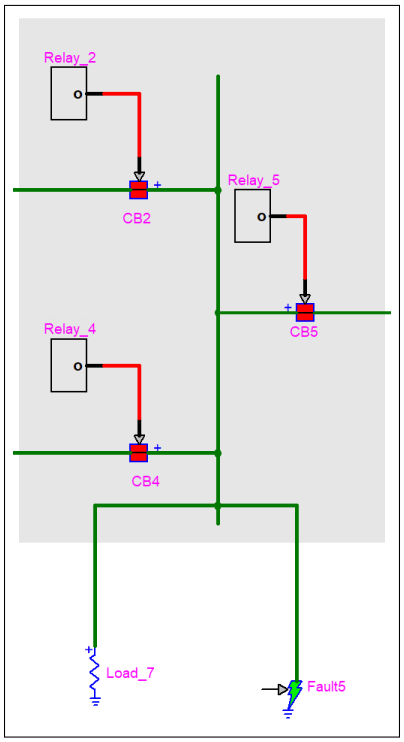

Figure 4.4: Zoom in on IEDs of interest as shown in HYPERSIM interface

where it is relayed to proxy machines: the proxies are part of NSM and discussed in the next section. After passing by the proxies, the network traffic is then forwarded back to HYPERSIM using the physical switch to reach its destination. Additionally, the control center and Phasor Data Concentrators (PDCs) are situated past a router or gateway, to represent how these are placed outside the substation.

In order to allow GOOSE and SV traffic to travel in the OpenStack network, we implemented small components that convert the PDUs into UDP packets upon sending and the reverse upon receiving. We do this because GOOSE and SV work directly on the Ethernet layer, meaning it cannot reach the proxies situated in the OpenStack network without this conversion. To identify the sender and receiver of the PDUs when they are converted to UDP, we assign them different UDP port numbers.

The HYPERSIM interface allows us to view the current state of the power system. For example, we can view the status of the five CBs, as shown in Figure 4.6 when running a simulation. A status of 1 indicates the breaker is closed, while a status of 0 means it is open. The first row of plots show the

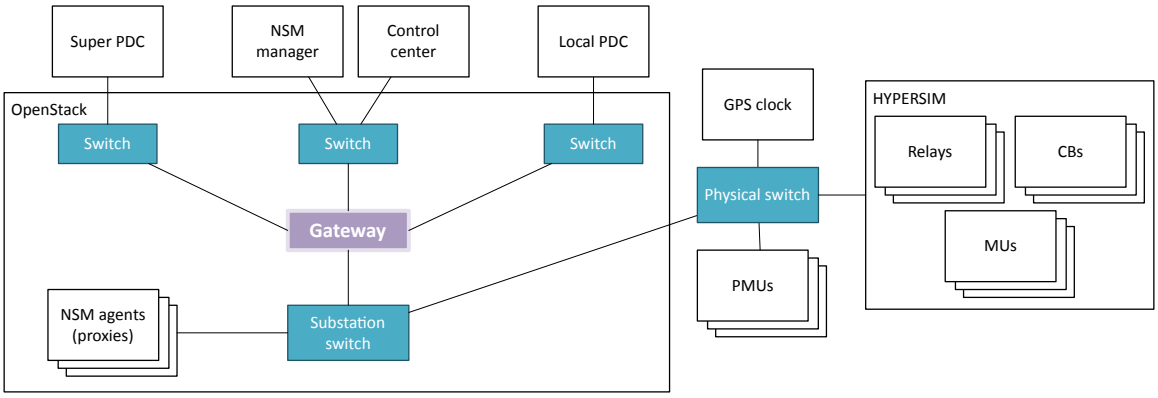

Figure 4.5: OpenStack network used for communications in HYPERSIM

current state of the breakers according to the relays (acting as the GOOSE publishers). The second row reports the status of the breaker, but according to the receiver of the GOOSE messages. The last row represents the actual status of the breaker. In this specific figure, CBs 1 and 2 go from open to closed due to a fault that we introduce in the power system simulation. The breakers closing is an automated reaction to maintain stability of the system. The remaining CBs remain closed as they are no longer affected by the fault. Because the GOOSE communications are working properly, the publishers and subscribers agree on the current status of the switch, hence why the plots in the same column are very similar. As we see later in Chapter 5, a cyberattack can disturb this synchronization.

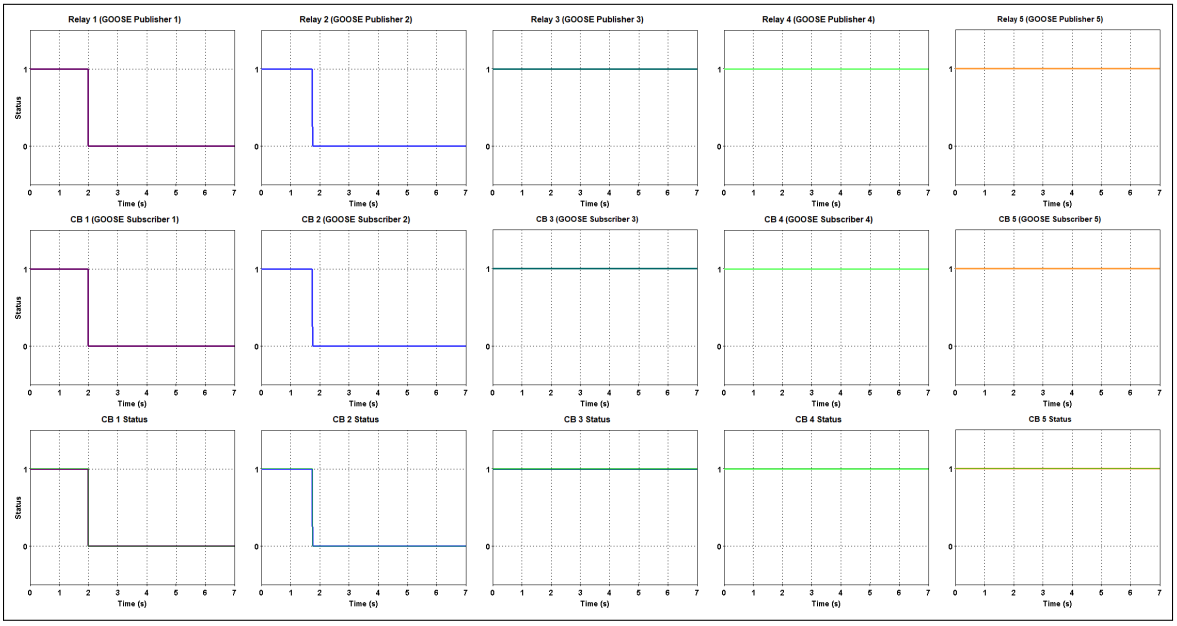

Figure 4.6: HYPERSIM interface to view status of breakers during a simulation

In addition, to viewing the status of the CBs, we can also use HYPERSIM to read the values of $stNum$ and $sqNum$ of the GOOSE publishers and subscribers while the simulation is running. This gives insight into the status of network communications. We show the values of $stNum$ and $sqNum$ for Relay_2 and CB2 in Figure 4.7 as captured during the simulation. Just like in Figure 4.6, we can notice a state change at around 1.5 seconds, which represents the opening of the CB. The state change means $stNum$ is incremented and $sqNum$ is reset to 0.

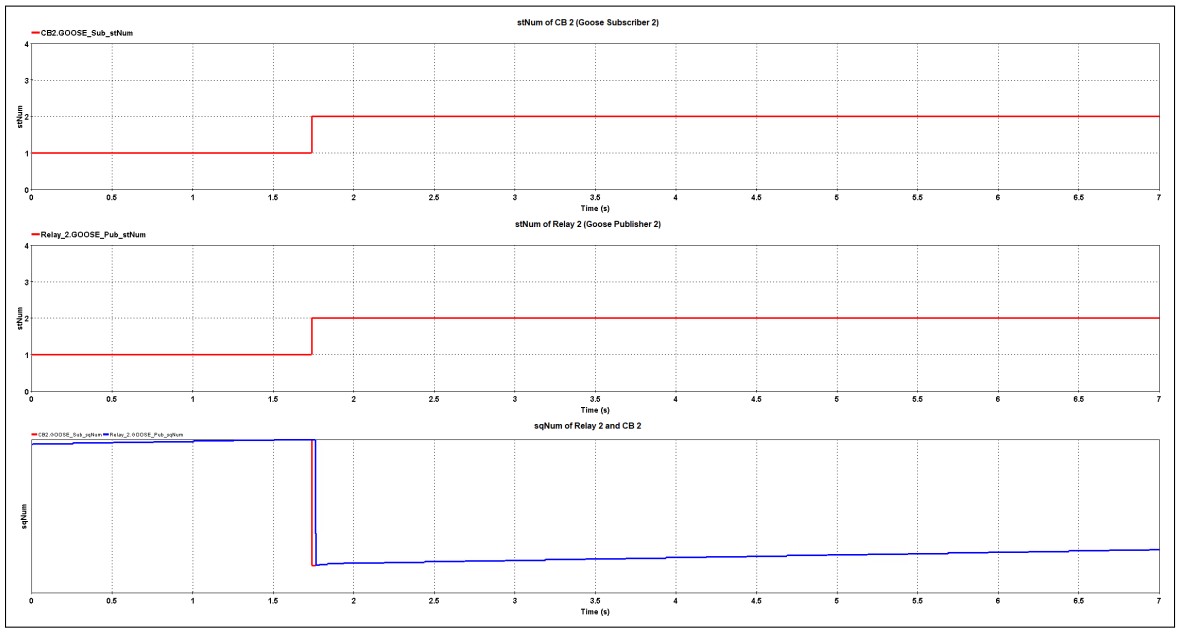

Figure 4.7: HYPERSIM interface to view $stNum$ and $sqNum$ values during a simulation

4.3.2 Components for Network and System Management

Once the co-simulation testbed is in place, the next step is to implement the NSM solution to monitor the network. We have previously discussed the required components for NSM in Section 4.2.2. In this section, we discuss the details specific to our implementation.

Implementation of Proxies as NSM Agents

To our knowledge, as of 2018, there are no available devices on the market that support IEC 62351-7: this was also the case in 2014 according to the report by EPRI [82]. The IEDs available for the testbed therefore cannot provide NSM DOs by themselves, leaving their implementation

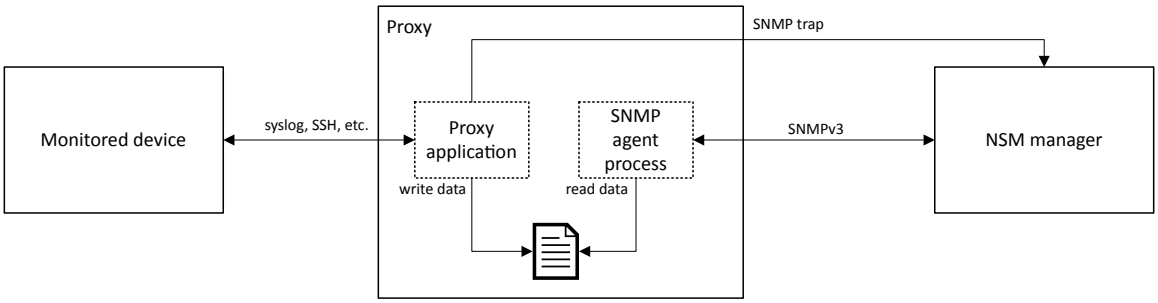

Figure 4.8: Design of proxy NSM agent

up to others. Because it is not possible to deploy additional code in the IEDs at our disposal, the responsibility of producing the SNMP MIBs and reporting to the NSM manager must be fulfilled by a different host.

In the SNE implementation documented in the EPRI report, an intermediate node, referred to as a proxy, is used to convert proprietary MIBs and data from other protocols into the IEC 62351-7 MIBs [82]. The proxy used by SNE is located at the NSM manager, meaning that the conversion of data from all monitored devices occurs at a central location. In our implementation, we use a different approach where the proxy is located much closer to the device it must maintain MIBs for. We show the inner workings of this proxy in Figure 4.8. There are a few reasons motivating the different design. Most importantly, the IEDs do not provide SNMP MIBs nor information on the status of their communications using IEC 61850 protocols. In order to obtain the data needed for protocol-related MIBs, the proxy itself must be situated such that it can inspect the traffic going to and from the device. Since a single proxy cannot always be placed in a location to simultaneously monitor all devices, we deploy one proxy per monitored device. In addition to inspecting traffic, the proxies can also interact with their respective devices using other protocols, such as SSH and syslog, to obtain more data for additional NSM DOs. The amount of information obtained this way depends heavily on the model of the device, as some models provide more services than others. EPRI reports that IEDs sometimes do not provide the information necessary to generate all NSM DOs [82] and this is also our experience.

Our approach has the advantage that the NSM manager does not need to distinguish between devices that natively support IEC 62351-7 and proxies. From its perspective, the physical switch of Figure 4.5 does not exist and the manager only communicates with devices that are compliant

with the standard (the proxies). It does not need to communicate with agents using any protocols other than SNMP or deal with any proprietary MIBs. Another advantage is that the traffic produced by the NSM manager and agents is closer to what the SNMP traffic would look like in an actual substation with NSM compliant devices. Deploying one proxy per monitored device is not costly for us as they are VMs within OpenStack, making it very easy to deploy multiple ones at will. The main difficulty of using proxies is their implementation as it can vary across devices. NSM DOs for communication protocols require traffic inspection that is usually done the same way for all protocols, but it cannot obtain information that only the host would be aware of (such as local configuration settings). NSM DOs that relate to device information rely heavily on other services provided by the monitored devices. If there are no useful services on a device, there can only be a few NSM DOs created for it.

Each of the proxies we deploy runs a version of the snmpd daemon, an SNMP agent part of the open-source NetSNMP application, that is customized to support IEC 62351-7 MIBs. This is done by writing new MIB modules for NetSNMP and recompiling snmpd [114]. The proxies also perform several tasks to collect necessary data to produce NSM DOs, such as traffic inspection, analyzing log data, and deciding when to send SNMP traps to the NSM manager. Many of these tasks are done using the Python [115] and Bash [116] languages, and the output of their execution is stored in a SQLite [117] database readable by snmpd. SQLite is chosen as it is lightweight and efficient. This enables the SNMP agent to obtain up-to-date information it can serve in the NSM DOs when it receives a request from the NSM manager.

Implementation of NSM Manager

The NSM manager performs three major tasks: it collects data from the NSM agents, analyzes the data for potential attacks, and provides an interface for a user to view the results of the analysis.

To perform data collection, the NSM manager runs an SNMP manager. It is responsible for sending and receiving communications over SNMPv3. To poll the agents, we use a Python application leveraging the PySNMP library [118]. Received NSM DOs are then added to an Elasticsearch [119] database that we refer to as the NSM DO database. In theory, the SNMP manager could also be used to send SNMP Set commands, but we have not found a use for these in our experiments as

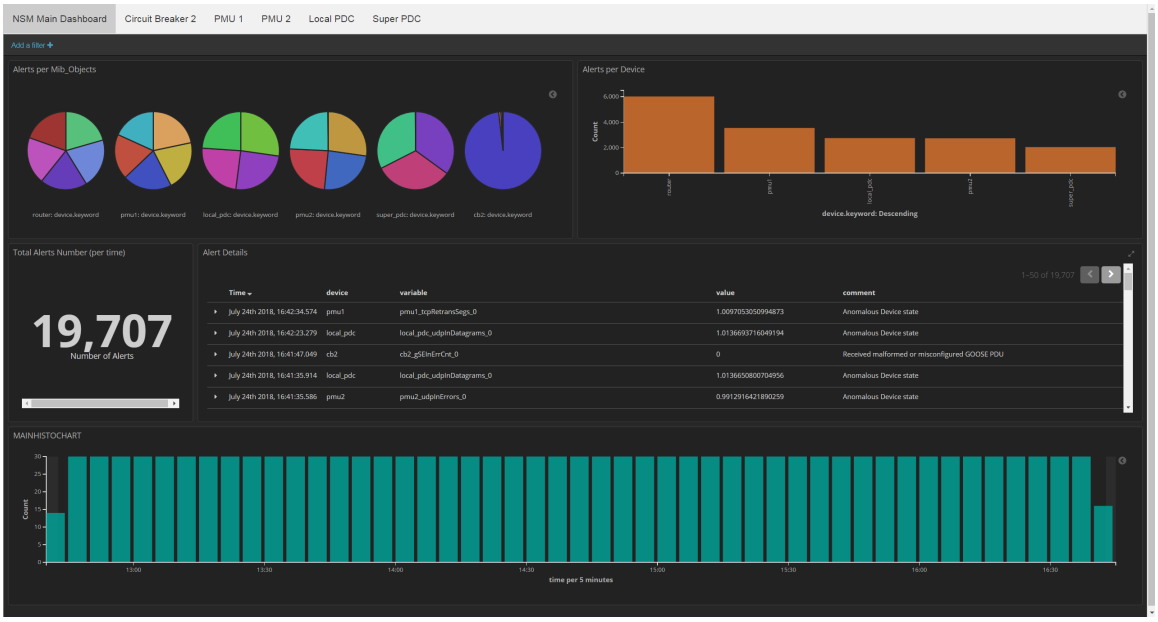

Figure 4.9: Kibana interface used to view anomaly alerts from NSM

the vast majority of the NSM DOs of IEC 62351-7 are read-only. To receive incoming SNMP traps, the NSM manager also runs a module that accepts unsolicited SNMPv3 communications, meaning traps, from the SNMP agents. We rely on the open-source snmptrapd application from NetSNMP [120] for this purpose. While our testbed does implement several traps from IEC 62351-7, we did not find that the data they contained was applicable to this specific work.

The data from the NSM DO database is used for analytics by a detection engine. It is written mainly in Python to leverage several useful libraries for analytics. The engine retrieves relevant NSM DOs and detects anomalies within the data using two techniques: rule-based detection and anomaly detection. Once the detection engine has found a potential attack, it places an anomaly alert in an Elasticsearch anomaly database (separate from the NSM DO database).

In order for operators to view the anomaly alerts and act upon them, we provide a web interface based on Kibana shown in Figure 4.9. Kibana [121] is a user interface compatible with Elasticsearch that provides several features such as dashboards to have an overall view of the anomalies, search capabilities to view specific anomalies,, and security features like authentication and use of SSL. This interface informs users in real-time about the status of the substation and enables them to react to mitigate threats when they are found.

NSM Data Objects Supported by Testbed

Table 4.4 shows the NSM DOs that are implemented and that can be collected within the testbed. As can be seen in the table, the set of NSM DOs supported varies considerably by device. In some cases, this is due to the particular role they play in the substation. Relays publish GOOSE PDUs and CBs are subscribed to receive them, explaining why they support NSM DOs from the PL2 and SL2 objects respectively. In other cases, it is due to the amount of information available to the proxy from the device. In particular, the PDCs provide a syslog service, and the logs sent contain a lot of information reflecting the current state of the PDC that sent it. Such information is not available for other devices such as the PMUs.

