POLITECNICO DI TORINO

MASTER's Degree in CYBERSECURITY

MASTER's Degree Thesis

Modeling and Analysis of Communication Protocols in Smart Energy Systems

Supervisors

Prof. Fulvio VALENZA

Prof. ssa Maria FERRARA

Candidate

Matteo FAILLA

MAY 2026

Abstract

Today's power systems are going through a process of decentralization and digitalization that is changing traditional smart grids into Smart Energy Systems. There are several types of communication protocols used for interaction between distributed generation, smart meters, building automation systems, energy storage and demand response events.

This process increases efficiency and flexibility and enables a better connection to renewable energy sources; however, it simultaneously makes the system more vulnerable than the classical architecture was. Many communication protocols developed earlier under the criteria of reliability and efficiency now need rigorous cybersecurity requirements.

This thesis will analyze the security and efficiency trade-offs of the main communication protocols used at different levels of a Smart Energy System. At the macro level, it will examine IEC 61850 and DNP3 in relation to substation automation and SCADA communication, relating them to their respective security extensions (IEC 62351 and DNP3 Secure Authentication). At the edge and prosumer level, DLMS/COSEM and OpenADR are analyzed for smart metering and demand response applications. At the micro level, it will focus on BACnet and BACnet/SC in the context of building energy management systems.

The second part of the thesis applies a hybrid threat modeling approach based on TAMELESS, extended with a Large Language Model which integrates MITRE ATT&CK methodologies. This approach, combining formal reasoning and contextual vulnerability identification, is used to evaluate unsecured, partially secured, and fully secured Smart Energy Systems scenarios.

From the findings, it is clear that communication security protocols are critical in ensuring that various attack vectors are blocked within Smart Energy System environments; however, this alone is insufficient. In addition, other techniques like defense in depth and host security measures must also be implemented. It can be concluded that the security of Smart Energy Systems requires context-specific approaches.

Acknowledgments

Al termine di questo percorso di studi, desidero ringraziare tutti coloro che mi hanno aiutato, direttamente e indirettamente, in questi ultimi anni accademici e in particolare durante la stesura della tesi.

Ringrazio innanzitutto i relatori di questa tesi, il Prof. Fulvio Valenza e la professoressa Maria Ferrara, per avermi guidato nella giusta direzione e supportato nel lavoro di tesi, dandomi l'opportunità di approfondire gli argomenti trattati.

Ringrazio poi tutti i miei colleghi e amici, che durante il mio percorso universitario mi hanno regalato molti momenti di svago e complicità e che mi hanno sostenuto durante la mia vita universitaria. Senza di voi non sarei mai riuscito a vivere questo percorso con la stessa serenità, motivazione e leggerezza.

Ringrazio infine la mia famiglia, che più di tutti mi ha incoraggiato e supportato senza mai giudicarmi o farmi mancare il proprio affetto, e ha sempre creduto in me a prescindere da ogni situazione. Siete stati sempre il mio punto di riferimento e non smetterete mai di esserlo.

"Progress is not achieved by luck or accident, but by working on yourself daily." - Epictetus (from Discourses)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and Objectives

The energy industry is currently experiencing significant changes due to the growing presence of renewable energy resources, the decentralization of energy production, and the digitization of energy infrastructure. Centralized energy grids, characterized by a unidirectional system where energy is produced by large energy plants and consumed by passive users, have become obsolete and cannot support the current trends in the energy market, where photovoltaic installations, wind turbines, energy storage units, electric cars, smart meters, and smart building control systems must constantly communicate with the energy grid.

This paradigm shift gave rise to Smart Energy Systems. A Smart Energy System encompasses the entire process of energy production, distribution, storage, consumption, and management. It accomplishes this via the use of digital technologies, sensors, communication protocols, and automation. Within this framework, users may become prosumers, who generate energy locally and exchange it with the grid. At the same time, demand response and demand flexibility mechanisms allow buildings and industries to respond to grid conditions or energy price variations to adapt their energy consumption

These digital interconnections enable efficiency and flexibility, but at the same time they increase the vulnerability of energy infrastructures to cyber threats. Because Smart Energy Systems are cyber-physical systems, a cyberattack on protocol, smart meter or SCADA component can lead to physical effects: failure of a device, manipulation of loads, wrong breaker operation, or even localized blackouts. For this reason, cybersecurity cannot be seen only as an IT issue but must be analyzed in relation to the physical and operational features of the grid.

Each type of a protocol used in a particular part of a Smart Energy System will have its own requirements. Thus, for instance, substation automation protocol standards

like IEC 61850 should have low-latency messages for protection functions, Building Automation protocols such as BACnet will need efficient local control combined with current needs for cybersecurity (BACnet/SC), while Demand Response protocols like OpenADR can depend on standard web security mechanisms due to less strict time limits.

The objective of this thesis is to analyze the most relevant communication protocols employed at different levels of a Smart Energy systems. In particular, the thesis focuses on the security mechanisms and the trade-off between security and efficiency, considering how security affects the performance and applicability of these protocols. The protocols examined are IEC 61850, DNP3, DLMS/COSEM, OpenADR, and BACnet/BACnet Secure Connect.