Table 4.4: NSM DOs of IEC 62351-7 implemented for the devices on the testbed

NSM DO	Parent object	MU	Relay	CB	PDC	PMU	Switch
PSPIed	Environmental Agent::Environmental				✓		
SecurityNotification	Environmental Agent				✓		
ConfigurationVersion	IED Agent::IED				✓		
FirmwareVersion	IED Agent::IED				✓		
LastEvent	IED Agent::IED				✓		
MisEvCnt	IED Agent::IED				✓		
PhyHealth	IED Agent::IED				✓		
PhyHealthChgCnt	IED Agent::IED				✓		
RBACDbUpdate	IED Agent::IED				✓		
SvcViol	IED Agent::IED				✓		
Watchdog	IED Agent::IED				✓		
WrmStrCnt	IED Agent::IED				✓		
SecurityNotification	IED Agent				✓		
Notification	IED Agent				✓		
ClockIssue	Clocks Agent::Clock.Clocks				✓		
HoldOver	Clocks Agent::Clock.Clocks				✓		
TimeTraceable	Clocks Agent::Clock.Clocks				✓		
DestMacAddr	GSE::GSEProvider.PL2		✓				
CBRef	GSE::GSEProvider.PL2		✓				
TxPduPerSecond	GSE::GSEProvider.PL2		✓				
InErrCnt	GSE::GSEProvider			✓			
SrcMacAddr	GSE::GSEProvider.SL2			✓			
ConfRevMis	GSE::GSEProvider.SL2			✓			
NdsComm	GSE::GSEProvider.SL2			✓			
TalExpCnt	GSE::GSEProvider.SL2			✓			
CBRef	GSE::GSEProvider.SL2			✓			
RxPduPerSecond	GSE::GSEProvider.SL2			✓			
CBRef	SV::SVProvider.SL2	✓					
RxPduPerSecond	SV::SVProvider.SL2	✓					
SrcMacAddr	SV::SVProvider.SL2	✓					

In addition to the NSM DOs of IEC 62351-7, we also implement a few OIDs from TCP-MIB and UDP-MIB as they are mentioned by IEC 62351-7 [8]. We list them in Table 4.5. The network

switches already support these MIBs, but the proxies require a different approach relying on traffic inspection to provide them.

Table 4.5: MIBs outside of IEC 62351-7 implemented for the devices on the testbed

OID	MIB	MU	Relay	CB	PDC	PMU	Switch
tcpInSegs	TCP-MIB				✓	✓	✓
tcpOutSegs	TCP-MIB				✓	✓	✓
tcpRetransSegs	TCP-MIB				✓	✓	✓
tcpOutRsts	TCP-MIB				✓	✓	✓
tcpInErrs	TCP-MIB				✓	✓	✓
tcpActiveOpens	TCP-MIB				✓	✓	✓
tcpListenerTable	TCP-MIB				✓	✓	✓
udpInDatagrams	UDP-MIB				✓	✓	✓
udpOutDatagrams	UDP-MIB				✓	✓	✓
udpNoPorts	UDP-MIB				✓	✓	✓
udpInErrors	UDP-MIB				✓	✓	✓
udpEndpointTable	UDP-MIB				✓	✓	✓

4.4 Real-time Data Collection and Detection

NSM is expected to run continuously in order to detect threats within the substation in real-time. The NSM agents and NSM manager are therefore creating and collecting data at all times in our testbed. The details of this collection follow.

4.4.1 Updating Data in NSM Agents

The NSM agents within the proxies continously monitor their associated host to populate the values of the NSM DOs. In general, the SQLite database containing the up-to-date data for the NSM DOs is updated every few seconds, since updating this database in real-time is not realistic for certain values. For instance, when updating a NSM DO representing a count of network packets received, accessing SQLite on every single packet can cause performance problems and result in altogether missed packets if the amount of network traffic is sufficiently high. This is actually a common issue with SNMP agents in general: Cisco mentions in their support center that it is normal for one of their switches to only update MIB statistics every 10 seconds, because its focus should be on switching tasks instead [122]. In our testbed, we set the update frequency to every two

seconds, as we find that it results in satisfactory performance. The NSM manager thus receives data that is at most two seconds old whenever it requests NSM DOs from the agents.

4.4.2 NSM Manager Polling

The SNMP manager process running in the NSM manager is configured to poll NSM DOs from all the NSM agents every 10 seconds. This results in six sets of NSM DOs collected at the NSM manager every minute. This polling frequency is selected for our particular setup as it provides sufficient time for the NSM manager to receive the SNMP replies from all the agents, save them, and process them using the detection engine before the next polling time. This does not represent the ideal polling rate for all substation networks as this depends heavily on the network configuration and context.

4.4.3 Detection Engine

The work on detection is a joint work with Mr. Abdullah Albarakati and Dr. Rachid Hadjidj.

The detection engine makes use of both rule-based detection and anomaly detection. This allows us to benefit from the advantages of both methods [102]. Rule-based detection is efficient, but it cannot detect attacks it is not explicitly programmed for. Anomaly detection, on the other hand, has the potential to detect unknown attacks [106], with the downside that it requires training. The detection engine analyzes the NSM DOs saved by the data collection to detect if any attacks occurred. It starts by converting the data from the NSM DO database into a format more useful for detection purposes. It combines the data from all the different devices in such a way that one can have a global view of all the substation devices at a given moment in time. We refer to this as a "snapshot" of the substation. To create a snapshot, this module requests the latest data from each device for given time t (i.e. the data with the timestamp closest to t without exceeding it). The relevant values for each device are then copied into a new data structure for the snapshot. Snapshots are saved in their own Elasticsearch database. It continuously take these snapshots as their input for analysis.

Rules used by the rule-based detection are written mainly in the form of simple conditions or

thresholds. For instance, we can add a rule to indicate that the value of the NSM DO ConfigurationVersion should not change. If an attacker then attempts to update the configuration of a device, perhaps to make it malfunction, NSM detects this change efficiently because the NSM DO changes as well. However, there are some cases where rules are not sufficient to detect a problem. We rely on anomaly detection for those. Specifically, we make use of LSTM, an algorithm suited to time-varying multivariate series, using Keras [112] and TensorFlow [111]. Other algorithms that we considered are RNN and GRU, though our testing indicated that LSTM is the most accurate for our purposes.

When using LSTM, each NSM DO is treated as a separate variable or feature that varies over time. Each variable is dependent on its value in the previous snapshot, and also on the current value of the other variables. This is because NSM DOs do not change independently from each other. For example, the NSM DO representing the amount of PDUs sent by a publisher is likely to influence the NSM DO for the amount of PDUs received by a subscriber. As such, we give as input to LSTM a set of past snapshots, sorted by time. The LSTM model uses this set to predict the values that should be in the next snapshot. If the prediction is close to the real snapshot, this is considered normal behavior. If the prediction is off by a certain amount (determined during training), the snapshot is considered anomalous. We actually make use of several LSTM models, each one predicting the value of a single NSM DO based on the input snapshots. This approach has the advantage of pinpointing which of the NSM DOs contains the anomalous value.

If the engine detects an anomaly in a given snapshot, this results in the creation of an alert that specifies the anomaly and the NSM DOs that allowed for its detection. Alerts are placed in a special Elasticsearch database that is made accessible to operators through the Kibana interface.

Chapter 5

Security Assessment of Network and System Management

In this chapter, we assess the security of NSM used in the context of a digital substation by testing its capabilities against relevant attack scenarios. We elaborate attacks against IEC 61850 protocols, then design detection algorithms using NSM DOs to attempt to detect them. The algorithms are validated using the testbed described in Chapter 4. In this way, we can establish the attacks that are reliably addressed through the application of NSM.

5.1 Classification of Cyberattacks Targeting Substation

5.1.1 Definition of Attacker's Objective

In keeping with the security goals of IEC 62351 [7], previously outlined in Section 2.5.1, we consider that the more important threats to the IEC 61850 substation are those that aim to interfere with power delivery. These target the goals of availability, integrity, authentication, authorization and non-repudiation. Examples of such objectives are:

- Damaging critical primary equipment (e.g. transformers);
- Damaging critical devices in the information infrastructure (e.g. IEDs);
- Causing an unnecessary blackout through malicious automation or control commands;

- Preventing an operator from executing control operations;
- Hiding tracks after a security event to interfere with recovery efforts.

We consider that the attacker ultimately aims to achieve one of these objectives. In order to do so, the attacker is capable of executing any attack from a set of basic cyberattacks. By combining these basic building blocks, the attacker can elaborate a multi-step attack to achieve one of the aforementioned end goals.

5.1.2 Elaboration of Capabilities Available to Attacker

To systematically elaborate the capabilities of the attacker, we first define a list of actions that an attacker can perform on some target (e.g. an IED, a network link, a file, etc.). Each target can be altered by the attacker through four possible actions:

(1) Read (or steal) from the target;

(2) Add an instance of this target;

(3) Modify the target;

(4) Delete (or corrupt irreparably) the target.

We then consider the set of potential targets in a substation that can be affected by these actions. These vary significantly depending on the architecture. The general list we define for the purposes of this thesis is not necessarily applicable to all substations. We show the targets considered in Table 5.1. By applying each of the possible actions to the targets, we can derive several capabilities (basic cyberattacks). For example, the action "add" applied to processes running in a host represents the addition of a process, possibly malware, by the attacker. If adding multiple processes, this can also constitute an attempt at a DoS attack that exhausts the host's resources, such as a forkbomb attack. A "delete" action on a process, on the other hand, represents a DoS attack where the attacker kills or crashes an essential process in the host. We list the possible capabilities found with this approach in Table 5.1. Some actions are not applicable to specific targets, so we do not list those combinations.

Table 5.1: Attacker capabilities on hosts and networks

Target	Action	Capabilities
Process	Read	Read process list
	Add	Run malware, saturate host's resources
	Modify	Run modified (malicious) process
	Delete	Kill or crash essential process
Logs	Read	Read logs
	Add	Add false logs
	Modify	Falsify existing logs
	Delete	Delete logs to hide traces of attack
Credentials, keys	Read	Steal or brute-force credentials, keys
	Add	Add malicious users and keys
	Modify	Change existing users and keys to hijack
	Delete	Delete existing users and keys to lock out
User activity	Read	Log keystrokes, spy on users
	Add	Execute unauthorized commands
	Delete	Block legitimate commands
Configuration	Read	Read application settings
	Add	Force use of unused settings, downgrade attacks
	Modify	Modify settings, factory reset
	Delete	Delete settings
Files	Read	Read files
	Add	Add files
	Modify	Modify files
	Delete	Delete, corrupt files
Operating system (OS)	Read	Read OS settings
	Modify	Modify essential parts of OS
	Delete	Delete or corrupt OS (see PDoS)
Network packets	Read	Sniff and analyze network traffic
	Add	Inject forged packet, replay previous packet, flooding attack
	Modify	Intercept and modify packet
	Delete	Drop, delay or corrupt packet
Network hosts	Read	Scan to map network
	Add	Connect unauthorized host to network
	Modify	Compromise host using exploit, stolen credentials, etc.
	Delete	Crash or shut down the host using exploit, stolen credentials, etc.
Network links	Read	Scan to map network
	Modify	Execute MitM attack

5.1.3 Study of Denial-of-Service Attacks

Attacks against availability, namely DoS attacks, are among the most relevant when studying the security benefits of IEC 62351-7 and NSM. Part 1 of IEC 62351-1 explicitly names NSM as a security countermeasure for availability [7]. As stated in Section 2.5, the IEC 62351-7 standard explicitly mentions DoS attacks as a gap that is not filled by previous parts of the security standard [8]. For this reason, we study and categorize DoS attacks in detail in this section.

We define a DoS attack as an attack that aims to reduce or eliminate the availability of a system. Often, the expression "DoS attack" is used to refer to a packet flooding attack, but we do not use this definition as it does not encompass the many other possible DoS attacks such as PDoS. This fits the definition by OWASP [12] and the requirements of IEC 62351-7, which states that DoS can take

many forms [8]. Due to this, it is difficult to enumerate all possible variants and elaborate a response to mitigate each of them. Instead, we consider the following aspects of DoS attacks, similarly to how OWASP documents application security risks [123]: the threat agents, the attack vector used, the impact of the attack and the duration of the attack. A specific DoS attack can belong to multiple categories, which are shown in Figure 5.1 and discussed in this section.

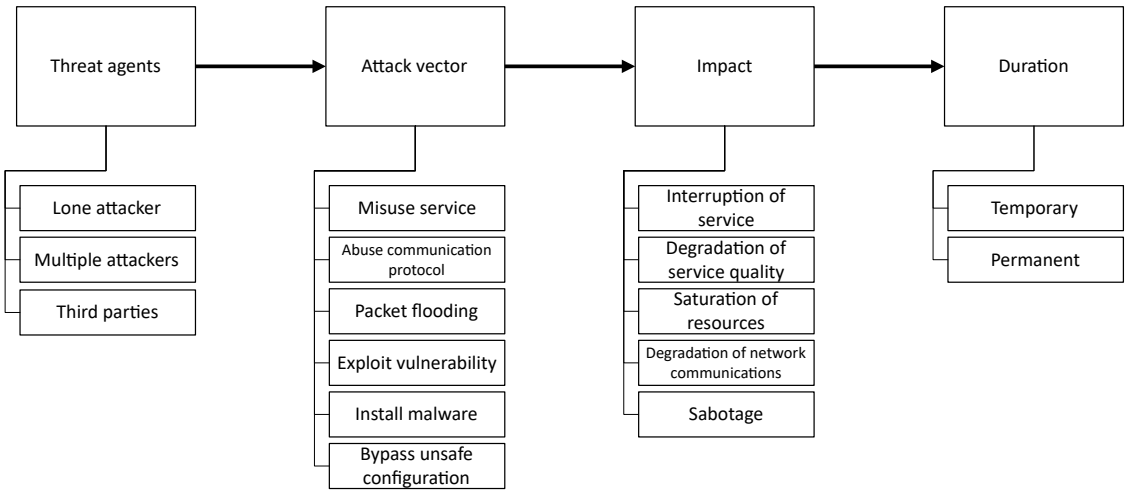

Figure 5.1: Categories of DoS attacks.

Threat Agents

The number of hosts required to execute a successful DoS attack can vary. This can affect the odds of successfully carrying out the attack, as well as the strategies used for proper mitigation.

Lone vs. multiple attackers Some DoS attacks are effective when executed by a single attacker. Other variants benefit from coordination of multiple attackers sharing the same goal. This is the main distinction between a Denial-of-Service (DoS) attack and a Distributed Denial-of-Service (DDoS) attack. The advantage gained by attackers that cooperate is that they can combine resources to increase the impact of the attack. Their target must also detect and isolate many of the attackers to successfully stop them. Doing so is usually more difficult than locating and shutting down a single source for an attack.

Third parties A reflected DoS attack involves the attacker tricking unwilling devices, called the reflectors, to participate in the attack. An example is the DRDoS attack using SNMP [38], where a spoofed SNMP request sent by the attacker to outside machines causes them to mistakenly send their own responses to the target and overwhelm the latter.

Attack Vector

There are many attack vectors that can be used in a DoS attack. We discuss some of the vectors that are commonly used.

Misuse service An attacker can execute a DoS attack against a service by using the service in unintended or dangerous ways if it is not designed to account for those situations. For example, a poorly-designed file storage service that allows users to upload files, but omits enforcing size limits, can be abused by an attacker uploading huge files that occupy all available space. By doing so, other users cannot upload any of their own files. In the context of ICS's, the infamous Stuxnet malware can change the speed of a PLC's motors to repeatedly speed up and slow down to extreme speeds outside their normal range, potentially damaging them [17].

Abuse communication protocol Several DoS attacks that rely on a network communication protocol are possible due to a property of the protocol used. The fact that UDP does not prevent modification of the source IP address in UDP datagrams enables DRDoS. This attack relies on spoofed source IP addresses to trick reflectors [38]. TCP requires that the server maintain the state of all TCP connections, making SYN flood attacks possible [124]. Amplification attacks exploit protocols, such as SNMP [38] or DNS, that provide request-response exchanges where the response is potentially much bigger than the request: this significantly increases the amount of malicious traffic that the attacker is able to direct towards a target.

Packet flooding An attacker can send an overwhelming amount of malicious traffic to a target that is not equipped to handle the high load. This kind of attack often also abuses a communication protocol to produce the traffic.

Exploit vulnerability According to RFC 2828, a vulnerability is a weakness in the design or implementation of a system that can be exploited to compromise security [125]. Some examples are buffer overflows, Structured Query Language (SQL) injections or application-specific bugs. Vulnerabilities can be used as part of DoS attacks when they affect availability, for example by causing a device to crash or causing a device to enter a bootloop. An example of a possible DoS attack using this attack vector is a bug in the Server Message Block (SMB) protocol that could crash the Windows OS entirely when exploited [126]. Software updates often aim to remove such vulnerabilities, meaning that exploiting a specific one might only work if the target device uses a corresponding specific version of a given software. Unpatched devices are ideal targets for an attacker as they are likely to have several vulnerabilities.

Install malware Malware can be used to automatically carry out a DoS attack. The Morris worm of 1988, among the first of computer worms, caused a massive DoS attack unintended by its author by spreading from machine to machine. It eventually resulted in clogged network routes and unresponsive machines [127]. More modern examples are BrickerBot, which spreads itself and executes PDoS attacks on Internet of Things (IoT) devices [128], and ransomware that can lock away critical system files in an attempt to extort money.

Bypass unsafe configuration Human error can undermine even the best defenses. If a device does not follow standard security practices, such as avoiding default credentials, using complex passwords, configuring a firewall, or disabling remote access, it can be trivial for an attacker to bypass defenses she would normally encounter. Both the infamous Mirai botnet and BrickerBot relied on guessing Telnet passwords to compromise IoT devices left configured with default credentials [128]. Similarly, devices connected to the Internet can be used as reflectors for SNMP amplification attacks, because manufacturers ship them with SNMP active by default with very unsafe credentials [38].

Impact of Successful Attack

Any DoS attack reduces the availability of its target, but how much it affects it can vary considerably. This section discusses the possible outcomes of such attacks.

Interruption of service The target stops performing its tasks, as if it has stopped the running necessary processes to complete them. An attacker can cause this literally by compromising a host and stopping the processes. There are more subtle and easy ways to achieve a similar effect by exploiting flaws in the target system. For instance, an attacker can lock a user out of her account for an online service by deliberately failing to log in to said account several times. This tricks the service into blocking login attempts for some time in an effort to prevent password guessing attacks, making the service unavailable to the user [12].

Degradation of service quality Stopping a service entirely is not always required to damage a system's availability. In some cases, it is sufficient to make the target perform its tasks more slowly or less reliably for an attack to be successful. An example is malware that uses the CPU of a device, causing any other task to take more time to complete. This might suffice to discourage users from using the device at all. In contexts where there are specific performance requirements, this kind of attack can prevent the system from meeting these requirements and have similar consequences to a service interruption. For instance, a host that takes too long to respond to a network message can make the sender time out and assume that the receiver is not working at all.

Saturation of resources The target can no longer perform its tasks because it has saturated the resources necessary to accomplish them as a result of the attack. The resources affected can include the target's CPU, memory, disk space, number of available network connections, and more. TCP SYN flood attacks overwhelm a server by opening enough pending TCP connections that it has no more resources to accept new ones [124]. A forkbomb continuously creates new processes to saturate the OS's process table and monopolizes the use of the CPU [129]. In some cases, resource saturation can cause crashes or damage to the equipment that is not designed to handle the high load.

Degradation of network communications If a the target requires communication over a network to provide a service, an attacker can attack it by aiming for its surrounding network nodes and links. Redirecting packets or delaying them so that they time out prevents the packets from reaching their destination. Sending an overwhelming amount of traffic through DDoS on the intermediate network can have the same effect by failing critical routers [130]. Even though the target itself is healthy, it is essentially useless if it cannot communicate over the network.

Sabotage The target device is completely shut down. IEC 62351-7 explicitly considers this "the ultimate DoS attack" [8]. An attacker can do this using unauthorized commands to shut down the device, or by exploiting a vulnerability that causes relevant processes to crash. Going a step further, if the target's boot record or important OS files are erased or corrupted, the device cannot start at all. This is referred to as PDoS and is the main purpose of the malware BrickerBot, which targets IoT devices to wipe them [128]. Wiping modules are a common feature in malware such as CrashOverride [19] as they cause further damage and allow attackers to hide their tracks. Ransomware, a type of malware that encrypts files and requests a ransom in exchange for the needed decryption key, essentially causes PDoS if it happens to encrypt critical system files: the Petya ransomware did this by encrypting the Master Boot Record (MBR) [131].