A second objective is to evaluate how security choices at the protocol level can influence the overall resilience of a Smart Energy System. To do so, the thesis applies a hybrid threat-modeling methodology based on TAMELESS, a formal threat-modeling framework for cyber-physical systems, extended with a Large Language Model and MITRE ATT&CK for ICS. This approach makes it possible to compare unsecured, partially secured, and fully secured architectures, showing how different security mechanisms can interrupt or fail to interrupt cyber-physical attack paths.

Therefore, the thesis deals with the following questions:

1. How do the main communication protocols used in Smart Energy Systems differ in terms of architecture, operational context, and security requirements?
2. Which security mechanisms are available for each protocol, and what performance trade-offs do they introduce?
3. How can formal and generative threat-modeling techniques be combined to evaluate the cyber-physical consequences of attacks on Smart Energy Systems?
4. To what extent can secure protocol variants reduce the risk of cascading attacks across the micro, edge, and macro levels of the grid?

The final objective is to show that there isn't a single security solution to address the security of the entire Smart Energy System. Instead, each context of operation requires specific security mechanisms and should be selected accordingly. At the same time, the evaluation proves that cryptography alone is not sufficient: secure communication must be combined with host-level protection, proper certificate management, and defense-in-depth strategies.

1.2 Structure of the Thesis

This thesis is organized into nine main chapters, followed by the final conclusions and future developments.

Chapter 1 introduces the concept of Smart Energy Systems. It describes the transition to decentralized infrastructures and the main components of a Smart Energy System. It also introduces the role of communication protocols and explains why different grid levels require different latency and security properties.

Chapters 2 to 7 analyze the main communication protocols used in different levels of a Smart Energy System. They introduce the operational context of each protocol and its role within the energy system, describe its architecture, protocol stack, communication mechanisms, and main use cases. Particular attention is given to the security properties of each protocol and to the trade-off between cybersecurity and operational efficiency.

More specifically: chapter 2 focuses on IEC 61850, the main standard for substation automation; chapter 3 examines DNP3, a SCADA communication protocol widely used in power distribution networks; chapter 4 analyzes DLMS/COSEM, the main standard for smart metering and energy data management; chapter 5 introduces Demand Response and Demand Flexibility, providing the conceptual background needed for the analysis of OpenADR; chapter 6 focuses on OpenADR, the main open standard for automated demand response communication; chapter 7 examines BACnet and BACnet Secure Connect, focusing on building automation and Building Energy Management Systems.

Chapter 8 introduces threat modeling and the TAMELESS framework. It presents the main concepts of threat modeling and reviews established methodologies such as STRIDE, PASTA, VAST, Trike, and DREAD. It then introduces TAMELESS as a hybrid threat modeling framework for cyber-physical systems.

Chapter 9 presents the methodology, setup, and evaluation. It first presents the architecture of the extended TAMELESS toolchain and how the TAMELESS framework was adapted to analyze Smart Energy System protocols and their security trade-offs. It then defines the evaluation methodology and compares unsecured, partially secured, and fully secured scenarios at the micro level, edge/prosumer level, and macro level.

Chapter 10: conclusions and future work. It summarizes the main findings of the thesis and discusses possible future developments.

Chapter 2

Smart Energy System

2.1 Definition and Core Components of a Smart Energy System

A Smart Energy System (SES) is an integrated infrastructure where sensors, communication networks and digital technologies are used to monitor, control and optimize in a "smart" way the generation, distribution and consumption of energy. Traditionally, power grids were unidirectional systems where a centralized power plant transmitted electricity (mainly generated with fossil fuels) blindly and over long distances to passive consumers. A SES is a new model that moves towards a bidirectional and decentralized system. Three fundamental components characterize a Smart Energy System (SES):

1. Decentralized and Distributed Generation

In a Smart Energy System, energy is generated closer to where it is consumed, leveraging localized renewable energy sources such as photovoltaics, hydropower and wind turbines. This allows to reduce transmission losses and improves the autonomy and resilience of the grid.

2. Bidirectional flow

The system creates a bidirectional flow of both electricity and information, using IoT sensors and smart meters, allowing to balance demand and supply in real time. This is even more important if we consider the discontinuous nature of renewable energy sources.

3. Active Consumer and Energy Storage

While in traditional power grids end users are a passive entity (consumers), in a Smart Energy System they can be active participants, acting as prosumers (consumers + producers) by transmitting excess energy back to the grid. Energy Storage System

play an important role in the flexibility of the system because they enable to store excess energy during periods of high generation and release it when there is a growing demand.