Duration of Impact

A DoS attack can last temporarily, meaning it is only effective as long as the attacker is active, or it can persist after the attack is executed. The typical DDoS attack ceases to have an effect once the attacker stops sending packets or gets blacklisted. In a PDoS attack, described previously, this is not the case and intervention is required to solve the issue [128]. This kind of attack forces the owners of the devices to spend time and effort to reverse the effect.

5.1.4 Denial-of-Service Attacks in IEC 61850 Substation

As availability is of high priority for a substation, DoS attacks are a major concern for utilities. The substation relies on several pieces of equipment being available and working correctly in order to function. The primary equipment, including the Large Power Transformers (LPTs), must be

working and in good condition; the MUs streaming SV PDUs must run without interruption to maintain visibility of measurements; IEDs must be able to send GOOSE PDUs quickly enough to react to changing situations and protect the primary equipment; the control center must be able to monitor the substation; the operator must be able to log in and send control commands using MMS in a timely manner; the time must be synchronized among all devices for accuracy; the network devices and links must forward communications in a fast and reliable manner. All of these are examples of targets for a DoS attack that could impact the delivery of power or the health of the electrical equipment. Just like typical IT networks, the substation network is vulnerable to the attacks described in Section 5.1.3 and the categories of DoS attacks shown in Figure 5.1 are also applicable to substations.. There are a few differences in the way that these attacks must be conducted due to the different equipment present.

Attack Vectors in Substation

In general, the attack vectors that can be used for a DoS attack in an IT network are also available in a substation. Packet flooding might be even more effective in the context of the substation, given the lower amount of resources available to IEDs [8] and their strict performance requirements. Additionally, attacks using malware tailored to the substation, like how Stuxnet was designed to target nuclear power plants [17], are a very real threat. The malware CrashOverride is made to target IEC 61850, among other ICS protocols, and it is able to wipe devices it has compromised [19]. Such malware is unlikely to be the last of its kind. Much like in IT networks, IEDs that have poor configuration can also be exploited by an attacker.

Some unique attack vectors to the IEC 61850 substation exist due to the unique communication protocols and devices that are used there. In the substation, protocols found in IT networks such as FTP, HTTP or Telnet are used along with the three unique protocols of IEC 61850: GOOSE, SV and MMS. By misusing GOOSE and SV, it is possible to send or block trip signals to circuit breakers, and to falsify measurements from the MUs. All are actions that threaten the continuous operation of the power system. MMS can similarly be used to send commands to substation devices, even remotely. Additionally, the three protocols have unique characteristics and vulnerabilities that can be exploited. These are discussed further in Section 5.3.

Impact and Duration of Successful Attack in Substation

Due to the performance requirements of the IEC 61850 substation, it is not necessarily required to completely stop a given service or saturate resources to cause problems in the power system. If a relay needs to send PDUs within a window measured in milliseconds, a small delay attack or a slight change in its internal clock could be sufficient to make all of its messages time out. The more limited resources of IEDs also mean that it is easier to saturate their resources and render them non-functional. Among the worst case scenarios is the sabotage of IEDs and other critical devices, as a PDoS attack can be devastating in a substation. According to the NESCOR failure scenario AMI.28 "Failed Patching Causes AMI Devices to Stop Operating" [55], this kind of problem can even occur by accident due to a firmware update that fails. Having to restore affected IEDs can lengthen the time to recovery due to the travel time needed to reach the substation and the time needed to reconfigure the system. If a piece of primary equipment, such as a LPT, is damaged by a DoS attack, replacing or repairing it, a process that can potentially last months, can entail costs in the range of millions for the utility [132].

Challenges in Implementing Defenses

Often, manufacturers of IEDs do not include services that could help monitor its status in enough detail [82]. If they are included in an IED, it is often in a proprietary format that makes it difficult to integrate with other existing solutions. Traditional IT solutions deployed in an IEC 61850 substation are unlikely to perform well as some of the communication protocols present are specific to the substation context. Common protocols such as UDP, FTP and the like are already well-known, but protocols such as IEC 61850's MMS, GOOSE and SV require special consideration as possible attacks can take advantage of their unique characteristics or require knowledge of power systems to understand. Additionally, no matter which security countermeasures are included in the substation, they must not have significant impacts on performance. For these reasons, traditional IT solutions require adapting before being effective in the IEC 61850 substation.

5.2 Elaboration of Attack Trees for IEC 61850 Substation

Based on the discussion of Section 5.1.2, we can construct many possible attacks against a substation using attack trees where nodes are actions listed in Table 5.1. We are interested in a complete scenario that illustrates all the steps the attacker must take before achieving her goal, similarly to the approach used by the detailed NESCOR failure scenarios [56]. We were not able to find a NESCOR scenario pertaining to attacks within the substation or to IEC 61850 specifically. We therefore elaborate our own trees based on the existing ones.

Our attack trees are elaborated based on a specific substation architecture, because individual nodes in the tree and their order can be affected by the connectivity between the hosts. They are also based on the attacker's possible actions as defined in Table 5.1. Note that when we design attacks, we do not make assumptions about whether the specifications of the various parts of IEC 62351 are applied. These specifications can have a potential impact on the feasibility of certain attacks, for example by adding encryption to some communications. However, since it is plausible that the equipment in the substation does not conform to these specifications due to concerns about backward compatibility [7] or performance [43], we cannot assume that they are in use.

5.2.1 Description of Target Substation

The substation communication architecture considered for the elaboration of attack trees is shown in Figure 5.2. We use this architecture because it reflects the one used in the testbed described in Chapter 4, allowing us to apply the attack trees to it. The station and process buses are merged in a single switch, labeled the substation switch. Any devices directly connected to the substation switch is considered part of the substation. These are the GPS clock, P&C relays, CBs, MUs and PMUs. We group them together to simplify the diagram. Meanwhile, the gateway connects the substation to the WAN so it can communicate with the remote control center and the PDCs.

In this network, the MUs stream SV traffic to the relays. The relays then act on this data and are expected to send GOOSE trip messages to the CBs if the data coming from the MUs indicates instability. The PMUs transmit synchrophasor data to the PDCs located outside the substation. This particular communication relies on the C37.118.1 protocol [133], which we will not discuss much

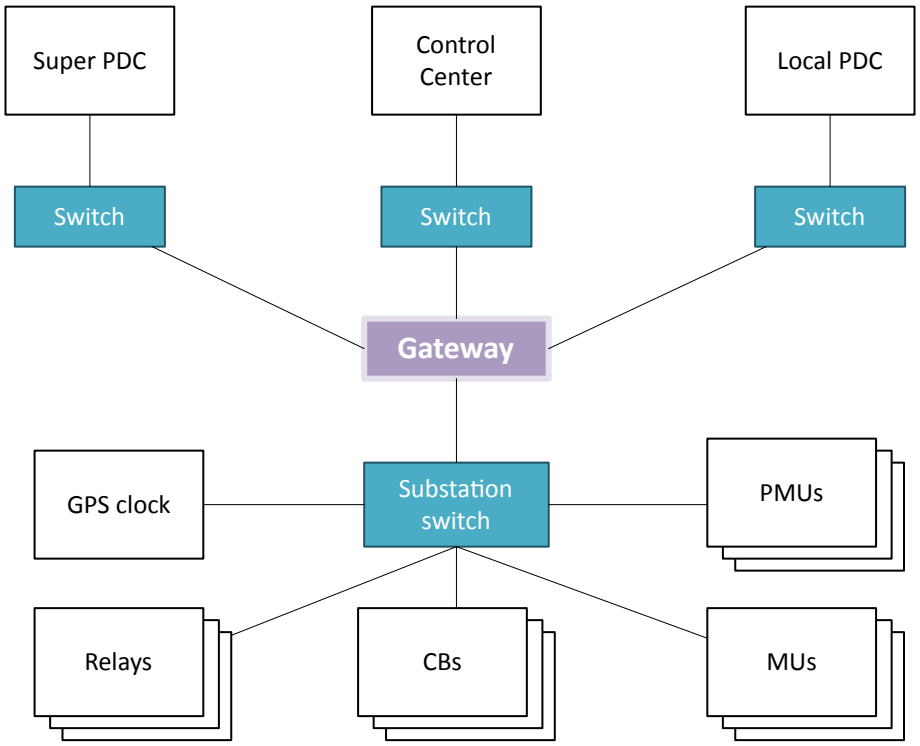

Figure 5.2: Substation architecture considered when constructing attack trees

as it is out of our scope.

5.2.2 Description of Attack Trees

Due to the complexity of some of the attack trees, we emulate the approach by NESCOR where subtrees common to several attacks are defined separately [56]. The subtrees are defined at the end of this section. In the attack trees discussed, a white node indicates a single action, a blue node represents an entire subtree, and a red node represents the end goal of an attack.

5.2.3 Attack Tree: Prevent Tripping Breakers to Damage Equipment

The attacker's objective in this scenario is to interfere with functions needed to trip CBs when they are supposed to. By blocking legitimate trip commands, the system operates in a dangerous state that can damage valuable equipment such as transformers. There are several ways to achieve this goal. If considering network attacks, interfering with GOOSE and SV communications can either hide faults when they occur or block the GOOSE trip messages from reaching the CB. In

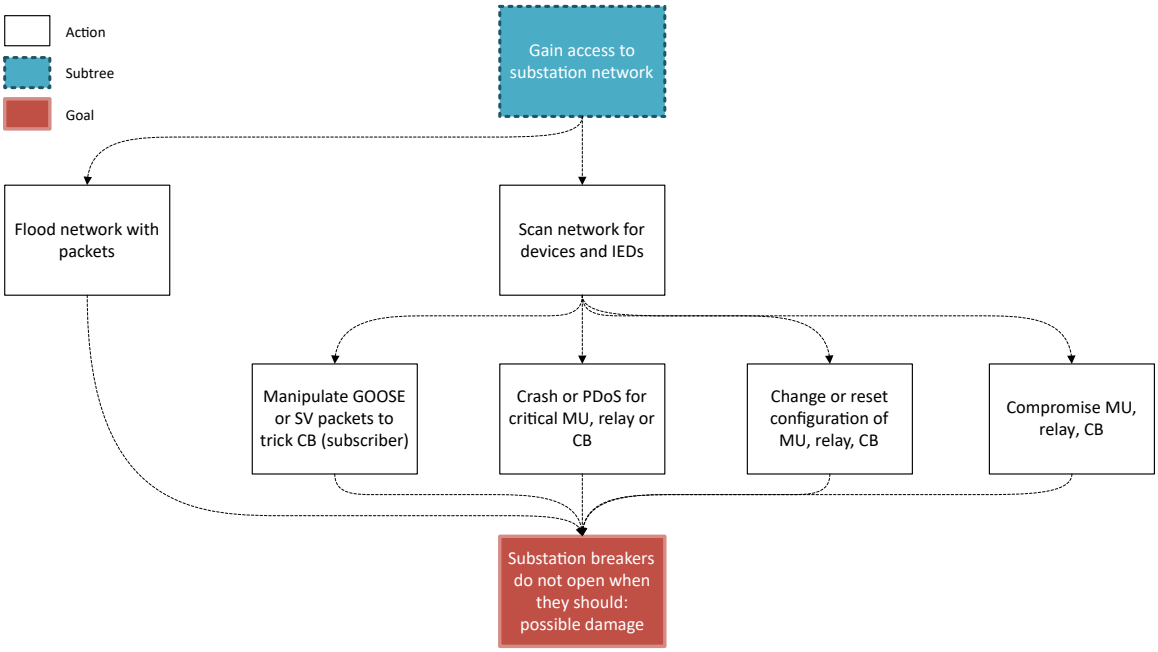

Figure 5.3: Attack tree to damage equipment by preventing trip commands from reaching CBs

other cases, an attacker might prefer to crash a critical IED or network node using some exploit, as this requires less knowledge of IEC 61850 protocols. We show many of these possibilities in the attack tree of Figure 5.3.

5.2.4 Attack Tree: Tripping Breakers Unnecessarily to Cause Blackout

The goal of this attack is to force CBs to trip unnecessarily while the system is stable, causing a blackout. This is reminiscent of the 2015 attack on Ukraine's infrastructure [18] and the NESCOR failure scenario DGM.11, where several substations receive commands in succession to trip breakers. This is considered a high-risk scenario by NESCOR [56].

Tripping a CB requires that the P&C relay issues a GOOSE trip message. The attacker has a few options. Sending a command through MMS can make the relay send this message. Alternatively, one can produce a fake GOOSE PDU by manipulating network packets or by compromising the relay and using it to send the message. Another approach is to trick the relay, either by aiming for the SV traffic to falsify the measurements and indicate faults, or by changing its configuration. In both cases, the relay can be made to react by tripping the CB. We show the attack tree for this scenario in Figure 5.4.

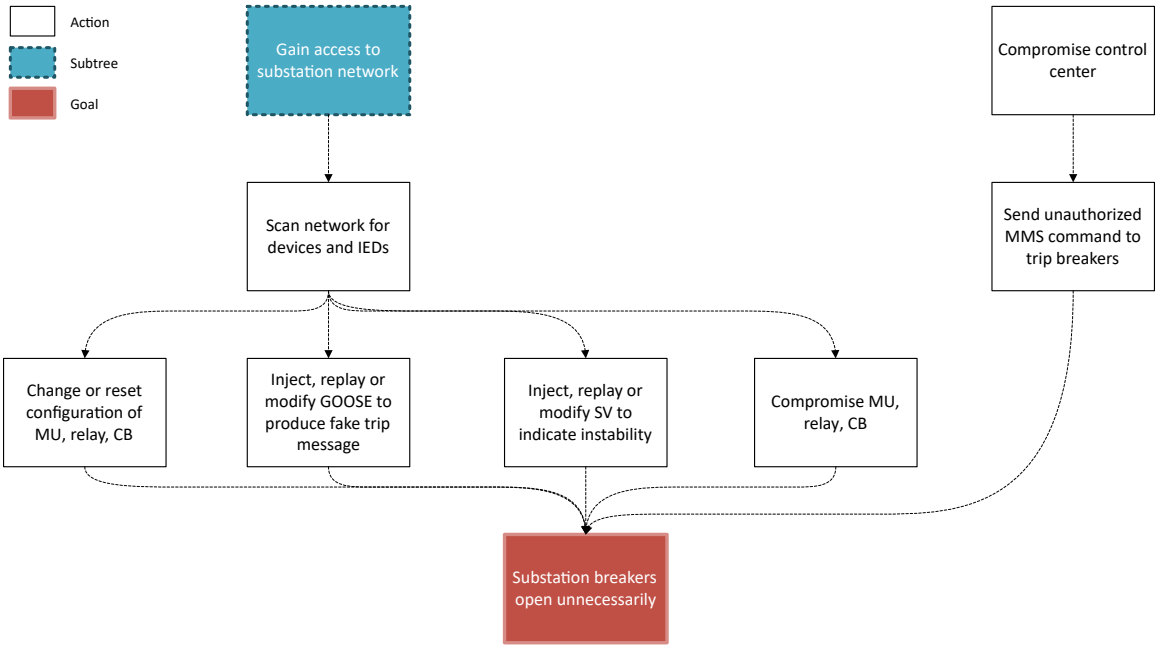

Figure 5.4: Attack tree to cause a blackout by tripping CBs unnecessarily.

5.2.5 Sub-trees

Gain Access to Substation Network In many cases, the attacker must first infiltrate the substation network before she can perform any additional actions. NESCOR scenarios do include a subtree for this case [56], but it is different from our own, shown in Figure 5.5.

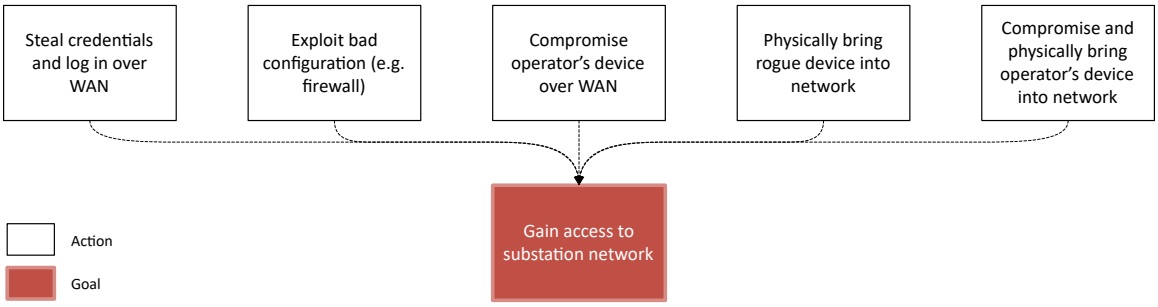

Figure 5.5: Attack tree to gain access to the substation network.

5.3 Design of Attacks on GOOSE, SV and MMS Protocols

In this section, we elaborate a methodology to design cyberattacks specific to the communication protocols of IEC 61850 substations. Many of the nodes shown in the attack trees of Section 5.2 are

applicable to any IT network. On the other hand, nodes that describe actions performed on network packets confined to substations, namely the IEC 61850 protocols MMS, GOOSE and SV, are unique to this context. One such example is the node "Inject, replay or modify GOOSE to produce fake trip message". These are the nodes we are most interested in studying and ultimately test against NSM.

In this section, we delve into the details of possible cyberattacks on IEC 61850 communication protocols. In the existing literature, several such attacks are already described (see Chapter 3), but they tend to be scattered across various publications in an unorganized fashion. In this work, we derive potential attacks in a more systematic manner in order to capture more possible variants. These are very important as attacks that are similar might require different methods to be detected.

5.3.1 Overall Methodology to Design Cyberattacks on Communication Protocols

When examining possible attacks on a network protocol, we consider the ways in which an attacker can affect the network packets. From Table 5.1, we have established that network packets can be manipulated in 7 different ways by an attacker. Packets can be *sniffed* (read) to retrieve the information they carry; the attacker can construct her own seemingly valid packet and *inject* it into the network; the attacker can attempt to *replay* (re-inject) a packet that was previously sent; large amounts of packets can be sent at once to *flood* the network; the attacker can execute a MitM attack and *modify* packets between the source and destination; packets can also be *dropped* entirely or simply *delayed* after a MitM attack.

Many of these attacks require knowledge of the protocol to be effective. For example, injecting a malicious GOOSE PDU requires that this PDU comply with a few rules, notably by having the correct value for the $stNum$ and by following the structure expected for GOOSE PDUs, before it can have an effect. Flooding attacks, where many PDUs are injected, are not as concerned by this as the goal is to saturate network links and host resources rather than affect the communication protocol itself. We therefore consider it as an attack that can happen in any protocol. We also assume that any protocol's packets can be sniffed by an attacker. The main differences across the protocols is the information that the attacker can obtain this way, as information in encrypted fields is much more difficult to extract from the packet. Hence, we consider that flooding and sniffing attacks can happen using any protocol.

When it comes to the remaining attacks, the protocol must be analyzed to define exactly how they must be carried out. Relevant information about the protocol includes its message structure, the meaning of the fields that are part of the structure, and how the protocol behaves according to these fields. This information gives insight into the possible ways to manipulate packets that can trick the entities into performing certain actions. We are also concerned with whether the fields are protected using cryptography, such as encryption, MACs or digital signatures, as these make it much more difficult for an attacker to alter them and maintain the validity of the message. In our analysis, we do not assume that these attacks are rendered impossible by cryptography, because it is possible for the attacker to crack or steal the required cryptographic keys, however difficult that may be. Once the analysis of the protocol is done, it is then possible to define how to conduct injection, replay, modification, drop and delay attacks specific to the protocol.

5.3.2 Methodology to Design DoS Attacks on GOOSE and SV Protocols

In this section, we discuss how to design DoS attacks (both denial-of-service and degradation-of-service) specific to the GOOSE and SV protocols, assuming we have access to information about their behavior. In this scenario, the objective of the attacker is to block or degrade communications using these protocols. The attacker can achieve this by dropping all PDUs being exchanged, but we are interested in other methods as well. These variants are relevant because they require different capabilities on the part of the attacker and different methods to be detected by the defender.

The target of this DoS attack is a GOOSE or SV PDU going from publisher to subscribers within the substation network. If attacking multiple PDUs, this attack can simply be repeated for each of them. The goal of the DoS attack is either of the following:

- Invalidate the target PDU so it is discarded by the subscriber;
- Degrade the quality of the PDU.

Because GOOSE and SV both use the same publisher-subscriber model, this attack causes silent failures in communications because the subscriber does not alert the publisher when such errors occur. The main idea behind this methodology is to find the conditions when PDUs are discarded or degraded, and then recreate one of these conditions. Here are the steps needed to achieve this:

(1) **Define the DoS conditions**, i.e. the conditions that cause PDUs to be discarded, ignored or degraded, based on the specifications of the protocol;

(2) **Define the variables used in the DoS conditions**, including data held in the PDUs, configuration parameters, time of arrival of PDUs, etc. These variables are potential targets for alteration to cause a DoS condition to become true;

(3) For each variable defined, **select which of five cyberattacks are applicable to alter the variable**. We consider injection, replay, drop, delay, and modification attacks;

(4) Using the cyberattacks found, **calculate inputs that recreate any DoS condition**. If an input is found, it is a DoS attack;

(5) **Analyze the plausibility** of the attacks found.

Define the DoS conditions At this step, we look for the known situations where a subscriber drops a given PDU or where the data in the PDU is of bad quality. In the case of GOOSE and SV, these conditions are usually described as text in the IEC standards. Actual implementations of the protocols might also have their own DoS conditions for situations not explicitly covered by the IEC.

Define the variables used in the DoS conditions As part of defining the DoS conditions, the relevant variables also have to be defined as they are the potential targets for the attacker. Variables include characteristics of the target PDU, arrival time of PDUs, current state of the subscriber, and more.

Select cyberattacks that can alter a variable We now look into how an attacker can alter the variables used in the DoS conditions. We consider 5 cyberattacks among the 7 described in Section 5.3.1 to influence PDUs:

(1) Injection: Create a malicious PDU and send it to the subscriber;

(2) Replay: Record contents of a legitimate PDU, then send it later to the subscriber. It is the same as injection, but the contents must have been previously created by the legitimate sender;

Table 5.2: Cyberattacks against network communications and what they affect

Attack	Target variables	Requires keys	Requires MitM
Injection	Previous PDUs' characteristics	Yes	No
Replay	Previous PDUs' characteristics	No	No
Drop	Previous PDUs' characteristics	No	Yes
Delay	Arrival time of PDUs (increase only)	No	Yes
Modification	Any PDU's characteristics	Yes, if PDU must be valid	Yes

(3) Drop: Prevent a legitimate PDU from reaching the subscriber;

(4) Delay: Slow down a legitimate PDU such that it reaches the subscriber later than expected. We assume that we cannot speed up a PDU, since GOOSE and SV are high-priority messages and are already forwarded by switches very quickly;

(5) Modification: Change the contents of a legitimate PDU.

These attacks vary in terms of which variables they can affect as well as their requirements. A summary of what the cyberattacks can affect and their requirements is shown in Table 5.2. Injection, replay and drop attacks can be used to change the previous PDUs (before the target message) received by the subscriber. This can affect the latter's state. Delay attacks apply to variables that are time-based, usually arrival times. Modification attacks are applicable for fields or properties of PDUs themselves, excluding arrival times. Since injection and modification attacks involve creating a PDU that the legitimate publisher likely never created nor signed, we assume that they cannot produce valid encrypted or digitally signed data without knowing the required key. On the other hand, modifying a PDU to render it invalid (essentially corrupting the PDU) is possible without knowing any key. Finally, injection and replay attacks are the only ones that do not require a MitM attacker. They do not intercept the original traffic in any way, as they only add to it. The main distinction between the injection and replay is that the contents of a replayed PDU is restricted to what the publisher has sent before. We do not consider the possibility of sniffing PDUs because it cannot change the variables in DoS conditions. We do not consider the possibility of flooding the network with PDUs, since flooding does not rely on PDU contents.

Calculate inputs that recreate any DoS condition Based on the list of DoS conditions defined at the start, and the possible attacks on variables as found by the previous step, we can look for the inputs that cause one of the DoS conditions to become true. Every input (or range of similar inputs) found is a theoretical cyberattack against the protocol.

Analyze the plausibility Applying this methodology as is allows us to find a list of possible cyberattacks. However, we have not yet considered the difficulty in carrying them out. In some cases, the DoS condition only becomes true in very rare situations (such as a variable overflowing) or the requirements for the attack are more difficult to fulfill.

Overall, this methodology can be applied to any implementation of GOOSE or SV. All that is needed is to update the DoS conditions to match the behavior of the implementation to be analyzed. The DoS conditions are the most critical part of this method as the rest depends on them. In Sections 5.3.3 and 5.3.4, we apply this methodology to the GOOSE and SV protocols as defined by IEC standards to find several cyberattacks unique to these protocols.

5.3.3 Design of Attacks on GOOSE Protocol

In order to apply the methodology of Section 5.3.2 to the GOOSE protocol, we first need to define the behavior of the latter, because attacks against the protocol depend on exploiting some of its unique characteristics. In practice, its behavior actually differs between implementations of the protocol, which do not necessarily follow the guidelines of IEC standards [70]. The methodology can be used regardless of the implementation considered, as it allows for the addition of any DoS conditions.

Definition of DoS Conditions and Variables for GOOSE

We apply the first steps of the methodology defined in Section 5.3.2. We define the DoS conditions applicable to GOOSE based on the behavior defined by IEC. The names and numbers of the conditions are referred to in later sections. We refer to the target message as pdu_i ($i > 0$) as it is the i^{th} legitimate message or PDU. We define the DoS conditions in Table 5.3.

From these conditions, we find the variables in Table 5.4.

Table 5.3: DoS conditions for GOOSE based on IEC standards

#	Name	DoS condition
1	Invalid length	$len(pdu_i) \neq length_i \rightarrow$ discard
2	Lower $stNum$	$stNum_i < stNum_{i-1} \wedge stNum_{i-1} \neq 2^{32} - 1 \rightarrow$ discard
3	Skew period	$stNum_i > stNum_{i-1} \wedge t_i + skew < time_i \rightarrow$ discard
4	TAL expiration	$time_{i-1} + TAL_{i-1} < time_i \rightarrow$ questionable, timeout
5	Invalid $confRev$	$confRev_i \neq confRev_e \rightarrow$ discard
6	$test$ flag is on	$testEna = true \wedge test_{i-1} = true \wedge test_i = false \rightarrow$ discard
7	Invalid signature	$authEna = true \wedge sig(pdu_i) \neq AV_i \rightarrow$ discard

Table 5.4: Variables used in DoS conditions for GOOSE

Variable	Meaning	Inject/replay/drop	Delay	Modify
$confRev_e$	Expected $confRev$, configured			
$skew$	Skew period, configured			
$testEna$	Whether tests are enabled, configured			
$authEna$	Whether $AuthenticationValue$ is enabled, configured			
$stNum_{i-1}$	Previously received PDU's $stNum$	✓		✓
$test_{i-1}$	Previously received PDU's $test$	✓		✓
TAL_{i-1}	Previously received PDU's TAL	✓		✓
$time_{i-1}$	Arrival time of previously received PDU	✓	✓	
$length_i$	Current PDU's $length$ (the field)			✓
$stNum_i$	Current PDU's $stNum$			✓
t_i	Current PDU's t			✓
$confRev_i$	Current PDU's $confRev$			✓
$test_i$	Current PDU's $test$			✓
AV_i	Current PDU's $AuthenticationValue$			✓
$len(pdu_i)$	Actual length of current PDU			✓
$sig(pdu_i)$	Digital signature as computed from current PDU's contents			✓
$time_i$	Arrival time of current PDU		✓	

Determining Cyberattacks to Alter Variables

Using the rules defined in Section 5.3.2, we indicate the possible attacks next to the variables in Table 5.4.

Calculation of Inputs to Recreate Any DoS Condition

Looking at each DoS condition and based on the possible attacks found, we can find the following inputs to fulfill a DoS condition:

(1) **Invalid length**: $len(pdu_i) \neq length_i \rightarrow$ discard

- Modify $len(pdu_i)$;
- Modify $length_i$.

(2) **Lower** $stNum$ **discard**: $stNum_i < stNum_{i-1} \wedge stNum_{i-1} \neq 2^{32} - 1 \rightarrow$ discard

- Modify $stNum_i$ to decrease;
- Inject/replay/modify $stNum_{i-1}$ to increase;
- Drop $stNum_{i-1}$ if $= 2^{32} - 1$ to prevent reset by overflow (this forces $stNum_{i-1} \neq 2^{32} - 1$ to remain $true$).

(3) **Outside skew period**: $stNum_i > stNum_{i-1} \wedge t_i + skew < time_i \rightarrow$ discard

- Modify t_i to make it older;
- Delay $time_i$ to make it newer.

(4) **TAL expiration**: $time_{i-1} + TAL_{i-1} < time_i \rightarrow$ questionable, timeout

- Inject/replay/modify TAL_{i-1} to decrease;
- Drop $time_{i-1}$ to make it older;
- Delay $time_i$ to make it newer.

(5) **Invalid** $confRev$: $confRev_i \neq confRev_{expect} \rightarrow$ discard

- Modify $confRev_i$.

(6) **Test flag is on**: $testEna = true \wedge test_{i-1} = true \wedge test_i = false \rightarrow$ discard

- Inject/replay/modify $test_{i-1}$ to make it $true$;
- Drop $test_{i-1}$ to make it $true$ (unnecessary, as it will already be $true$);
- Modify $test_i$ to make it $false$.

(7) **Invalid signature**: $authEna = true \wedge sig(pdu_i) \neq AV_i \rightarrow$ discard

- Modify anything in pdu_i;
- Modify AV_i to invalidate.

Any input that turns a DoS condition true represents a theoretical DoS attack. Ultimately, applying the methodology results in a set of 22 DoS attacks against the GOOSE protocol as specified by IEC standards. They are shown in Table 5.5. We label the attacks from G1 to G22 to make it easier to reference them. Note that this set includes attacks already described in the literature (G4 in [62], G5 and G14 in [42]). To the best of our knowledge, the remaining attacks are not yet documented. Notably, G8 and G9 exploit the skew filtering, G10 and G11 are variants of G14 that do not require MitM capabilities, and G16 to G20 target the $test$ field. This demonstrates the effectiveness of this methodology at deriving attacks systematically. We next analyze the attacks to assess their practical use.

Table 5.5: DoS attacks on GOOSE and their requirements

ID	Attack	#	Need keys	Need MitM
G1	Modify $len(pdu_i)$	1	No	Yes
G2	Modify $length_i$	1	Yes	Yes
G3	Modify $stNum_i$ to decrease	2	Yes	Yes
G4	Inject $stNum_{i-1}$ to increase	2	Yes	No
G5	Replay $stNum_{i-1}$ to increase[1]	2	No	No
G6	Modify $stNum_{i-1}$ to increase	2	Yes	Yes
G7	Drop $stNum_{i-1}$ to prevent overflow[2]	2	No	Yes
G8	Modify t_i to make it older	3	Yes	Yes
G9	Delay $time_i$ to make it newer[3]	3	No	Yes
G10	Inject TAL_{i-1} to decrease	4	Yes	No
G11	Replay TAL_{i-1} to decrease[4]	4	No	No
G12	Modify TAL_{i-1} to decrease	4	Yes	Yes
G13	Drop $time_{i-1}$ to make it older	4	No	Yes
G14	Delay $time_i$ to make it newer[5]	4	No	Yes
G15	Modify $confRev_i$	5	Yes	Yes
G16	Inject $test_{i-1}$ to make it $true$	6	Yes	No
G17	Replay $test_{i-1}$ to make it $true$[4]	6	No	No
G18	Modify $test_{i-1}$ to make it $true$	6	Yes	Yes
G19	Drop $test_{i-1}$ to make it $true$	6	No	Yes
G20	Modify $test_i$ to make it $false$	6	Yes	Yes
G21	Modify anything in $apdu_i$	7	No	Yes
G22	Modify AV_i to invalidate	7	No	Yes

[1] Requires $stNum$ reset (somewhat rare) and must be done within skew period
[2] Overflow of $stNum$ is extremely rare
[3] Small delay between many PDUs to exceed skew period
[4] Must be done within skew period
[5] Large delay between two PDUs to cause TAL expiration

Analysis of Plausibility

Table 5.5 shows that several of the DoS attacks on GOOSE, namely ones relying on modification and injection of PDUs, are mitigated by the use of $AuthenticationValue$ of IEC 62351-6, since its presence forces the attacker to acquire the required signature key before performing any modification or injection that avoids triggering DoS condition #7 (invalid signature). However, because we do not make assumptions as to whether the digital signatures are usable in the substation or whether the attacker has compromised the keys, attacks marked as requiring the keys are still considered plausible in this work. These attacks cause discarding of messages for different reasons (G15 targets $confRev$, G8 targets the t field, etc.) and can require different detection methods as a result. Digital signatures do not prevent attacks such as delay, drops, or modifications of the PDU length (G1) or of the signature itself (G21, G22), but these are not in the scope of IEC 62351-6 and must be addressed using other means.

Attacks on $stNum$ (G3 to G7) Of particular concern are the attacks that affect DoS condition #2, the discarding of lower $stNum$. The value of $stNum$ is critical for synchronizing the publisher and subscribers. If a higher $stNum$ value is sent to the subscriber (G4 to G6), the synchronization is thrown off and future messages pdu_{i+1}, pdu_{i+2}, ... are also likely to be discarded. In fact, G4 (named "GOOSE poisoning" by Kush *et al.* [62]) has been shown to be very effective using a single forged PDU. As it does not require MitM and has devastating effects, this is a major threat to be addressed. Attack G5 is a variant of G4 that is described in a conference paper by Strobel *et al.* [42] and that circumvents the protection offered by signatures through replaying a legitimate PDU with a higher $stNum$. The attack described in the paper has the attacker reset the $stNum$ to 0 by either waiting for an $stNum$ overflow, which is very rare but not impossible if the substation remains active for years, or by delaying PDUs to force a TAL expiration (see G14) [42]. The latter method relies on the assumption that *both* publisher and subscriber reset their $stNum$ to 0 on TAL expirations. However, it is unclear how this can be achieved when considering that the subscribers cannot warn the publisher that they noticed a delay. Thus, it is not known if delays meant to cause $stNum$ resets are practical. The specification does not explicitly state if the reset applies to only to the subscriber, or to the subscriber and publisher:

> "If there is a message timeout, the starting Stnum [of 0] shall be re-established. If Stnum rolls-over, the starting Stnum [of 0] shall be re-established."[28]

Delay (G9, G14) and replay (G5, G11, G17) attacks There are two delay attacks, G9 and G14, that are executed similarly but target different weaknesses of GOOSE. G14 aims to deliberately cause a TAL expiration, as explained before. G9 targets skew filtering instead. Skew filtering is introduced in IEC 62351-6 to address replay attacks [28], as described in Section 2.5.4, yet G5, G11 and G17 rely on replay and can be executed if done within the skew period (of 10 to 120 seconds). G9 exploits the filtering by delaying PDUs until they are considered outside the skew period, causing message drops. Arguably, exceeding the skew period might not be required, as a latency of several seconds between publisher and subscriber can be sufficient to have major consequences. One might ask how such a large delay can be introduced in the communications without causing a TAL expiration like G14 does. The answer is to simply add a small delay between *many* pairs of PDUs. In other words, one can slightly increase the difference between $time_{i-1}$ and $time_i$ for many messages. By doing that repeatedly, this builds up a large total delay without triggering a TAL timeout, because the time between any two successive PDUs is always within the TAL specified by the last PDU.

Toggling $test$ flag (G16 to G20) These attacks aim to trigger DoS condition #6 by targeting the $test$ flag. They can be quite damaging since a single PDU with the True flag is needed to theoretically cause all next PDUs to be discarded. However, this is completely reliant on whether IED accepts test values. Presumably, a correctly configured IED that is deployed should not have this feature enabled, hence the attack relies on human error. We also consider that while we found G19 by applying this methodology, it is not an effective attack in practice, since any PDU with a $false\ test$ flag are already discarded if they are ever preceded by a PDU with a True $test$ flag. Dropping them has no impact as a result.

5.3.4 Design of Attacks on SV Protocol

The SV protocol is very similar to GOOSE, though it has simpler behavior than the latter. It is an interesting target for a DoS attack as it provides the measurement data that many IEDs rely on. We apply the methodology from Section 5.3.2 to SV communications to find potential DoS attacks.

Definition of DoS Conditions and Variables for SV

We define the DoS conditions and related variables based on the behavior of the SV protocol as described by IEC standards. The DoS conditions are listed in Table 5.6.

Table 5.6: DoS conditions for SV based on IEC standards

#	Name	DoS condition
1	Invalid length	$len(pdu_i) \neq length_i \rightarrow$ discard
2	Skew period	$timestamp_i + skew < time_i \rightarrow$ discard
3	Invalid signature	$authEna = true \wedge sig(pdu_i) \neq AV_i \rightarrow$ discard
4	Lower $smpCnt$	$smpCnt_i < smpCnt_{i-1} \wedge \neg reset \rightarrow$ discard
		$sync \vee (smpCnt_{i-1} = 2^{16} - 1) \rightarrow$ reset

The variables used in the DoS conditions are listed in Table 5.7.

Table 5.7: Variables used in DoS conditions for SV

Variable	Meaning	Inject/replay/drop	Delay	Modify
$skew$	Skew period, configured			
$reset$	Whether a reset occurred			
$sync$	Whether a sync pulse occurred			
$authEna$	Whether $AuthenticationValue$ is enabled, configured			
$smpCnt_{i-1}$:	Previously received ASDU's $smpCnt$	✓		✓
$length_i$	Current PDU's $length$ (the field)			✓
AV_i	Current PDU's $AuthenticationValue$			✓
$len(pdu_i)$	Actual length of current PDU			✓
$sig(pdu_i)$	Digital signature as computed from current PDU's contents			✓
$time_i$	Arrival time of current PDU		✓	
$smpCnt_i$	Current ASDU's $smpCnt$			✓
$timestamp_i$	Current ASDU's $timestamp$			✓

Determining Cyberattacks to Alter Variables

Using the rules as explained in Section 5.3.2, we locate the variables that can be altered using the following cyberattacks and mark them in Table 5.7.

Calculation of Inputs to Recreate Any DoS Condition

The following are inputs to fulfill a DoS condition:

(1) **Invalid length**: $len(pdu_i) \neq length_i \rightarrow$ discard

- Modify $len(pdu_i)$;
- Modify $length_i$.

(2) **Skew period**: $timestamp_i + skew < time_i \rightarrow$ discard

- Modify $timestamp_i$ to make it older;
- Delay $time_i$ to make it newer.

(3) **Invalid signature**: $authEna = true \wedge sig(pdu_i) \neq AV_i \rightarrow$ discard

- Modify anything in pdu_i;
- Modify AV_i to invalidate.