2.2 Smart Grids vs. Smart Energy Systems

The concept of SES is often interchanged in literature with the one of Smart Grid. However, while Smart Grids focus entirely on the electricity sector, a Smart Energy System is a holistic system integrating multiple energy sectors. Lund et al. defined a SES as an approach in which smart electricity, thermal and gas grids are combined and coordinated in a synergic way to achieve an optimal solution for both each individual sector and the overall energy system. According to Lund et al., a SES integrates three main infrastructures: Smart Electricity Grids, Smart Thermal Grids (District Heating and Cooling), and Smart Gas Grids.

2.3 Physical Architecture of a Smart Energy System

Xu et al., defined the physical layout of a SES. A SES can be conceptually divided into four subsystems: energy generation systems, energy end users, energy distribution and storage systems, and smart energy management systems.

- Energy generation systems

Energy generation systems, such as photovoltaic modules, hydropower plants and wind turbines, transform primary energy sources into secondary energy sources (electricity, gasoline, hydrogen, heat). Both fossil and renewable energy are used to ensure proper availability, since renewable energy sources (solar, geothermal, hydro, wind, and marine energy) are often unpredictable and unbalanced during the year.

- Energy end users

As previously explained, end users are now active entities that act as energy prosumers. Principal end users include residential, commercial and industrial sector, where buildings constitute the entity of primary energy demand.

- Energy distribution and storage systems

Energy distribution system is the bridge that transfers energy from energy generation systems to end users, while energy storage systems store excess energy so that it can

be released when the demand exceeds the generation. Together they allow to balance supply and demand in a flexible and dynamic way.

- Smart energy management systems

They represent the main control and management unit that integrates energy generation, distribution, storage systems and end user. Through real time data processing, it manages the whole systems and provides smart control and communication strategies in order to maintain stability and maximize efficiency.

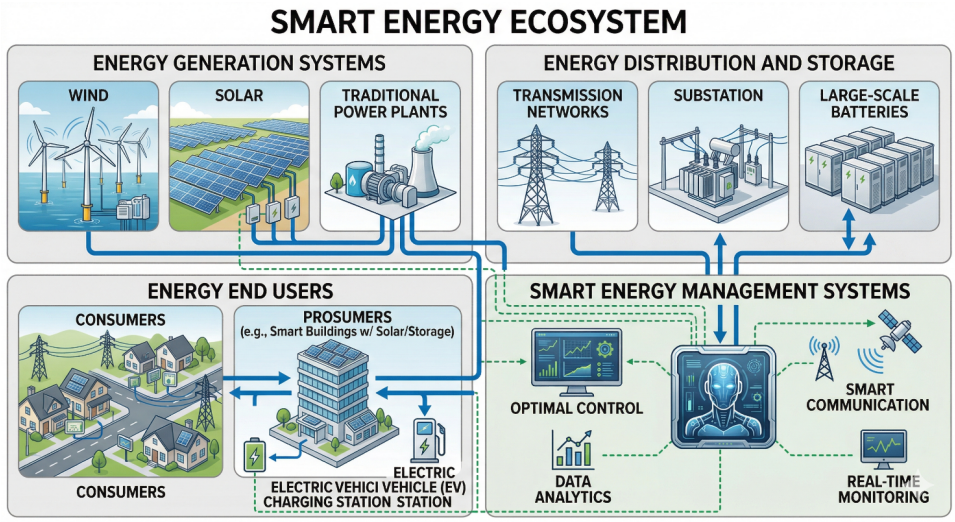

Figure 1: Architecture of a Smart Energy System

2.4 Communication protocols in SES

Due to the presence of a distributed and bidirectional flow, it's essential to define a set of communication protocols allowing the continuous and large transmission of metering data and real time data in Smart Energy Systems, from the edge to the core systems and vice versa.

SES are highly heterogeneous, therefore a single protocol cannot be used as a standalone solution for the whole grid. Each protocol satisfies a different set of needs depending on the level at which they operate. For example, a high voltage station at the generation level requires completely different data structures or speed compared to the HVAC system of a smart building at the end-user level.

While generation and distribution systems rely on heavy-duty OT protocols for absolute reliability, end-user systems use local protocols that are optimized for high-speed sensor polling, and the smart energy management system employs web-based IT protocols to bridge these domains over wide area networks.

2.4.1 Performance Requirements and Latency Constraints

In a Smart Energy System, because each level has different requirements and involves specific physical tasks, the definition of real-time can vary largely. Donitzky et al. provides in [3] an explanation of these constraints, analyzing the required data rates and maximum latencies tolerated for different SES use cases. These requirements ultimately define the architecture of communication protocols. Here is an overview of the different use cases and their requirements.

Regulation of reactive power

Renewable energy is highly volatile and when entering the grid, it can cause voltage fluctuations. Grid systems must adjust reactive power with smart inverters to keep the voltage stable. This context requires a medium level of latency as this kind of operation happens in the span of seconds.

Synchrophasor Power Management Unit (PMU)

PMU are advanced sensors deployed on the macro level (grid/transmission) to measure in real time electrical waves of a power grid, in order to detect any fault or anomaly. Consequently, this level requires really low latencies (< 10 ms) and cannot afford heavy encryption or complex PKI certificates.