(4) **Lower** $smpCnt$ **discard**: $smpCnt_i < smpCnt_{i-1} \wedge \neg reset \rightarrow$ discard

- Modify $smpCnt_i$ to decrease;
- Inject/replay $smpCnt_{i-1}$ to increase;
- Modify $smpCnt_{i-1}$ to increase;
- Force $reset$ to stay at $false$ (see next).

(5) **Lower** $smpCnt$ **discard (reset)**: $sync \vee (smpCnt_{i-1} = 2^{16} - 1) \rightarrow$ reset

- Drop $smpCnt_{i-1}$ if $= 2^{16} - 1$ to prevent reset by overflow.

We summarize the 11 possible DoS attacks against SV in Table 5.8 and label them S1 to S11.

Table 5.8: DoS attacks on SV and their requirements

ID	Attack	#	Need keys	Need MitM
S1	Modify $len(apdu_i)$	1	No	Yes
S2	Modify $length_i$	1	Yes	Yes
S3	Modify $timestamp_i$ to make it older	2	Yes	Yes
S4	Delay $time_i$ to make it newer	2	No	Yes
S5	Modify anything in $apdu_i$	3	No	Yes
S6	Modify AV_i to invalidate	3	No	Yes
S7	Modify $smpCnt_i$ to decrease	4	Yes	Yes
S8	Inject $smpCnt_{i-1}$ to increase	4	Yes	No
S9	Replay $smpCnt_{i-1}$ to increase[1]	4	No	No
S10	Modify $smpCnt_{i-1}$ to increase	4	Yes	Yes
S11	Drop $smpCnt_{i-1}$ if $= 2^{16} - 1$ to prevent reset by overflow[2]	4	No	Yes

[1] Requires $smpCnt$ reset (very frequent) and must be done within skew period
[2] Many opportunities due to frequent overflows, but does not work if sync pulses are used

Analysis of Plausibility

The DoS attacks found on SV resemble ones that target GOOSE. S1, S2, S5 and S6, involving modification of the PDU length and the digital signature, are identical to G1, G2, G21 and G22 which are discussed in Section 5.3.3. The remaining attacks have noticeable differences from their GOOSE counterparts due to the unique behavior of SV.

Attacks on $smpCnt$ The $smpCnt$, used to order the SV PDUs received by the subscriber [24], is exploitable in S7 to S11, since SV subscribers are supposed to drop ASDUs holding a lower $smpCnt$ than the last one received. A single execution of an attack that increases the $smpCnt$ at the subscriber is sufficient to block communications, at least until the next sync pulse. Attack S9, which relies on replay after a $smpCnt$ reset, is notable for not requiring the signature key nor MitM capabilities. Publisher and subscribers of SV reset $smpCnt$ when they receive sync pulses [26], which happen typically every second [29]. Resets also happen when the $smpCnt$ overflows. This can occur frequently, as it increases by thousands per second, while $smpCnt$ cannot exceed 65,535 ($2^{16} - 1$) [26]. The frequent resets render the skew filtering of IEC 62351-6 [28] as described in Section 2.5.4 less effective in preventing S9. An attacker can easily record a legitimate PDU with a higher $smpCnt$, wait a few seconds for the inevitable next $smpCnt$ reset, and replay the PDU while remaining within the skew period of 2 minutes. Attack S11 involves dropping select PDUs to prevent resets by overflow of $smpCnt$ and also works despite signatures. However, it cannot be

executed if resets by sync pulses are enabled, as the latter allow the subscriber to reset the $smpCnt$ on its own.

Skew period Attacks S3 and S4 exploit the skew period itself. By delaying SV PDUs for over 2 minutes, the subscriber drops the SV traffic. Unlike in GOOSE, there are no explicitly stated limits on how much time can pass between messages. The attacker does not need to worry about message timeouts. She can simply hold back the traffic for 2 minutes before sending it to trigger DoS condition #4.

5.3.5 Design of Attacks on IEC 61850 MMS Protocol

Differences from GOOSE and SV

Possible cyberattacks on IEC 61850 MMS differ from those possible on GOOSE and SV. MMS uses a client-server model, not a publisher-subscriber model. Causing messages to be discarded or dropped at the receiving end of the communication, similarly to what is done in the methodology presented in Section 5.3.2, is not stealthy as the recipient has the capability to send error PDUs in response. IEC 61850 MMS also provides a wide variety of services. As a result, there are many possible MMS PDUs and scenarios to consider. Due to the complexity of IEC 61850 MMS, we limit ourselves to general attacks against the MMS protocol.

Security Assessment of IEC 61850 MMS

IEC 61850 MMS is potentially vulnerable to attacks that target MMS as defined in ISO 9506. Incidentally, MMS as originally designed does not provide sufficient security. At best, it supports access control lists that identify the clients allowed to read or modify a given MMS variable [32]. IEC 62351-4 specifies using TLS to prevent several attacks, though it applies only to the transport layer [41]. In cases where a device does not support TLS or where multiple TLS connections are needed to forward a PDU [66], the benefits of TLS are lost. The security measures defined by IEC 62351-4 that apply to the A-Profile do not address these cases. They only provide authentication for the PDUs used during association creation, AARQ and AARE, by including a digital signature in the $SignedValue$ field [41], leaving some gaps [69]:

- $SignedValue$ only takes a timestamp, the $time$ field, as input, ignoring the rest of the PDU's contents. Hence, tampering with other fields does not invalidate it;
- The recipient accepts a PDU if $time$ is within 10 minutes of its local clock, giving attackers a 10-minute time window during which they can attempt to re-use the $SignedValue$. This is mentioned even by the standard itself [41];
- PDUs exchanged after the association is created are not protected.

As such, the security of the A-Profile is limited, leaving MMS to rely almost entirely on TLS for its security. This implies that the part of IEC 62351 concerned with TLS, IEC 62351-3, is the one that provides the most security features for MMS.

List of Attacks Against MMS

The list of potential attacks against MMS depends on whether TLS is in use. TLS is designed to address several threats, including injection, replay, modification of messages, and sniffing. Its presence thus can be sufficient to prevent certain attacks.

If TLS is used TLS already addresses attacks on integrity (injection, replay and modification) and supports encryption to ensure confidentiality of the contents of PDUs. To execute any of these attacks, the attacker is forced to obtain the relevant private keys and encryption keys. The difficulty in accomplishing this varies according to the configured cipher suite used by TLS. A secure cipher suite makes it infeasible for the attacker to brute-force keys to bypass defenses provided by TLS. It should be noted that not all cipher suites allowed by IEC 62351-4 are considered secure [69].

If TLS is not used Without TLS, the security provided by the A-Profile alone is limited. Since the $SignedValue$ remains valid despite changes to PDUs, it can be read from a legitimate PDU, which is not encrypted in this scenario, and copied to a malicious one. The attacker can alter MMS communications by waiting for a legitimate party to send a signed AARQ, sending a malicious AARQ that copies the $SignedValue$, and either dropping the original one or hoping that the malicious AARQ reaches the destination first (due to the requirement that every $SignedValue$ be only

Table 5.9: General attacks on MMS and their requirements

ID	Attack	Works against TLS	Need MitM
M1	Inject PDU[1]	Yes, if key known	No
M2	Replay PDU[2]	No, by design	No
M3	Sniff PDU contents	Yes, if key known	No
M4	Modify PDU	Yes, if key known	Yes
M5	Flood with PDUs	Yes	No
M6	Drop PDU	Yes	Yes
M7	Delay PDU	Yes	Yes
M8	Modify PDU to invalidate	Yes	Yes

[1] Requires sniffing $SignedValue$ from legitimate PDU within 10 minutes
[2] Must be done within 10 minutes

accepted once). Alternatively, the attacker can tamper with PDUs transferred after the connection is open, because they are not protected. Essentially, the only attack the $SignedValue$ prevents is the attacker injecting her own valid PDUs when there has been no MMS communication on the network for over 10 minutes.

In all cases With or without TLS, no measures ensure the availability of MMS. This is expected as the issue of availability is outside the scope of either TLS or IEC 62351 parts 3, 4 and 6. This leaves MMS open to a variety of DoS attacks, including flooding attacks, dropping PDUs, delaying PDUs, and modifying legitimate PDUs to render them invalid. Table 5.9 contains the list of possible attacks found on MMS.

5.4 Attack Execution in Co-simulation Testbed

We discuss in this section the cyberattacks that we can execute in practice using our testbed, presented in Chapter 4.

5.4.1 Selection of Attacks to Execute

Analysis of IEC 61850 Protocols on Testbed

The behavior of protocols such as GOOSE and SV does not always match the requirements of IEC 61850 and IEC 62351, depending on the implementation [70]. The analysis in this work is

done on behavior as defined by IEC as it should be the most widespread, given that it is standardized. The implementation we use has somewhat different behavior, causing some of the attacks to work differently or not be applicable. If an attack depends on behavior that is not implemented in the testbed, it can often still be executed and tested against NSM. It is the impact on the physical system that might differ.

We start by determining which of the DoS conditions for GOOSE are applied in the testbed. To do so, we execute several attacks that modify the variables in the DoS conditions in an attempt to make the conditions true, and note the behavior of the testbed in response. We need to execute a representative subset of attacks. We summarize the results from executing all of these tests in Table 5.10. In summary, the majority of changes to variables do not produce the result expected when referring to the IEC standards. Some of the variables are absent altogether.

Table 5.10: Results of tests on GOOSE ran on testbed

Test	Expected result from IEC	Result in testbed
Modify $length_i$	PDU discarded	PDU always accepted If $length_i > len(pdu_i)$, Wireshark finds malformed PDU
Modify TAL_{i-1} to decrease	Subscriber state is Questionable, $stNum$ resets to 0	No noticeable change
Modify t_i to make it older	If $time_i - t_i > 2$ min., PDU with higher $stNum$ discarded	PDU always accepted
Modify $stNum_{i-1}$ to increase	PDU discarded if $stNum_i < stNum_{i-1}$, except on overflow	PDU always discarded if $stNum_i < stNum_{i-1}$
Modify $test_{i-1}$ to make it $true$	PDUs with True $test$ flag are discarded	PDU always accepted
Modify $confRev_i$	PDU discarded	PDU always accepted
Overflow $stNum_{i-1}$	PDU always accepted	PDU always discarded
Modify $len(pdu_i)$	PDU discarded	If PDU truncated: PDU data is unused ($allData$ is deleted) and Wireshark finds malformed PDU If PDU padded: PDU always accepted
Delay $time_i$ to make it newer	If $time_i - t_i > 2$ min., PDU with higher $stNum$ discarded	If $time_i - t_i > 2$ min., PDU accepted Delay of < 2 min. works as expected

We also do the tests for SV. The results are shown in Table 5.11. Much like for GOOSE, the DoS conditions work differently than as dictated by IEC standards.

Table 5.11: Results of tests on SV ran on testbed

Test	Expected result from IEC	Result in testbed
Modify $length_i$	PDU discarded	PDU always accepted
Modify $smpCnt_{i-1}$ to increase	PDU discarded if $smpCnt_i < smpCnt_{i-1}$, except on overflow	PDU always accepted
Modify $len(pdu_i)$	PDU discarded	If PDU truncated: quality of data degrades and Wireshark finds malformed PDU If PDU padded: PDU always accepted

Table 5.12: DoS conditions for GOOSE and their applicability to the testbed

#	Name	Applicability to testbed
1	Invalid length	Does not work as PDU is accepted, but decreasing $len(pdu_i)$ greatly affects the data quality
2	Lower $stNum$	Works as expected until $stNum$ reaches $2^{32} - 1$, because it never resets to 0
3	Skew period	N/A: no skew period, and t is present but ignored
4	TAL expiration	Does not work, $stNum$ does not reset to 0
5	Invalid $confRev$	Does not work, $confRev$ is present but ignored
6	$test$ flag is on	Does not work, $test$ flag is present but ignored
7	Invalid signature	N/A: No $AuthenticationValue$

Analysis of GOOSE on testbed Based on the tests performed previously, Table 5.12 shows how the DoS conditions based on IEC standards compare to the implementation on the testbed. As shown, there are only two DoS conditions that apply to the testbed (the first two), and even those have small differences. The testbed only accepts PDUs with a $stNum$ that is equal or higher than the current one, but it does not handle the admittedly rare case of $stNum$ overflows. The testbed also accepts PDUs that have the wrong length. In the case of truncated PDUs, though, it cannot use the data because truncation deletes the bytes in the very important $allData$ field, resulting in DoS regardless. Other fields are seemingly not checked at all.

Analysis of SV on testbed Table 5.13 compares DoS conditions based on IEC standards with the SV protocol behavior on the testbed. In short, the conditions (such as checking the $smpCnt$ or length) are either not applied or are non-applicable due to missing fields in the PDUs. SV PDUs that are truncated or padded are marked as errors by Wireshark, but they are accepted by the testbed. Like for GOOSE, the subscriber accepts PDUs with an incorrect length. Even if the PDUs is truncated,

Table 5.13: DoS conditions for SV and their applicability to the testbed

#	Name	Applicability to testbed
1	Invalid length	Does not work as PDU is accepted, but decreasing $len(pdu_i)$ greatly affects the data quality
2	Skew period	N/A: No skew period or $timestamp$ field
3	Invalid signature	N/A: No $AuthenticationValue$
4	Lower $smpCnt$	Does not work, $smpCnt$ is present but ignored

the subscriber reads whatever bytes are remaining in the $sample$ field, resulting in incomplete data. In our experience, the attacker must delete most of or all of the bytes in $sample$ to significantly affect the testbed.

Analysis of MMS on testbed The testbed does not provide support for IEC 61850 MMS, so none of them can be tested.

Potential Attacks on IEC 61850 Protocols on Testbed

Given the limitations outlined previously, there are 8 out of 21 attacks on GOOSE and 4 out of 11 attacks on SV that are not applicable to our testbed. Attacks G5, G7, and S11 rely on $stNum$ or $smpCnt$ overflows, which do not occur in our specific implementation. Attack G11 is ineffective because TAL is always constant. Similarly, G17, G19 and G20 do not work as the $test$ flag is always $false$. Attacks G21, G22, S5 and S6 rely on the absent $AuthenticationValue$ field, while S3 relies on the similarly missing $timestamp$. The remaining attacks can be executed properly.

5.4.2 Execution of Attack in Testbed

To execute the attacks on the testbed, we set up a MitM attacker to intercept the traffic going from a given relay to its subscribed CB. We have to do so in a way that the traffic is only intercepted after it has been analyzed by the relay's NSM proxy, but before it has been analyzed by the CB's NSM proxy. This is to reflect the case where the NSM agents are running in the hosts themselves.

For each attack to be carried out, we tweak the settings of the substation switch to give it MitM capabilities and have it execute the attack in question. The switch can analyze PDUs and take actions under certain conditions to inject, modify, delay or drop specific PDUs. We can toggle the

switch's malicious behavior on and off as needed. We can do this using a web application shown in Figure 5.6. While the malicious switch is active, the NSM manager continues to collect data as normal. The data retrieved is then processed by our analytics engine to determine whether an attack is under way or not.

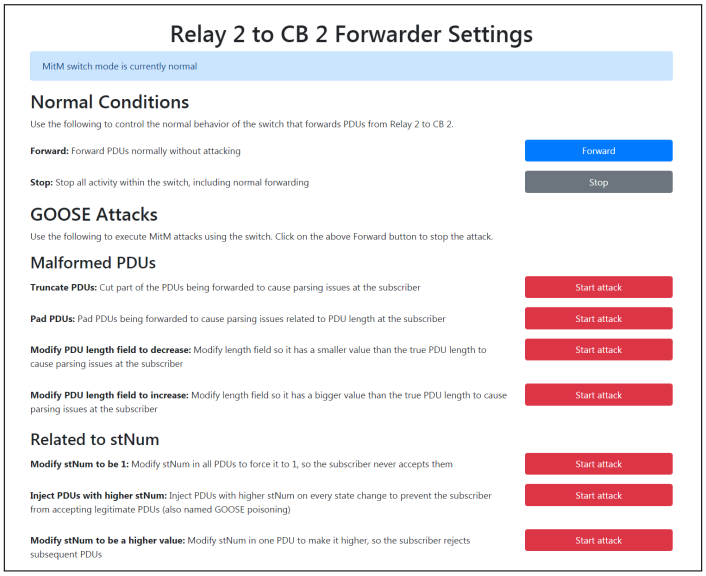

Figure 5.6: Web application used to toggle attacks in MitM switch

Using the HYPERSIM interfaces, we can view the physical effect of the attack on the power system simulation. As an example, we show the effects of the GOOSE delay attack G14 on GOOSE communications between Relay_2 and CB2 in Figure 5.7. While executing the attack, the distance between the publisher's $sqNum$ (in blue) and the subscriber's $sqNum$ (in red) grows over time, representing how the subscriber is receiving PDUs that are becoming more and more outdated. There is thus a significant lag buildup over time.

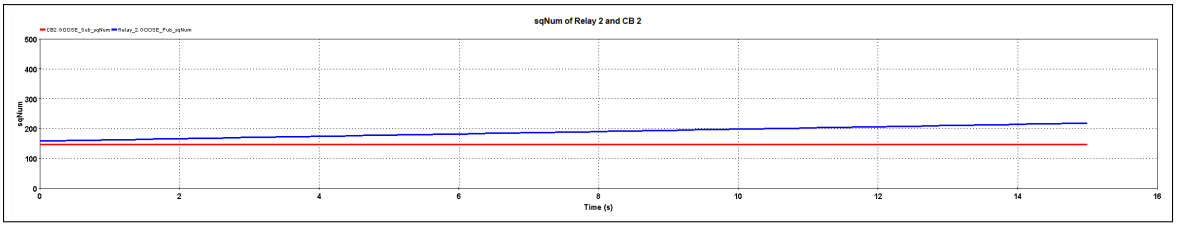

Figure 5.7: HYPERSIM interface showing divergence of $sqNum$ values during GOOSE delay attack (G14)

We are interested in what occurs if a fault is introduced while the attack is ongoing. We showed

previously in Figure 4.6 that in our setup, CBs 1 and 2 should trip and become open as a response to this particular fault. However, if we repeat the experiment while the system is under a delay attack, the result is different. We show it in Figure 5.8. Here, Relay_2's messages to CB2 arrive late and the latter reacts several seconds later than it should (see column 2). It takes so long to open the breaker that CBs 4 and 5 (in the last two columns) are forced to trip to prevent instability, causing a blackout. This illustrates the potential devastating consequences of a delay attack within a substation.

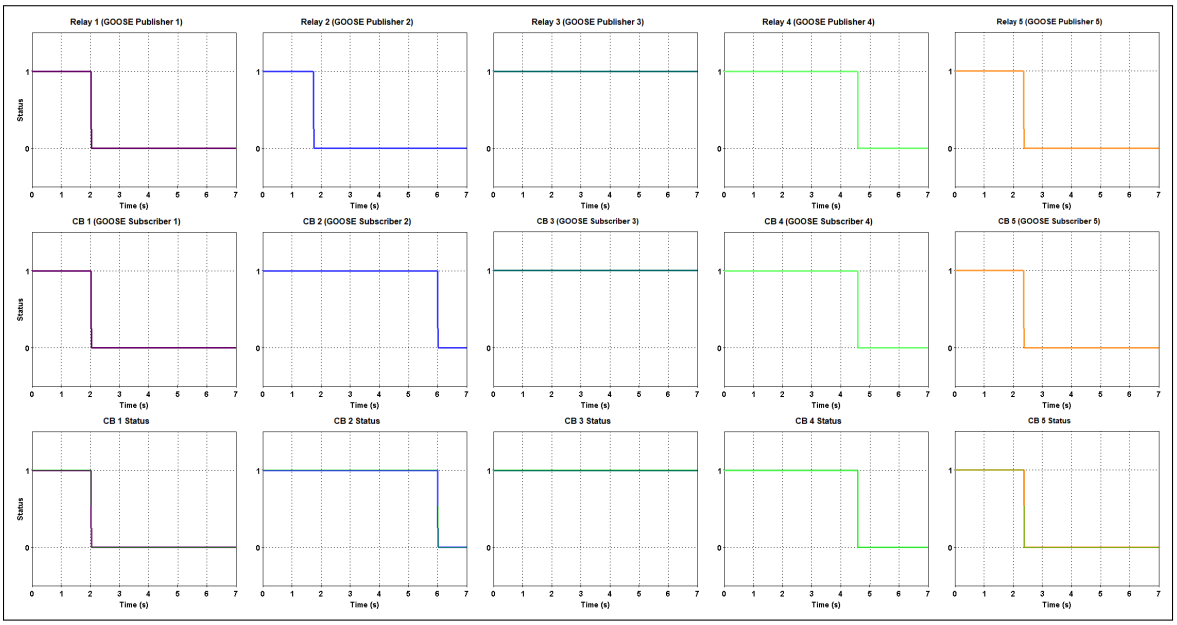

Figure 5.8: HYPERSIM interface showing physical impact of GOOSE delay attack (G14)

5.5 Detection of Attack Using NSM Data Objects

Our hypothesis is that we can detect some of the attacks from Table 5.15 by using the NSM DOs named in Section 4.1.2. This section outlines the NSM DOs and algorithms used to detect the attacks and which attacks avoid detection by NSM alone. We do not make use of other security measures along with NSM, because we want to evaluate the capabilities of NSM by itself.

5.5.1 Rule-based Detection for GOOSE and SV

There are many attacks that can be monitored by a rule that uses the values from NSM DOs. For example, if we consider that there should be no communication errors in the network under normal

conditions, then we can expect that a NSM DO that counts errors will never change unless an attack or problem occurs. All that is needed is to alert any time this value changes. We outline in this section the NSM DOs that can be used in rule-based detection.

GOOSE

InErrCnt (count of incoming errors) This NSM DO counts the number of received PDUs in error. Malformed PDUs or configuration mismatches count as errors, according to the standard [8]. In our implementation of this NSM DO in the proxies, we verify five aspects of the PDU: whether the $length$ field matches the actual length of the contents, whether the $APPID$ is within the legal range, whether the $confRev$ value (related to configuration) is the one expected, whether the tags (fields) expected in a GOOSE PDU are present and in the right order, and whether the PDU is successfully parsed. We can therefore use InErrCnt to count errors related to the length of the PDU and issues with $confRev$. Because this NSM DO only counts errors, we expect that it does not change in normal working conditions, allowing us to detect attacks on any change to this value.

How to use: Any change to this value indicates an anomaly

Attacks detected: G1, G2 and G15

TalExpCnt (count of TAL expirations) This NSM DOs counts the number of TAL expirations, which occur whenever there is a time gap between PDUs that is higher than the TAL specified in the publisher's last PDU. Using this NSM DO helps detect attacks that involve changing the TAL field or delaying a PDU to deliberately cause an expiration. However, we do not expect that it can capture changes to TAL that do not result in an expiration.

How to use: Any change to this value indicates an anomaly

Attacks detected: G10 to G13

ConfRevMis (configuration revision mismatch) This NSM DO tracks whether the values of $confRev$ in incoming PDUs matches what is expected by the subscriber. In our implementation, we set this to True when there is a discrepancy in the $confRev$ for the last PDU received, and to False when there is not. PDUs with bad $confRev$ values are also counted in InErrCnt, so there

is some redundancy in the NSM DOs. The main distinction is that ConfRevMis indicates whether there is a problem at the present moment, while InErrCnt indicates past events.

How to use: If this value is set to True, there is an anomaly

Attacks detected: G15

SV

In our implementation, we do not have access to any NSM DO that we can use in rule-based detection to detect attacks on SV. There are fewer NSM DOs specific to SV when compared to GOOSE. The most notable ones are PDUSizeFail, MessageIntegrityFailCnt and DecryptFailCnt. None of them are applicable to the testbed, because we neither use SV over IP (required for PDUSizeFail) nor the $AuthenticationValue$ or encryption that could provide integrity and confidentiality.

5.5.2 Anomaly Detection for GOOSE and SV

The work on anomaly detection is a joint work with Mr. Abdullah Albarakati and Dr. Rachid Hadjidj.

Anomaly detection is performed using LSTM models. The NSM DOs can be used as input to machine learning algorithms and train models capable of real-time detection.

TxPduPerSecond and RxPduPerSecond (transferred and received PDUs per second) These NSM DOs track the number PDUs per second that are sent by the publisher or received by the subscriber respectively. While it is unclear from the IEC 62351-7 standard alone how these values are meant to be calculated, in this work, we use it to store the average number of PDUs sent or received within the last 20 seconds. Assuming this interpretation is close to the original intent, then we can use these NSM DOs to potentially detect any attack that alters the expected amount of traffic at a given time. Such attacks are injection, delay or drop attacks. Because traffic can vary in normal conditions, anomaly detection is used to discern normal behavior of the system from anomalous behavior. This also means that an attack that does not cause a significant deviation from normal values is unlikely to be detected using these NSM DOs.

How to use: A deviation from normal expected values indicates an anomaly

Attacks detected: G4, G9, G10, G13, G16, S4, S8, S9, S11

5.5.3 Detection for MMS

While we cannot carry out experiments on MMS, we study the potential of NSM DOs at detecting attacks on this protocol. We associate in Table 5.14 the NSM DOs that can detect the general attacks against MMS as defined in Table 5.9.

Table 5.14: NSM DOs to be used to detect attacks on MMS

ID	Attack	NSM DOs for detection	Addressed by IEC 62351-3, -4
M1	Inject PDU	Count of PDUs sent and received by type	Yes
M2	Replay PDU	Count of PDUs sent and received by type	Yes
M3	Sniff PDU contents		Yes
M4	Modify PDU		Yes
M5	Flood with PDUs	Count of PDUs sent and received by type	No
M6	Drop PDU	Count of PDUs sent and received by type, count of unacknowledged requests, rate of reports received	No
M7	Delay PDU	Time between reports, count of PDUs sent and received by type	No
M8	Modify PDU to invalidate	Count of PDUs sent and received by type (errors), count of decryption failures (A- and T-Profiles)	No

From Table 5.14, we can see that the NSM DOs for MMS can potentially detect all variants of DoS attacks that could bypass the defenses of IEC 62351-4. They therefore work as advertised by IEC 62351-7. This is an improvement over the NSM DOs for GOOSE and SV.

The first attack against MMS that NSM DOs are unable to detect in Table 5.14 is attack M3, sniffing PDUs. This is not surprising for a few reasons: first, NSM is not designed to detect passive listening attacks as the hosts it queries for NSM DOs are not affected by them. Second, the real solution to address this is through TLS, which can be used with encryption to greatly increase the difficulty of sniffing for an attacker.

The second attack that cannot be detected by NSM DOs is attack M4, modification of an MMS PDU, which is only possible if TLS is not used or if the attacker knows the required encryption keys for TLS. This is because replacing one valid PDU with another PDU with different contents is likely to affect the NSM DOs the exact same way. The underlying problem that allows for modification

attacks to work is that the A-Profile security fails to prevent them if TLS is not in use. It is likely that solutions to allow for detecting modification attacks on MMS lie outside of NSM.

5.5.4 Attacks without Relevant NSM Data Objects

This leaves some of the attacks without any NSM DOs that can detect them. They are all modification attacks, as these do not modify the amount of traffic on the network, if we ignore the very small delay required to complete the modification. These are G3, G6, G18, G20, S1 to S3, S5 to S7 and S10. In theory, if the digital signature of $AuthenticationValue$ is used, these attacks are detectable as they cause integrity errors counted by MessageIntegrityFailCnt.

5.6 Results

We show in this section the results from running every attack 10 times while the NSM solution is functional and running the detection engine in real-time. We are interested in the accuracy of the NSM detection (i.e. whether it detects an actual attack), if there were any false positives (where the NSM DOs indicated an anomaly when there was no attack), and how long it takes for the NSM solution to detect the attack.

5.6.1 Attacks Detected

GOOSE

G1 and G2 both involve tampering with the length of the PDU in an attempt to invalidate it. Such an attack is easy to detect using NSM because it directly affects the NSM DO InErrCnt. This object counts PDUs in errors, which includes malformed ones [8]. In our implementation, we interpret the definition of a malformed PDU as including ones with inconsistent values in their $length$ fields. We find this is a reasonable assumption and it is consistent with Wireshark's interpretation of such PDUs. The same NSM DO is also used to successfully detect attack G15 that modifies the $confRev$ value. This is because we treat errors with $confRev$ as configuration mismatches, which are counted in InErrCnt. In addition, the NSM DO ConfRevMis monitors the current state of the $confRev$ field as well, leaving us with two indicators to detect this specific attack.

Table 5.15: Attacks on GOOSE and SV to run on the testbed

ID	Attacks detected	False positives	Average time to detect (s)
G1 (truncate)	10/10	0	18.5070
G1 (pad)	10/10	0	19.6262
G2 (decrease)	10/10	0	19.5478
G2 (increase)	10/10	0	17.3790
G3	0/10	0	N/A
G4	0/10	0	N/A
G6	0/10	0	N/A
G8	0/10	0	N/A
G9 (500 ms delay)	5/5	0	28.9075
G10	10/10	0	14.5624
G12	10/10	0	17.9097
G13	10/10	0	15.0932
G14 (10 ms delay)	5/5	0	38.8368
G14 (1 ms delay)	0/5	0	N/A
G15	10/10	0	19.8630
G16	0/10	0	N/A
G18	0/10	0	N/A
S1 (decrease)	0/10	0	N/A
S1 (increase)	1/10	0	45.6158
S2 (decrease)	0/10	1	N/A
S2 (increase)	0/10	0	N/A
S4 (every 10 PDUs)	10/10	1	31.9919
S4 (every 100 PDUs)	1/10	0	35.6277
S4 (every 1,000 PDUs)	0/10	0	N/A
S7	0/10	0	N/A
S8	0/10	0	N/A
S9	0/10	0	N/A
S10	0/10	0	N/A

G14 aims to cause a TAL expiration (DoS condition #4). We execute it by introducing a delay of 500 ms between PDUs, since that is the value of TAL used on the testbed. When executing G14, we have found that NSM is effective at detecting it. The NSM DO TalExpCnt tracks the number of expirations that the subscriber has detected. In normal circumstances, this value should not change, so any increase to this counter is suspicious. Additionally, the NSM DO RxPduPerSecond tracks the rate of PDUs per second, which is altered significantly by introducing such a large delay. Because of these two NSM DOs, there is likely no way for the attacker to cause a TAL expiration without being noticed by the NSM manager. This is true for attacks G10 to G14. While we could not verify this in our experiments, we believe that this also significantly limits the opportunities to execute the replay attack G5 as described by Strobel *et al.* [42]. This is because the attack requires a $stNum$ reset, which can be caused by either a TAL expiration or a (very rare) $stNum$ overflow. Any attempt to

cause the former is very detectable by NSM using TalExpCnt, as described previously. To remain stealthy, the attacker is must instead rely on an overflow.

SV

Our experiments show that none of the attacks on the SV protocol can be reliably detected using NSM DOs.

5.6.2 Attacks Not Detected

GOOSE

G9 involves delaying a PDU sufficiently to cause it to be dropped due to the skew filtering (DoS condition #3). We run this attack with different amounts of delay to see how this affects detection by NSM. We introduce a constant amount of delay between every pair of PDUs. We select 10 milliseconds and 1 millisecond as the amounts. We stop the attack once the attacker has accumulated a total of 4 seconds of delay. From our experiments, we find that the attacker can avoid detection by NSM. This is because RxPduPerSecond is the only NSM DO that is affected by changes in traffic rates. In our tests, when considering an attacker introducing delays of 10 milliseconds, the impact on RxPduPerSecond causes a large enough deviation from normal that the NSM manager is able to detect the attack accurately. However, if we reduce the delays to only 1 millisecond, the change in RxPduPerSecond is too small to be detected by NSM's anomaly detection even after running several trials. With this method, the attacker is able to introduce 1 second of total delay for every 1000 PDUs and without alerting NSM. Even if NSM could detect this particular variant, we could adapt the attack by introducing small delays in a randomized pattern, making it more difficult to detect. This attack necessarily takes time to carry out: in our tests, it takes about 15 minutes of constant execution to introduce a total of 4 seconds of delay. Once it is complete, it is very effective. We consider this a major threat to the substation because there are currently no measures in IEC 62351-6 or IEC 62351-7 that can address this reliably.

Among attacks that could not be detected by NSM, we have G16. This attack only requires injecting a single PDU, because even one PDU with a $test$ flag set to True tricks a misconfigured

IED into accepting only PDUs that also have this flag set. All subsequent legitimate traffic is then dropped. The injection does not cause a major change in RxPduPerSecond, allowing it to dodge detection by NSM. For the same reason, injecting a single PDU with a higher $stNum$ (G4) is also not detected by NSM. This is a much worse attack as its targeting $stNum$ does not rely on a configuration error like targeting $test$ does.

Modification attacks G3, G6, G8 and G18 target the $stNum$, t and $test$ fields, which are not monitored by NSM. Because they have no obvious effect on traffic rates, we are not able to detect them using NSM.

SV

Our experiments show that no attacks on SV are detected reliably by NSM. The only applicable NSM DO at our disposal is RxPduPerSecond, the rate of PDUs sent or received per second. Because this is the SV protocol, we expect this value to be constant. It should remain at 4,800 at all times in our case, regardless of the activity in the power model. In practice, the value of this NSM DO oscillates slightly between 4,804.8 and 4,859.4. This might be because the rate is calculated by a proxy, rather than the monitored device, and because of network delays. This NSM DO by itself should theoretically assist in detecting injection, replay, drop and delay attacks. However, because the attacks we design only alter the traffic slightly, it is not enough to enable reliable detection. Modification attacks S1, S2, S7 and S10 are not detected correctly by NSM because they do not affect traffic rate. Unlike for GOOSE PDUs, there are no NSM DOs that tracks invalid length, because PDUSizeFail explicitly only applies to SV over IP.

Using LSTM models, we are able to detect delay attack S4 only in certain conditions. The delay attack on SV is executed slightly differently than how it is done for GOOSE. Rather than delay all PDUs, delay is only introduced periodically, specifically after some amount of PDUs have been forwarded normally (e.g. every 10 PDUs). Changing the amount of PDUs that are forwarded normally between delays affects how quickly total delay can be accumulated, as well as detection by NSM. We find that by introducing delay every 10 PDUs (meaning about 500 times per second), NSM can detect the attack. However, if we instead introduce a delay only every 100 or 1,000 PDUs, this is no longer the case as it does not significantly affect RxPduPerSecond. In our testbed, this

attack causes the measurements to arrive too late to the subscriber, as one would expect.

Attacks S8 and S9 involve adding a PDU to the network with a higher $smpCnt$ at unsuitable times to trick the subscriber into dropping PDUs with lower $smpCnt$ in between clock ticks. These attacks must be repeated every second (after every clock tick) to have their intended effect. Since this involves adding one SV PDU per second to the already existing 4,800, this causes no detectable changes to the traffic rate.

5.6.3 Discussion

The experiments show that NSM DOs by themselves can be used to detect several attacks on GOOSE. Some of them are detected using efficient rule-based detection, while others rely on more sophisticated machine learning algorithms. Attacks on SV prove much more challenging to detect with our NSM implementation. We demonstrated how some of the attacks on GOOSE and SV that can be detected by NSM, such as delaying PDUs, can be tweaked to evade detection. This might indicate the need for either additional NSM DOs or alternative solutions to detect such attacks. In all cases, as we expected initially, pure modification attacks are very difficult to detect using NSM if they target some property of the traffic that is not monitored explicitly by NSM.

There are several attacks that are not possible in our testbed as is. We rely on an existing driver to support IEC 61850 protocols in our power system simulation. It currently does not provide support for some features needed to enable certain attacks, and it does not support MMS. Notably, it does not include the security features provided by IEC 62351-6. This is not unique to this specific driver [70], making it difficult to find a suitable replacement. A potential future work to improve the testbed is to modify or rewrite the driver so it includes the specifications of IEC 62351-6. Additionally, it should be noted that our results depend on a few assumptions on the meaning of specific NSM DOs, most notably TxPduPerSecond and RxPduPerSecond. This is discussed in more detail in Section 5.7. To the best of our capabilities, we implement the NSM DOs in a manner that reflects their description in IEC 62351-7. By doing so, we simultaneously provide an example of how equipment manufacturers might interpret the definitions of the NSM DOs. Thus, the experiments documented in this chapter provide insight into the design of NSM and its effectiveness against cyberattacks targeting IEC 61850 protocols.

5.7 Recommendations for Network and System Management

Overall, the NSM DOs of IEC 62351-7 are helpful in addressing many kinds of attacks. There are still a few attacks that cannot yet be addressed by NSM. This section provides an assessment and gap analysis of NSM, based on the discussion of previous sections, and recommendations to address them when applicable.

We have previously stated in Section 2.4 that we cannot modify the type of existing MIB objects without causing conflicts in OIDs. Thus, recommendations involving changes to the existing NSM DOs are more complicated to put into practice, due to the need to deprecate previous NSM DOs defined by IEC 62351-7. Our recommendations are centered on adding new NSM DOs and on improving the descriptions of existing ones, which avoids the need for any such changes.

5.7.1 Addition of Select NSM Data Objects

Based on the experiments done using our NSM implementation, the NSM DOs do not detect all possible DoS attacks we designed specifically for the substation. It is debatable whether the best solution is to design additional NSM DOs that could capture information about these events, or if it is best to rely on some other security countermeasure such as Deep Packet Inspection (DPI) or ID. This section describe suggestions of NSM DOs that could be added to an implementation of NSM and what benefits they would bring.

Count Accepted and Discarded GOOSE/SV PDUs

In the current edition of IEC 62351-7, there is not much information available from NSM DOs when it comes to monitoring the amount of GOOSE or SV traffic. The closest NSM DOs to match this requirement are the ones monitoring the rates of PDUs per second. This does not give much input on the overall amount of PDUs being exchanged, or whether the PDUs are being accepted at the subscriber. To determine whether a DoS attack might be underway against GOOSE or SV, it is important to find out the amount of PDUs that are structurally valid but that are rejected by the subscriber. This would help detect many of the DoS attacks listed in Table 5.5, even if it would not give an indication as to why the PDUs are rejected. If we look at the MMS PDUs of IEC 62351-7

as examples, these can track the total number of MMS PDUs exchanged as well as the number of RejectPDUs sent and received. In the same vein, though the subscriber cannot send PDUs to indicate rejection in GOOSE or SV, the subscriber can provide indications using these potential NSM DOs:

(1) Count for PDUs sent (publisher) and received (subscriber) on an association;

(2) Count for PDUs received on an association that were valid, but discarded for some other reason.

With these values, the NSM manager can determine the amount of PDUs exchanged between publisher and subscriber and notice if the subscriber is ever ahead of the publisher (potentially indicating injection or replay attacks). It can also determine the number of PDUs that are discarded, helping to locate DoS attacks, and determine the number of accepted PDUs by combining the information from both NSM DOs. Note that in the case of the SV protocol, which has very high amounts of traffic, counters are recommended to be made larger than usual (64-bit rather than 32-bit) because they can easily overflow otherwise.