Smart metering, demand response

This is the micro level (end user) that interacts with the grid. It includes reading smart meters for billing or sending Demand Response signals to a building telling it to reduce its HVAC load. Since physical actions (like a building slowly cooling down or a management system reading a smart meters at intervals) are in the order of seconds to minutes, this level of communication is much more tolerant in terms of latency and can afford some heavier security overheads.

Injection of energy into grid from distributed generation

This level involves distributed energy resources and microgrids. When a microgrid injects distributed energy into the main grid, it needs to coordinate voltage and phase data to synchronize with the macro-grid. This coordination permits a communication delay of 5 to 60 seconds, allowing loose latency requirements and the possibility of applying robust cryptography without affecting its operational capability.

Secondary Substation as Last-Mile Data Hub

This is the distribution level, where secondary substations act as last-mile data hubs, meaning they gather and process edge data before transmitting it to the core management system. They require communication latencies between 50 and 100 ms, which doesn't support heavy cryptographic overhead but allows lighter security mechanisms.

DATA RATE AND LATENCIES FOR VARIOUS USE CASES[2]

USE CASE DESCRIPTION	DATA RATE PER DEVICE	LATENCY (ROUND TRIP TIME)	LATENCY CLASS
Regulation of reactive power	5-100 kbps	0.5-5s	Medium
Synchrophasor power management unit (PMU)	> 5 kbps	<10 ms	Low
Smart metering, including demand response and DER integration (see following **use case 1**)	1 kbps	1 s-15 mn	High
Injection of energy into grid from distributed generation (see following **use case 2**)	1-5 kbps	5-60 s	High
Secondary substation as last-mile data hub (see following **use case 3**)	2-100 kbps	50-100 ms	Medium

Table 1: Data rate and latencies for various use cases, from Donitzky et al. [3]

2.4.2 Protocols Overview

This thesis will focus on the most prominent protocols that operate at each of the four levels established previously, IEC 61850 and DNP3 at the generation and distribution levels, BACnet at the end-user level, and OpenADR and DLMS/COSEM at the system management level.

In the next sections each protocol will be reviewed more closely and a high-level discussion of the trade-off between cryptographic security, computational efficiency, and network latency will be given, while Donitzky et al. analysis can serve as a low-level reference.

1. Energy generation systems (macro level)

- IEC 61850 (Substation Automation)

IEC 61850 is an international standard for substation automation which relies on GOOSE messages to execute critical and real-time commands. These messages have really strict privacy requirements because they command physical breakers to trip during electrical faults. To maximize performance, data is transmitted in clear text, while a newer extension (IEC 62351) adds security features such as encryption and digital signatures. However, the encryption and authentication overhead can sometimes be a threat to the latency required.

- DNP3 (Distributed Network Protocol)

DNP3 is the main protocol used in SCADA (Supervisory Control and Data Acquisition) systems to monitor and control generation and distribution grids. Classic

DNP3 was designed for efficiency over serial links without any native security. A later version, DNP3 Secure Authentication (DNP3-SA), was introduced to defend against spoofing and injection attacks by using HMACs and challenge-response mechanisms for authentication, without relying on heavy TLS handshake and encryption.

They are typically complementary: IEC 61850 deals with communication in a substation while DNP3 from the substation to the control center. They could overlap in distribution automation where devices are spread over a wider area and both protocols could technically work, but DNP3 is typically used in this case.

2. Energy distribution and storage (macro - meso level)

- DLMS/COSEM (Smart Metering)

DLMS/COSEM is the predominant international standard (IEC 62056) for communication in smart meters and energy management (electricity, gas, water, heat), acting as the link between distribution systems and end users. Smart meters are usually constrained and low power edge devices, so efficiency is one of the most important requirements. DLMS/COSEM uses a set of security suites to secure the communication: as base level it implements High Level Security (HLS) with challenge-response authentication; for additional security, other suits employ Authenticated Encryption (AES-GCM), digital signatures and Role-Based Access Control (RBAC). In scenarios where edge devices have battery limitations or operate on congested legacy networks, only a base level of security with basic authentication and no encryption is used to maintain real time efficiency.

3. Energy End Users (Micro Level – Smart Buildings)

- BACnet (Building Automation and Control networks)

BACnet is historically the most common communication protocol used in Building Energy Management Systems (BEMS) to manage HVAC and lightning. It natively lacked any security protection as messages are sent in cleartext and without any authentication (allowing many attacks). It prioritized minimal overhead using lightweight and connectionless UDP packets.

- BACnet/SC

BACnet Secure Connect (BACnet/SC) is a recent BACnet extension that introduces security leveraging Secure WebSockets over TLS 1.3, adding authentication with X.509 certificates and encryption to the original protocol version. In many cases it's not a standalone solution and thanks to its backward compatibility it's integrated

with native BACnet/IP and BACnet MS/TP, still used in many edge devices that cannot handle the initial certification validation latency and a TLS session.

4. Smart energy management systems (control layer)

- OpenADR (Open Automated Demand Response)

OpenADR is an open standard for the exchange of Demand Response signals between smart energy management systems and BEMs. It is an application-level standard that runs over standard IT networks and manages security by leveraging standard HTTPS. Because Demand Response events (such as demanding a building to reduce HVAC load during peak hours) have highly relaxed latency constraints (from 1 second up to 15 minutes), the network can afford the computational overhead and the handshake delay of a TLS encryption without risking the operational efficiency of grids.

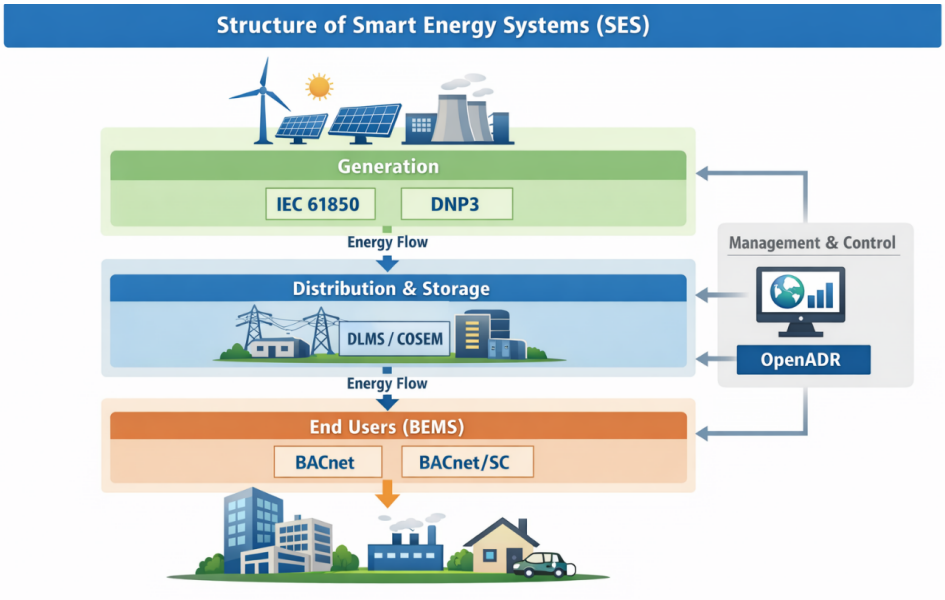

Figure 2: Mapping of main protocols of a SES into its architecture

Chapter 3

IEC 61850

IEC 61850 is an international standard for communication in substations. While other protocols like DNP3 are mainly involved with remote operations between SCADA systems and remote units, these are not suited for the high reliability and low latency required by substations. IEC 61850 solves this issue and enables the integration of measurement, monitoring and control within substations, using Ethernet-based communications. Its object-oriented data model, together with the definition of a universal language for substation configuration (SCL), allows interoperability between different manufacturers.

3.1 Substation definition

In a power grid, an electrical substation is the station where voltage is transformed and electricity is switched, controlled and protected. Transmission substations operate at high voltages and connect generation facilities with the transmission network, while distribution substations drop down voltage for the delivery to end users. Substation perform essentially three critical functions: voltage transformation, switching operations to reconfigure topology, protection operations to detect and isolate faults before failure or equipment damage. The main elements of substation include power transformers for voltage conversion, circuit breakers and disconnect switches for isolation and fault interruption, instrument transformers for measurement and protection, and protection relays and control system for automatic decisions.

3.2 Extension to other domains

IEC 61850 was developed as a communication standard for Substation Automation Systems (SAS), but because of its interoperability and data modeling features, it extended from its traditional scope. IEC 61850-7-420 (2009) extended the standard

to Distributed Energy Resources (DER), enabling communication at the generation and distribution levels with solar inverters, wind turbines, and battery storage systems. IEC 61850-90-5 (2012) introduced routable versions of the standard protocols (R-GOOSE and R-SV) for wide-area communication of data generated by PMUs (Phasor Measurement Unit) over transmission networks, extending the standard to the monitoring and control level. IEC 61850-90-8 (2016) addressed Electric Vehicle (EV) charging infrastructure at the distribution and customer levels. Recently, the standard also extended to smart meters and demand response systems, traditionally the domain of other protocols, like DLMS/COSEM, the dominant standard in Europe for these functions. Therefore, IEC 61850 can be considered as a comprehensive framework for all these levels.

3.3 Data Self-Description

Data Self-Description is a fundamental principle of IEC 61850 standard. It means that the data exchanged contains all the information needed to interpret it correctly without requiring any mapping list, like the type of data, its meaning, the unit of measurement, and other relevant metadata. This is one of the key elements for interoperability, because different models of devices can understand the structure of each other's data, making communication more efficient. The key aspects that enable data self-description in IEC 61850 are:

- The Data Object Model: data is grouped into conceptual objects like Logical Nodes, Data objects and Data attributes. Instead of relying on a list of anonymous registers, this logical organization defined by the standard automatically tells how the data is structured. Each device in IEC 61850 will map its internal data structure to these standardized data objects.
- Common Information Models (CIM): define the real world meaning of the data represented with the data object model. Different devices can therefore understand not only the structure of the data (defined by the object model) but also the semantics of the data they are exchanging. For example, a simple Boolean variable in a certain data object can be understood as "the circuit breaker is or is not in the closed position"

3.4 Substation configuration and architecture

IEC 61850 defines the Substation Configuration Language (SCL), a description language based on XML that can be used to configure a substation in terms of definition, for each element in the substation, of its functionalities and interactions. SCL files enable interoperability as they permit to exchange configuration data

between an IED Configuration Tool (ICT) and a System Configuration Tool (SCT) from different vendors.