Synchronization in GOOSE Using $stNum$ and t Fields

The NSM DOs for GOOSE could benefit from some additional NSM DOs to detect attacks that target vulnerable fields such as the $stNum$ and t fields. In general, the NSM DOs of IEC 62351-7 are not explicitly designed to include cases where the digital signatures of IEC 62351-6 are not used or where the keys for them are stolen by an attacker. Additionally, there are attacks, such as delay attacks, that target these fields and remain possible despite the signature and the security measures of the other parts of IEC 62351.

The fundamental issue enabling these attacks is that the GOOSE publisher and subscriber have to remain synchronized. The most important values for this purpose are the $stNum$ and t values. If these drift from each other, attacks shown previously can cause DoS, with neither side being able to detect the issue. If we consider the presence of the NSM manager, however, we can detect it by comparing the last $stNum$ and t values at the publisher and subscriber to see if they are consistent. Some example NSM DOs that would assist in this task are:

(1) Current $stNum$;

(2) t corresponding to the current $stNum$;

(3) Time of the last $stNum$ change (note: not the same as t. This one should be based on the device's local clock);

(4) Counter of $stNum$ resets;

(5) Time of the last $stNum$ reset.

By comparing the $stNum$ and time of the last $stNum$ change at publisher and subscriber, the NSM manager could detect when the subscriber is ahead of the publisher: something that should not occur in normal conditions. By examining the value of t, the manager can detect attempted replay attacks which would cause t to decrease when compared with the time of the last $stNum$ change. It can also assist in detecting delay attacks as these can cause the values of t at the publisher and subscriber to differ. However, this is not foolproof, as t remains constant if there are no state changes. A counter of the $stNum$ resets is needed to determine how often the $stNum$ goes back to 0, an event that enables some of the DoS attacks of Table 5.5. This role is partially fulfilled by the existing NSM DO to count TAL expirations, as these result in an $stNum$ reset according to IEC 62351-6 guidelines [28]. It does not account for $stNum$ overflows, which are possible though very rare [42].

It is possible to use DPI instead to monitor these vulnerable fields and detect attacks. Using DPI, the detection can happen faster as it is done in real-time, whereas the NSM manager must query for information periodically instead. The main advantage to having these values as NSM DOs are:

- It requests the information on the current $stNum$ and t directly from the hosts. The core of this synchronization problem is that only the hosts know the $stNum$ and t they are currently at, hence why requesting their information using SNMP is effective. Inspecting PDUs by itself does not guarantee that the host is actually accepting them;
- It is a low-cost solution, since all it requires is adding a few columns to tables in GSE

NSM DOs. The IED running an NSM agent must already update the SNMP tables relating to GSE and it has access to the current values of $stNum$ and t, so this is not a big load for the agent in theory;

- It does not require additional components in the network, or the ability to decrypt PDUs like DPI would, in cases where encryption is used with GOOSE.

The fact that the detection is not in real-time is indeed a disadvantage for NSM DOs. However, this kind of DoS attack usually does not have an immediate impact. For example, when forcing the GOOSE subscriber to use higher $stNum$ values, this does not necessarily trigger an event right away. Instead, it is future legitimate PDUs with $stNum$ changes that are affected, whenever they arrive. This means that there is often some time between the moment the attack is detectable and when it has an actual impact.

GOOSE $test$ Field

The NSM DOs for GSE monitor for values that could indicate communication errors in a GOOSE association, such as $confRev$ or $ndsCom$. Another value that can greatly affect the communication is the $test$ flag, which is not included in NSM DOs. The $test$ flag can be used in DoS attacks, as shown in Table 5.5, but only in cases where the configuration of an IED accepts test data. In general, these attacks are not the most threatening to a substation, since they rely on a configuration error to be effective. Given that other fields in GOOSE are already monitored by NSM, it would be a low-cost solution to include the $test$ flag as well to allow detecting such a configuration error.

Synchronization in SV

There are several DoS attacks that are difficult to detect in the case of SV. SV communications involve thousands of PDUs being transferred per second. The sample count $smpCnt$ is reset very often as a result. Using NSM DOs to track specific fields is therefore not reliable, given how often the fields change. A counter for discarded PDUs, as discussed earlier in this section, could at least indicate when measurements are getting lost. SV messages do carry a $timestamp$ field in IEC

62351-6, which can be helpful in addressing delay and replay attacks. The $timestamp$ is checked against the skew period to determine whether a PDU should be dropped [28]. While NSM DOs cannot track the $timestamp$ in each packet, a summary of all the $timestamp$ received could provide valuable information. The NSM DO RptReceptionDelay provides this kind of information for MMS [8]. Some examples of NSM DOs would be:

(1) Average "skew", meaning the absolute value of the difference between the $timestamp$ field and the current time, for many PDUs within some time period;

(2) Maximum skew among many PDUs within the same time period;

(3) Minimum skew among many PDUs within the same time period;

(4) Count of PDUs with a skew exceeding some threshold (that is much smaller than the skew period).

The IEC 62351-6 standard already expects the SV subscriber to calculate the difference between the $timestamp$ field with the current time for each PDU, as part of skew filtering [28]. To populate the suggested NSM DOs, the subscriber needs to record the results of this calculation. It is then possible to calculate these various statistics about the $timestamp$ field. Attempts at delay attacks are likely to affect the average and maximum values of the skew when compared to normal conditions. Replay attacks cause the maximum skew to increase noticeably, as the $timestamp$ of a replayed PDU is older than the $timestamp$ for the current PDUs.

Enhancements to GOOSE and SV

Though this work focuses on part 7 of IEC 62351, some changes in the IEC 61850 protocols could also prove useful to address some of the attacks discussed previously. Specifically, we find that some of the attacks on GOOSE and SV could motivate changes to the protocols. Our suggested enhancements would be used alongside the security extensions of IEC 62351-6.

Timestamps in GOOSE PDUs in GOOSE do not have any information to represent the time at which the PDU is created, as shown previously in the list of fields found in Table 2.3. Given a trace

of GOOSE communications, one cannot reliably determine the time frame during which the PDUs were originally sent. The t field does carry a timestamp, but it represents the moment when the $stNum$ last changed. It acts as a proper timestamp *only for the first PDU after a state change*, i.e. the one with a $sqNum$ of 0. Subsequent retransmission PDUs with higher $sqNum$ carry no such information, as they do not update the t field.

Given the above, we can only rely on the t field for detection purposes *after* a state change has occurred, when it is already too late to prevent or mitigate the attack. Indeed, many of the attacks we studied previously have their full effect upon the next state change. For instance, an attacker can stealthily introduce a delay of one minute between GOOSE publisher and subscriber (similar to attack G9 in Table 5.5) while in steady state and avoid detection by NSM or DPI. The attack has no impact until the publisher needs to send a critical message, such as a trip signal, to the subscriber. The latter will react far too slowly to protect the primary equipment as it should. In this scenario, we are unable to detect the actions of the attacker by inspecting the PDUs: we only notice the change in t once the critical message is sent. Including a timestamp field in GOOSE PDUs can mitigate this problem, as its value can be inspected at all times to detect this kind of attack before it has a chance to cause damage. Hence, we recommend adding timestamp information to GOOSE PDUs, much like how IEC 62351-6 already adds a $timestamp$ field to the SV PDUs.

Changes to SV security When it comes to the security of SV, we suggest the following to help in preventing stealthy delay and replay attacks:

(1) Allowing adjusting the skew period, currently set to 2 minutes;

(2) Discard PDUs with a $timestamp$ that is older than the last one received.

The latter suggestion is the same mechanism that is already specified by IEC 62351-6 for the $smpCnt$ field. The main difference is that the $smpCnt$ field resets to 0 often, while the $timestamp$ field should only ever increase for the stream of PDUs coming from one publisher. This theoretically blocks replay attacks. When combined with the NSM DOs suggested in this section, attacks that try to abuse this new mechanism cause changes in NSM DOs that would be tracking the $timestamp$ statistics.

5.7.2 Limitations of NSM Solution

Though NSM is meant to enhance cybersecurity, its potential use as an attack vector or target itself must be considered. The addition of any component to the substation network necessarily increases its complexity and brings with it new potential vulnerabilities. This section briefly discusses some of the possible limitations of NSM. They are included for informative purposes. Solutions to address these issues is not part of the scope of this work.

Device Incompatibility

Equipment manufactured before the publication of IEC 62351-7:2017 does not support the standard's NSM DOs (and it also might not support SNMP), leading to compatibility problems when attempting to connect them to an NSM system using SNMP. Solving this necessitates the implementation of a proxy that can convert the information from the device into the format of NSM DOs and serve it as SNMP MIBs. The proxy can be placed at the NSM manager, as done in the experiments documented in the report by EPRI [82]. It can also be found closer to the monitored devices, as we do in our own experiments. Either way, the proxy requires time and resources to develop. Additionally, because devices from different manufacturers do not work identically, the proxies might have to be customized to be compatible with the specific hardware they are supposed to monitor. This issue also affects network devices, despite the fact that they typically support SNMP, because the MIBs they serve can be proprietary and require mapping [82]. Further complicating matters, many IEDs do not expose the required information to populate all the NSM DOs, resulting in a proxy that supports only a portion of the NSM DOs [82].

In the ideal case, devices sold on the market would support IEC 62351-7 natively by shipping with a complete NSM agent implementation, eliminating this problem entirely. However, the IEC 62351-7 standard does not address the question of how to manage devices that are already deployed. Solutions to add support for IEC 62351-7 include firmware updates or the development of proxies.

Compromise of NSM Agent

A fundamental assumption when monitoring with SNMP is that the hosts being queried report accurate information. The NMS obtains data by sending SNMP requests over the network to each host and saving the data found in the resulting SNMP replies. Further analysis is conducted using this collected data. Changing the information that is reported to the NMS can therefore undermine the entire process, as the data is no longer reliable nor suitable for analytics.

One way to alter the SNMP traffic is by compromising the network carrying the requests. By doing so, the attacker can target the SNMP replies heading to the NMS. SNMPv3 addresses this by adding cryptography to provide confidentiality and integrity to SNMP messages. As a result, the network devices and links cannot modify the MIB information received at the NMS. This does become possible if SNMPv2 is used.

A more concerning attack is a compromise of the NSM agents themselves. The hosts responding to the SNMP requests are responsible for producing the messages and know all the credentials needed to do this. A host that is compromised leaks its keys and can be forced to serve false information to the NMS without the latter being able to detect it. This is a fundamental issue in SNMP and means that NSM is not really able to deal with a host compromise. Some measures can be taken to mitigate this problem somewhat. The NSM system can still monitor all other surrounded hosts that are not compromised. It is possible that they have information that can indicate unusual behavior coming from the malicious host. Additionally, because the attacker has to execute an attack in order to compromise the host in the first place, this action in itself can leave traces in the MIBs. Leveraging other technologies can also help address this. Since the hosts are no longer trustworthy, we can instead monitor their network communications using an IDS, which could reveal when a host is performing actions that it is not reporting in its MIBs. This makes it more difficult for the attacker to use their capabilities on the compromised host. Again, this is assuming that the IDS is also trustworthy.

Denial-of-Service Attack on NSM Manager

An attacker wishing to hide their tracks could consider reducing the visibility that the NSM manager has on the network. Much like any of the hosts it is monitoring, the NSM manager is vulnerable to DoS attacks. By flooding the manager with junk traffic or even junk MIB data in the form of traps, the users of the NSM manager might not be able to distinguish the legitimate traffic from the noise. Dropping or delaying SNMP replies destined for the NSM manager can also deprive it of useful information. Neither of these attacks is stealthy, as the NSM manager can quickly detect that it is receiving either too much traffic or that specific hosts are unresponsive. With its capabilities, the NSM manager is able to pinpoint at what location in the network the communications seem to be dropped. However, it still loses all the information it would normally have received during such an attack, making it difficult to track the attacker's activities during that time.

If the NSM manager itself is ever sabotaged or compromised by an attacker, the entire NSM system collapses as the manager is its central pillar. This is why it is useful to have a backup NSM manager that hopefully is not compromised itself, and also why the NSM manager should be located in its own security perimeter where firewalls can limit access to it as much as possible. Good log management can also assist for auditing purposes in the aftermath of a successful attack.

SNMP Vulnerabilities

The use of SNMP brings with it any vulnerability that affects the protocol, such as those found by Lawrence *et al.* [39]. Notably, SNMP can be used in DRDoS amplification attacks [38] that can potentially paralyze the entire network, not just the NSM system. The requirement to conduct this attack is to find out the credentials accepted by an SNMP agent so that it responds to GetBulk queries, and then send an excessive amount of GetBulks to flood the network. When using SNMPv2, which provides virtually no cybersecurity, the attacker can simply listen to traffic to locate the credentials (the community strings). IEC 62351-7 explicitly makes SNMPv3 use mandatory [8] and states that its setting noAuthNoPriv (provides no security) is deprecated. Thus, a conforming NSM system should not allow for such a trivial sniffing attack. This should prevent easy-to-execute DRDoS attacks, as an attacker is forced to first compromise an agent's SNMP credentials. However,

if this ever occurs, then the compromised host can be used as a reflector. This is made even worse if the hosts all share the same password, which is why it is recommended to have a unique key per SNMP agent. In general, DRDoS attacks should not be easy to carry out if NSM is implemented as per IEC 62351-7, but a manufacturer might mistakenly include SNMPv2 or low-security settings, thinking they can be used as a substitute.

Chapter 6

Conclusion

Digital substations are an essential part of the smart grid. Thus, they are also potential targets for cyberattacks aiming to disturb power grid operations. As part of the cybersecurity standard IEC 62351, IEC 62351-7 specifies the use of NSM to perform security monitoring in the context of the smart grid. NSM allows utilities to have visibility over the communication infrastructure that supports power systems in order to detect and mitigate issues before they cause further damage [8]. NSM complements the remaining parts of IEC 62351 by assisting with detection of cyberattacks, notably attacks on end devices and DoS attacks [8]. Given that IEC 62351-7 was published in 2017, it has not yet been the subject of much research.

The objective of this thesis is to study NSM, as per the IEC 62351-7 standard, and its capabilities in addressing cyberattacks specific to digital substations conforming to IEC 61850. In order to fulfill this objective, we make the following contributions:

- We design and implement a co-simulation smart grid security testbed that integrates NSM according to the IEC 62351-7 standard. This testbed combines the real-time power system simulator HYPERSIM, real hardware and NSM components by connecting them over OpenStack. It enables us to run many cyberattacks against a substation and test the capabilities of NSM.

- We design a methodology to elaborate cyberattacks on IEC 61850 substations, most notably

DoS attacks on GOOSE and SV protocols, as they are defined in IEC standards. This methodology can be adapted to different implementations of GOOSE and SV, as it allows for flexibility in defining the behavior of the protocol as DoS conditions. By applying this methodology, we are able to derive many attacks, many of which are not previously documented, while others are variants of already known attacks.

- We elaborate detection algorithms using the NSM DOs of IEC 62351-7 to detect and mitigate the attacks on IEC 61850 designed using our proposed methodology. We validate these algorithms in experiments using our testbed and evaluate the effectiveness of NSM in detecting attacks. In summary, there are several algorithms that can be used to reliably detect attacks specific to GOOSE, while attacks specific to the SV protocol prove more difficult to detect.

- We provide a security assessment and recommendations to enhance NSM in order to better address the cyberattacks that we found are not currently detectable. Most of the recommendations involve the addition of new NSM DOs, meaning that they can be seamlessly integrated with the current version of IEC 62351-7 without breaking compatibility. We also suggest potential enhancements to the GOOSE and SV protocols to help mitigate specific attacks that NSM cannot fully address.

In short, we find that NSM is able to address many kinds of attacks against IEC 61850 substations. While some attacks remain undetected according to our experiments, it is possible to enhance NSM by adding new NSM DOs.

6.1 Limitations and Future Work

Future work on NSM as per IEC 62351-7 can focus on many topics. The first is studying NSM with the MMS protocol in more detail, as we mostly limited ourselves to GOOSE and SV. This would require finding an implementation of MMS that is compatible with our co-simulation testbed. The second is to test cyberattacks on an implementation of IEC 61850 that fully conforms to the specifications of IEC 61850 and IEC 62351-6. In turn, this would enable studying in detail the physical impact of the cyberattacks designed in this thesis. As for the third topic, we limited

the number of NSM DOs we could test due to existing IEDs not supporting IEC 62351-7, requiring additional implementation of proxies (one per device model) to integrate the NSM DOs from the standard. The capabilities of NSM could be tested more thoroughly with a complete implementation of the NSM DOs from IEC 62351-7.

Bibliography

[1] F. Cleveland, "IEC 62351 security standards for the power system information infrastructure", IEC, 2012.

[2] (2017). NERC, [Online]. Available: https://www.nerc.com/Pages/default.aspx.

[3] S. W. Blume, *Electric Power Systems Basics for the Nonelectrical Professional*. Piscataway: IEEE Press, 2007.

[4] IEC 61850-1, "Communication networks and systems for power utility automation - part 1: Introduction and overview", 2013.

[5] A. R. Metke and R. L. Ekl, "Smart grid security technology", in *2010 Innovative Smart Grid Technologies (ISGT)*, 2010, pp. 1–7. DOI: 10.1109/ISGT.2010.5434760.

[6] T. Jansen, "Technical considerations for building secure substation automation systems", Electra, D2/B3/C2.01 Joint Working Group Report, Nov. 29, 2006.

[7] IEC 62351-1, "Communication network and system security - introduction to security issues", 2007.

[8] IEC/TS 62351-7, "Power systems management and associated information exchange - data and communications security - part 7: Network and system management (NSM) data object models", 2017.

[9] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. 1996.

[10] (Apr. 15, 2014). Fingerprint lock in Samsung Galaxy 5 easily defeated by whitehat hackers, [Online]. Available: https://arstechnica.com/information-technology/2014/04/fingerprint-lock-in-samsung-galaxy-5-easily-defeated-by-whitehat-hackers/.

[11] E. Barker and Q. Dang, "NIST special publication 800-57 part 3 revision 1", NIST.

[12] (Mar. 2, 2015). Denial of service, [Online]. Available: https://www.owasp.org/index.php/Denial_of_Service.

[13] (Nov. 28, 2016). DDoS attack types: The 12 types of DDoS attacks used by hackers, [Online]. Available: https://www.rivalhost.com/12-types-of-ddos-attacks-used-by-hackers.

[14] D. Ince, *Degradation of service attack*. [Online]. Available: http://www.oxfordreference.com/view/10.1093/acref/9780199571444.001.0001/acref-9780199571444-e-858.

[15] S. E. Collier, "Ten steps to a smarter grid", in *2009 IEEE Rural Electric Power Conference*, 2009, B2–B2–7. DOI: 10.1109/REPCON.2009.4919420.

[16] (Jun. 13, 2017). CrashOverride - analysis of the threat to electric grid operations, [Online]. Available: https://www.dragos.com/blog/crashoverride/CrashOverride-01.pdf.

[17] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier", *White paper, Symantec Corp., Security Response*, vol. 5, no. 6, 2011.

[18] (Feb. 25, 2016). Cyber-attack against Ukrainian critical infrastructure, [Online]. Available: https://ics-cert.us-cert.gov/alerts/IR-ALERT-H-16-056-01.

[19] (Jul. 12, 2017). CrashOverride malware, [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA17-163A.

[20] IEC 61850-5, "Communication networks and systems for power utility automation - part 5: Communication requirements for functions and device models", 2013.

[21] Z. Zhang, X. Huang, *et al.*, "Modeling and simulation of data flow for VLAN-based communication in substations", *IEEE Systems Journal*, vol. PP, pp. 1–12, 99 2015.

[22] IEC/TR 61850-90-4, "Communication networks and systems for power utility automation - part 90-4: Network engineering guidelines", 2013.

[23] IEC 61850-7-1, "Communication networks and systems for power utility automation - part 7-1: Basic communication structure - principles and models", 2011.