The substation automation system is divided into three levels of communication: The Station Level, The Bay Level and The Process Level.

- Station Level: it contains management devices to configure Human Machine Interfaces and SCADA systems, and it connects the substation with a remote control room to enable manual operations and monitoring over the substation.
- Bay Level: it consists of Intelligent Electronic Devices (IED) which connect station and process levels. IEDs receive data from the process level and take local decisions accordingly
- Process level: it is composed of switchyard devices (the physical equipment that monitors and controls the high-voltage electricity), such as sensors, actuators, merging units and transformers. This equipment collects system data and sends it to IED devices at the bay level for automatic operations.

IEC 61850 defines two types of buses, which are used to connect these three levels: the process bus, connecting process level with bay level, and station bus, connecting bay level with station level. The three main communication protocols specified by IEC 61850 are:

- MMS (Manufacturing Message Specification), for communication between station level and IEDs at bay level
- GOOSE (Generic Object Oriented Substation Event), for communication between devices within the bay, at really low latencies (2-4 ms)
- SV (Sampled Values), to transmit measurement samples (such as voltages currents) from process to bay level.

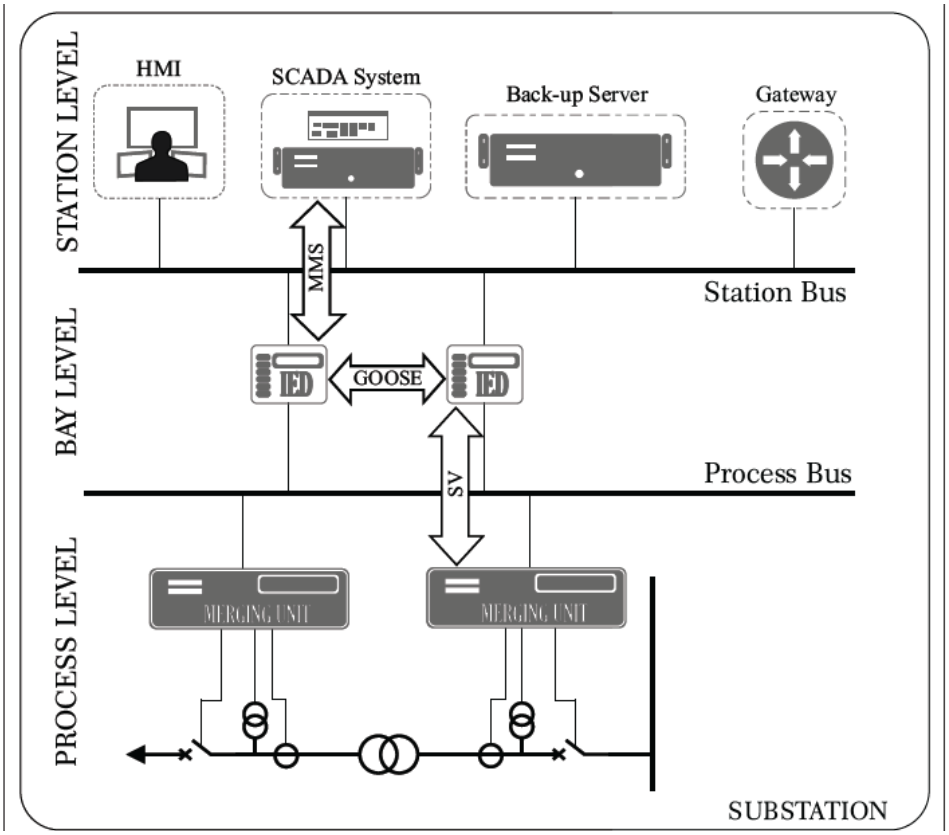

Figure 3: Substation architecture and internal communication protocols in IEC 61850, from [5]

3.5 Communication protocols

3.5.1 MMS

The Manufacturing Message Specification (MMS) protocol operates at the application level through a client-server model for communication, between the control room at the station level and bay-level devices. Before the transmission can take place, MMS expects an initial association to agree on what and how information is shared. The messages exchanged are request/respond to control data and reports to transmit measurement data. This transmission of messages provides a range of functions that allows the client to get the data model of the server, read or modify single values, delete entries and transfer files**.** While other protocols in IEC 61850 run over Ethernet or UDP/IP, MMS is the only one which runs over TCP/IP. This can be explained by the fact that the control room at the station level can be remotely connected, and so a reliable connection is needed. Typical latencies accepted are between 1-4 seconds since it deals with non-urgent control signals (which can tolerate overhead of connection-oriented TCP)

3.5.2 GOOSE

The Generic Object Oriented Substation Events (GOOSE) protocol is used for fast transmission of substation events (such as commands, alarms, indications, as messages) IEDs at bay level. It operates directly over Ethernet and It uses VLAN and priority tagging techniques to separate a single physical network into many virtual networks and set a priority level for messages. It employs a publisher-subscriber mechanism through multicast or broadcast MAC addresses, where the publisher periodically sends data. GOOSE uses datasets and when the value in any dataset changes, during an event, a change of state is detected and it's immediately published. The timing of these two types of messages (periodic messages when no events occur and publishing messages when an event occurs) are defined with Time Max and Time Min parameters. Time Max is in the scale of seconds, while Time Min is between 1 and 4 milliseconds, since events need to be instantly published. Other than high messaging speed, another characteristic of GOOSE is reliability, achieved with a retransmission strategy and integrity checks.

3.5.3 SV

Sampled Values (SV) protocol is used for information exchange between Merging Units (which convert analog voltage and current signals from transformers into digital packets) at process level and IEDs at bay level. Similarly to GOOSE, it operates directly over Ethernet, and it uses a publisher-subscriber model with multicast messages. More specifically, the publisher sends periodic messages (where the time interval depends on the parameters Samples Per Period, SPP, and signal frequency) under a specific topic. The subscriber filters from all the messages received from the system only the ones under the subscribed topic. SV requires a low latency (< 3 milliseconds) and time synchronization requires microsecond-level accuracy, for accurate protection of relay operation.

3.5.4 R-GOOSE and R-SV

Routable GOOSE (R-GOOSE) and Routable SV (R-SV) are adaptations of the respective GOOSE and SV protocols for message exchange between different substation over long range. They both work by taking the standard GOOSE/SV payload and encapsulating it inside UDP/IP packets. Now, because they have a layer 3 IP address, they can pass through routers and traverse WANs.

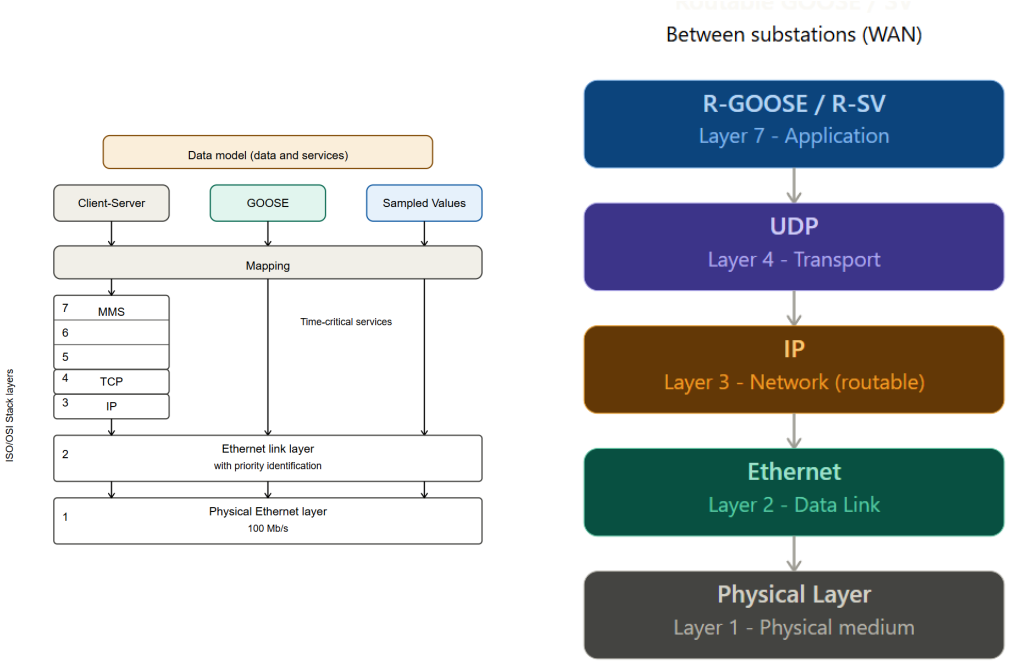

Figure 4: ISO/OSI layer mapping in IEC 61850, Adapted from: ABB Technical Application Guide No. 19, "IEC 61850 Standard" (2017)

3.6 Object-Oriented Data Model

IEC 61850 standard employs an object oriented technique that makes the data model reusable and inheritable. Each layer in the data model is defined as an abstract class which encapsulates a set of attributes that describe the characteristics of the class, and services that provide methods to operate on these attributes. The standard defines a hierarchical structure of nested elements, where each element encapsulates other basic elements.