[24] IEC 61850-7-2, "Communication networks and systems for power utility automation - part 7-2: Basic information and communication structure - abstract communication service interface (ACSI)", 2010.

[25] IEC 61850-7-4, "Communication networks and systems for power utility automation - part 7-4: Basic communication structure - compatible logical node classes and data object classes", 2010.

[26] IEC 61850-9-2, "Communication networks and systems for power utility automation - part 9-2: Specific communication service mapping (SCSM) - sampled values over ISO/IEC 8802-3", 2011.

[27] IEC/TS 61850-8-1, "Communication networks and systems for power utility automation - part 8-1: Specific communication service mapping (SCSM) - mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3", 2011.

[28] IEC/TS 62351-6, "Power systems management and associated information exchange - data and communications security - part 6: Security for IEC 61850", 2007.

[29] (Nov. 18, 2016). What does time synchronisation mean for Sampled Values?, [Online]. Available: https://ideology.atlassian.net/wiki/spaces/AP/pages/50069508/What+does+time+synchronisation+mean+for+Sampled+Values.

[30] ISO, "ISO 9506-1:2003", ISO, Aug. 2003.

[31] SISCO, "Overview and introduction to the Manufacturing Message Specification (MMS - revision 2)", SISCO, 1995.

[32] J. T. Sørensen and M. G. Jaatun, "An analysis of the Manufacturing Messaging Specification protocol", in *Ubiquitous Intelligence and Computing*, F. E. Sandnes, Y. Zhang, *et al.*, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 602–615, ISBN: 978-3-540-69293-5.

[33] R. L. O'Fallon, D. A. Klas, *et al.*, "IEC 61850 MMS SCADA network optimization for IEDs", in *DistribuTECH Conference, February 2011*, Feb. 2011.

[34] D. R. Mauro and K. J. Schmidt, *Essential SNMP, Second Edition*. O'Reilly Media, Inc., 2005, ISBN: 0596008406.

[35] W. Stallings, "SNMP and SNMPv2: The infrastructure for network management", *IEEE Communications Magazine*, vol. 36, no. 3, pp. 37–43, 1998, ISSN: 0163-6804. DOI: 10.1109/35.663326.

[36] (2018). Object identifier (OID) repository, [Online]. Available: http://oid-info.com/index.htm.

[37] M. Stump, *Securing SNMP: A look at Net-SNMP (SNMPv3)*, 2003.

[38] (Aug. 2012). SNMP reflected amplification DDoS attack mitigation, [Online]. Available: https://www.bitag.org/documents/SNMP-Reflected-Amplification-DDoS-Attack-Mitigation.pdf.

[39] N. Lawrence and P. Traynor, "Under new management: Practical attacks on SNMPv3", in *Presented as part of the 6th USENIX Workshop on Offensive Technologies*, Bellevue, WA: USENIX, 2012. [Online]. Available: https://www.usenix.org/conference/woot12/workshop-program/presentation/Lawrence.

[40] IEC 62351-3, "Power systems management and associated information exchange - data and communications security - part 3: Communication network and system security - profiles including TCP/IP", 2014.

[41] IEC/TS 62351-4, "Power systems management and associated information exchange - data and communications security - part 4: Profiles including MMS", 2007.

[42] M. Strobel, N. Wiedermann, and C. Eckert, “Novel weaknesses in IEC 62351 protected smart grid control systems”, in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 266–270.

[43] S. Fuloria, R. Anderson, *et al.*, “The protection of substation communications”, Cambridge University, 2010.

[44] F. Hohlbaum, M. Braendle, and F. Alvarez, “Cyber security: Practical considerations for implementing IEC 62351”, ABB, 2009.

[45] IEC/TS 62351-7, “Power systems management and associated information exchange - data and communications security - part 7: Network and system management (NSM) data object models”, 2010.

[46] (2017). IEC TS 62351-7:2010 — IEC Webstore — cyber security, smart city, [Online]. Available: https://webstore.iec.ch/publication/6910.

[47] (2017). IEC TS 62351-7:2017 — IEC Webstore — cyber security, smart city, [Online]. Available: https://webstore.iec.ch/publication/30593.

[48] E. R. Raghunarayan, “Management information base for the Transmission Control Protocol (TCP)”, The Internet Society, Mar. 2005.

[49] M. Mathis and J. Heffner, “TCP extended statistics MIB”, The IETF Trust, May 2007.

[50] W. Hardaker, “Transport Layer Security (TLS) transport model for the Simple Network Management Protocol (SNMP)”, IETF, Aug. 2010.

[51] B. Fenner and J. Flick, “Management information base for the User Datagram Protocol (UDP)”, The Internet Society, Jun. 2005.

[52] B. Haberman, “IP forwarding table MIB”, The Internet Society, Apr. 2006.

[53] E. S. Routhier, “Management information base for the Internet Protocol (IP)”, The Internet Society, Apr. 2006.

[54] D. McWalter, D. Thaler, and A. Kessler, “IP multicast MIB”, The IETF Trust, Dec. 2007.

[55] NESCOR, “Electric sector failure scenarios and impact analyses - version 3.0”, EPRI, 2015.

[56] ——, “Analysis of selected electric sector high risk failure scenarios”, EPRI, 2015.

[57] M. T. A. Rashid, S. Yussof, *et al.*, “A review of security attacks on IEC61850 substation automation system network”, in *2014 International Conference on Information Technology and Multimedia (ICIMU)*, 2014.

[58] J. Wright and S. Wolthusen, “Access control and availability vulnerabilities in the ISO/IEC 61850 substation automation protocol”, in *Critical Information Infrastructures Security*. Nov. 2017, p. 239.

[59] B. Kang, P. Maynard, *et al.*, “Investigating cyber-physical attacks against IEC 61850 photovoltaic inverter installations”, in *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, 2015, pp. 1–8.

[60] A. Valdes, C. Hang, *et al.*, “Design and simulation of fast substation protection in IEC 61850 environments”, in *2015 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, 2015, pp. 1–6.

[61] J. Hoyos, M. Dehus, and T. X. Brown, “Exploiting the GOOSE protocol: A practical attack on cyber-infrastructure”, in *2012 IEEE Globecom Workshops*, 2012, pp. 1508–1513.

[62] N. Kush, E. Ahmed, *et al.*, “Poisoned GOOSE: Exploiting the GOOSE protocol”, in *Proceedings of the Twelfth Australasian Information Security Conference (AISC 2014), Auckland, New Zealand*, 2014.

[63] O. Khaled, A. Marin, *et al.*, “Analysis of secure TCP/IP profile in 61850 based substation automation system for smart grids”, *Int. J. Distrib. Sen. Netw.*, vol. 2016, Apr. 2016, ISSN: 1550-1329. DOI: 10.1155/2016/5793183.

[64] J. G. Wright and S. D. Wolthusen, “Limitations of IEC62351-3’s public key management”, in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, 2016.

[65] M. M. R. Chowdhury, H. Raddatz, and J. E. Y. Rosseb, “Challenges when securing manufacturing message service in legacy industrial control systems”, in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, 2014, pp. 1–6. DOI: 10.1109/ETFA.2014.7005355.

[66] S. Fries, H. J. Hof, and M. Seewald, "Enhancing IEC 62351 to improve security for energy automation in smart grid environments", in *2010 Fifth International Conference on Internet and Web Applications and Services (ICIW)*, 2010, pp. 135–142.

[67] K. C. Ruland and J. Sassmannshausen, "Non-repudiation services for the MMS protocol of IEC 61850", in *Security Standardisation Research: Second International Conference, SSR 2015, Tokyo, Japan, December 15-16, 2015, Proceedings*, L. Chen and S. Matsuo, Eds. Cham: Springer International Publishing, 2015, pp. 70–85.

[68] P. Weerathunga, "Security aspects of smart grid communication", Western University, 2012.

[69] R. Schlegel, S. Obermeier, and J. Schneider, "Assessing the security of IEC 62351", in *Proceedings of the 3rd International Symposium for ICS & SCADA Cyber Security Research 2015*, 2015.

[70] M. El Hariri, T. A. Youssef, and O. A. Mohammed, "On the implementation of the IEC 61850 standard: Will different manufacturer devices behave similarly under identical conditions?", *Electronics*, vol. 5, no. 4, 2016.

[71] (May 9, 2018). Fuzzing, [Online]. Available: https://www.owasp.org/index.php/Fuzzing.

[72] (2017). State of fuzzing 2017, [Online]. Available: https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/state-of-fuzzing-2017.pdf.

[73] Y. Yang, H. T. Jiang, *et al.*, "Cybersecurity test-bed for IEC 61850 based smart substations", in *2015 IEEE Power Energy Society General Meeting*, 2015, pp. 1–5. DOI: 10.1109/PESGM.2015.7286357.

[74] R. Amoah, S. Suriadi, *et al.*, "Security analysis of the non-aggressive challenge response of the DNP3 protocol using a CPN model", in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 827–833.

[75] IEC/TS 62351-5, "Power systems management and associated information exchange - data and communications security - part 5: Security for IEC 60870-5 and derivatives", 2013.

[76] R. Amoah, S. Camtepe, and E. Foo, "Formal modelling and analysis of DNP3 secure authentication", *Journal of Network and Computer Applications*, 2016.

[77] C. J. F. Cremers, M. Dehnel-Wild, and K. Milner, "Secure authentication in the grid: A formal analysis of DNP3: SAv5", in *ESORICS*, 2017.

[78] A. Hall, "Realising the benefits of formal methods", in *Formal Methods and Software Engineering*, K.-K. Lau and R. Banach, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 1–4, ISBN: 978-3-540-32250-4.

[79] J. L. P. de Sa and R. Cartaxo, "Implementing substations automatic control functions designed with Petri Nets on IEC 61850", *IEEE Transactions on Power Delivery*, vol. 26, no. 2, pp. 1119–1127, 2011, ISSN: 0885-8977. DOI: 10.1109/TPWRD.2010.2090952.

[80] G. Kunz, J. Machado, *et al.*, "A formal methodology for accomplishing IEC 61850 real-time communication requirements", *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6582–6590, 2017, ISSN: 0278-0046. DOI: 10.1109/TIE.2017.2682042.

[81] L. Obregon, "Infrastructure security architecture for effective security monitoring", SANS Institute, Dec. 2, 2015.

[82] R. King, "Network system management: Implementations and applications of the IEC 62351-7 standard", EPRI, Technical Update.

[83] Y. Kwon, M.-S. Kim, *et al.*, "Network and system management object modeling for smart grid infrastructure", in *Network Operations and Management Symposium (APNOMS), 2015 17th Asia-Pacific*, 2015.

[84] C.-H. Kim, M.-S. Choi, *et al.*, "Security data extraction from IEC 61850 ACSI models for network and system management", in *WISA 2011 International Workshop on Information Security Applications*, 2011.

[85] S. Ju, Y. Lim, *et al.*, "Anomaly detection mechanism based on common NSM data objects for advanced metering infrastructure", *Advances in Computer Science : an International Journal*, 2014.

[86] K. Choi, X. Chen, *et al.*, "Intrusion detection of NSM based DoS attacks using data mining in smart grid", *Energies*, 2012.

[87] IEC/TR 61850-90-5, "Communication networks and systems for power utility automation - part 90-5: Use of IEC 61850 to transmit synchrophasor information according to IEEE C37.118", 2012.

[88] (2018). SteelCentral Riverbed Modeler, [Online]. Available: https://www.riverbed.com/products/steelcentral/steelcentral-riverbed-modeler.html.

[89] I. Ali, M. S. Thomas, *et al.*, "IEC 61850 substation communication network architecture for efficient energy system automation", *Energy Technology & Policy*, vol. 2, no. 1, pp. 82–91, 2015. DOI: 10.1080/23317000.2015.1043475.

[90] M. Golshani, G. A. Taylor, and I. Pisica, "Simulation of power system substation communications architecture based on IEC 61850 standard", in *2014 49th International Universities Power Engineering Conference (UPEC)*, 2014, pp. 1–6. DOI: 10.1109/UPEC.2014.6934745.

[91] (2018). OMNeT++ discrete event simulatior - home, [Online]. Available: https://www.omnetpp.org/.

[92] A. Hahn, A. Ashok, *et al.*, "Cyber-physical security testbeds: Architecture, application, and evaluation for smart grid", *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 847–855, 2013, ISSN: 1949-3053.

[93] (2018). PowerFactory, [Online]. Available: https://www.digsilent.de/en/powerfactory.html.

[94] (2018). REAL TIME POWER SYSTEM SIMULATION, [Online]. Available: https://www.rtds.com/real-time-power-system-simulation/.

[95] (2018). Simulator overview, [Online]. Available: https://www.powerworld.com/products/simulator/overview.

[96] T. B. Lo, M. F. Mendes, *et al.*, "Cloud IEC 61850: Architecture and integration of electrical automation systems", in *2014 Brazilian Symposium on Computing Systems Engineering*, 2014, pp. 13–18. DOI: 10.1109/SBESC.2014.30.

[97] R. Wjtowicz, R. Kowalik, and D. D. Rasolomampionona, "Next generation of power system protection automation - virtualization of protection systems", *IEEE Transactions on Power Delivery*, vol. 33, no. 4, pp. 2002–2010, 2018, ISSN: 0885-8977. DOI: 10.1109/TPWRD.2017.2786339.

[98] J. Yu, H. Lee, *et al.*, "Traffic flooding attack detection with SNMP MIB using SVM", *Computer Communications*, vol. 31, no. 17, pp. 4212 –4219, 2008, ISSN: 0140-3664. DOI: 10.1016/j.comcom.2008.09.018.

[99] J. Yu, H. Kang, *et al.*, "An in-depth analysis on traffic flooding attacks detection and system using data mining techniques", *Journal of Systems Architecture*, vol. 59, no. 10, Part B, pp. 1005 –1012, 2013, Advanced Smart Vehicular Communication System and Applications, ISSN: 1383-7621. DOI: 10.1016/j.sysarc.2013.08.008.

[100] P. M. Priya, V. Akilandeswari, *et al.*, "The Protocol Independent Detection and Classification (PIDC) system for DRDoS attack", in *2014 International Conference on Recent Trends in Information Technology*, 2014, pp. 1–7. DOI: 10.1109/ICRTIT.2014.6996154.

[101] W. Cerroni, G. Moro, *et al.*, "Decentralized detection of network attacks through P2P data clustering of SNMP data", *Computers & Security*, vol. 52, pp. 1 –16, 2015, ISSN: 0167-4048. DOI: 10.1016/j.cose.2015.03.006.

[102] K. McLaughlin, "Contract no 608224 deliverable D4.1 high-level design documentation and deployment architecture for multi-attribute SCADA intrusion detection system", 2015.

[103] Y. Yang, H. Q. Xu, *et al.*, "Multidimensional intrusion detection system for IEC 61850-based SCADA networks", *IEEE Transactions on Power Delivery*, vol. 32, no. 2, pp. 1068–1078, 2017, ISSN: 0885-8977. DOI: 10.1109/TPWRD.2016.2603339.

[104] U. K. Premaratne, J. Samarabandu, *et al.*, "An intrusion detection system for IEC61850 automated substations", *IEEE Transactions on Power Delivery*, 2010.

[105] J. Hong, C.-C. Liu, and M. Govindarasu, "Integrated anomaly detection for cyber security of the substations", *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1643–1653, 2014.

[106] H. Yoo and T. Shon, "Novel approach for detecting network anomalies for substation automation based on IEC 61850", *Multimedia Tools and Applications*, vol. 74, no. 1, pp. 303–318, 2015, ISSN: 1573-7721. DOI: 10.1007/s11042-014-1870-0.

[107] C. Feng, T. Li, and D. Chana, "Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks", in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2017, pp. 261–272. DOI: 10.1109/DSN.2017.34.

[108] N. Mohan, K. P. Soman, and R. Vinayakumar, "Deep power: Deep learning architectures for power quality disturbances classification", in *2017 International Conference on Technological Advancements in Power and Energy (TAP Energy)*, 2017, pp. 1–6. DOI: 10.1109/TAPENERGY.2017.8397249.

[109] (2018). Weka 3: Data mining software in Java, [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka/.

[110] (2018). Snort, [Online]. Available: https://www.snort.org/.

[111] (Aug. 8, 2018). TensorFlow, [Online]. Available: https://www.tensorflow.org/.

[112] (Jul. 23, 2018). Keras documentation, [Online]. Available: https://keras.io/.

[113] (2018). HYPERSIM, [Online]. Available: https://www.opal-rt.com/systems-hypersim/.

[114] (May 26, 2011). Net-SNMP tutorial – MIB module, [Online]. Available: http://net-snmp.sourceforge.net/tutorial/tutorial-5/toolkit/mib_module/index.html.

[115] (2018). Python, [Online]. Available: https://www.python.org/.

[116] (Dec. 15, 2017). GNU Bash, [Online]. Available: https://www.gnu.org/software/bash/.

[117] (2018). SQLite, [Online]. Available: https://www.sqlite.org/index.html.

[118] (2017). SNMP library for Python, [Online]. Available: http://pysnmp.sourceforge.net/.

[119] (2018). Elasticsearch, [Online]. Available: https://www.elastic.co/products/elasticsearch.

[120] (Jan. 15, 2004). SNMPTRAPD, [Online]. Available: http://net-snmp.sourceforge.net/docs/man/snmptrapd.html.

[121] (2018). Kibana, [Online]. Available: https://www.elastic.co/products/kibana.

[122] (Oct. 10, 2006). SNMP: Frequently asked questions about MIBs, [Online]. Available: https://www.cisco.com/c/en/us/support/docs/ip/simple-network-management-protocol-snmp/9226-mibs-9226.html#q20b.

[123] (2017). OWASP top 10 - 2017, [Online]. Available: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf.

[124] W. M. Eddy, "SYN flood attack", in *Encyclopedia of Cryptography and Security*, H. C. A. van Tilborg and S. Jajodia, Eds. Boston, MA: Springer US, 2011, pp. 1273–1274, ISBN: 978-1-4419-5906-5. DOI: 10.1007/978-1-4419-5906-5_276.

[125] R. Shirey, "RFC 2828: Internet security glossary", The Internet Society, May 2000.

[126] (Mar. 17, 2017). Vulnerability note VU#867968 - Microsoft Windows SMB tree connect response denial of service vulnerability, [Online]. Available: https://www.kb.cert.org/vuls/id/867968.

[127] *US v. Morris*, No. 774, Docket 90-1336. Court of Appeals, 2nd Circuit, 1991, vol. 928, p. 504.

[128] (May 4, 2017). BrickerBot results in PDoS attack, [Online]. Available: https://security.radware.com/ddos-threats-attacks/brickerbot-pdos-permanent-denial-of-service.

[129] A. Bohra, I. Neamtiu, *et al.*, "Remote repair of operating system state using backdoors", in *International Conference on Autonomic Computing, 2004. Proceedings.*, 2004, pp. 256–263. DOI: 10.1109/ICAC.2004.1301371.

[130] S. Hariri, G. Qu, *et al.*, "Impact analysis of faults and attacks in large-scale networks", *IEEE Security & Privacy*, vol. 1, no. 5, pp. 49–54, 2003, ISSN: 1540-7993. DOI: 10.1109/MSECP.2003.1236235.

[131] (Apr. 11, 2016). Decrypting the Petya ransomware, [Online]. Available: https://blog.checkpoint.com/2016/04/11/decrypting-the-petya-ransomware/.

[132] Infrastructure Security and Energy Restoration, Office of Electricity Delivery and Energy Reliability, and U.S. Department of Energy, "Large power transformers and the U.S. electric grid", U.S. Department of Energy, Apr. 2014.

[133] "IEEE standard for synchrophasor measurements for power systems", *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1–61, 2011. DOI: 10.1109/IEEESTD.2011.6111219.