- The server: it's the top level abstraction and its primary job in the architecture is to handle network connections (acting as a SCADA master talking to clients) and be a container for the lower layer represented by Logical Devices.
- Logical Devices: collection of data objects used to describe various devices and functions in a power system. It acts as a sub-container to keep unrelated functions cleanly separated within the same box.
- Logical Nodes: object that represents one specific, real-world device or automation function. They contain data objects, which are sub-elements of logical nodes.
- Data Objects: they describe specific types of information that belong to the Logical Node; they can be basic data types like Char, Boolean or Integer. The Data object can contain several data attributes.

- Data Attributes: contain the actual hard values, timestamps, and metadata used to represent several characteristics of the data. Every Data Object is made up of multiple Data Attributes so that the data is fully self-describing.

In a complete data path in IEC 61850 it's possible to see all these layers together. For example, given this path "Relay1/XCBR1STPos$stVal", Relay1 is the Logical device containing the automation function, XCBR1 is the Logical Node that represents the specific physical circuit breaker, Pos is the Data Object holding the information about the breaker's position and stVal is the Data Attribute carrying the actual True/False state of that position. The "ST" in the middle is a Functional Constraint (FC). A FC is a standard tag used to categorize the data based on its access rights and purpose. In this example, ST means "status", telling the receiving client system that this data is a read-only state.

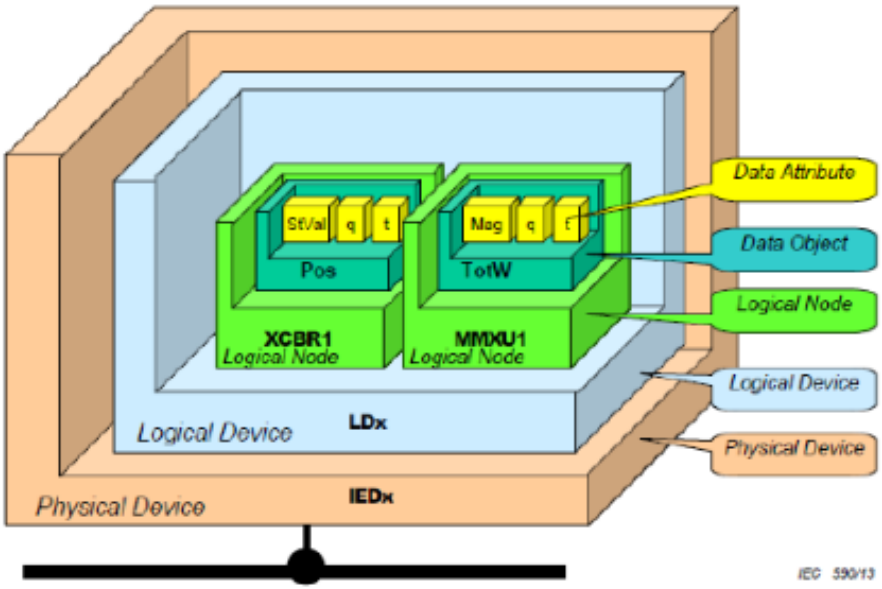

Figure 5: Data modeling structure of IEC 61850, from [12]

3.7 Message types

The IEC 61850 standard organizes network communications in seven different Message Types and three Performance Classes (P1, P2, P3) that define transfer or processing time for each message type. Performance Classes depend on the urgency and complexity of the communication, where P1 is referred to standard distribution bays, while P2 and P3 for highly demanding transmission bays that require fast processing. The seven message types can be categorized based on their operational urgency (and corresponding latency requirements):

- Time Critical (types 1 and 4): these types of messages deal with critical operations like immediate breaker trips and blocks or data sampling from transformers. In highly demanding performance classes P2 and P3, these messages require a low latency under 3 milliseconds, to protect the equipment.
- Non-critical (Types 2, 3, 5): these are routine messages for everyday operations that don't require strict latency requirements, such as state information updates ($\leq$ 100 ms), slow auto-control functions ($\leq$ 500 ms) and large file transfers about recordings or settings ($\geq$ 1000 ms).
- Specialized support (Types 6 and 7): these messages support the system's infrastructure. Type 6 messages are used to synchronize the internal clocks of IEDs, while Type 7 handles the transfer of access control orders for highly secure contexts.

3.8 Security

While traditional approaches were based on "security by obscurity", this approach is no longer valid in today scenarios, where IEC 61850 is now employing standardized and well known communication protocols and it has been extended to WANs. As [4] underlines, the main reasons for this change in the paradigm are:

- The interoperability and standardization of communication protocols (models are known without any obscurity)
- In deregulated markets, financial information like bids, prices and generation capacity is exchanged through communication systems. This means that manipulated power system data can lead to disruption in market operations and competitive advantage
- The integration of power system with wide area networks (with known models) has made this infrastructure vulnerable to many cyber attacks.

3.8.1 Attacks examples

In utility automation systems, the data exchanged can carry critical information that has a direct impact on the physical world: any security vulnerability can cause very serious effects. This is the main reason why a cyber attack in a utility system is much more critical than in a normal internet system. Hussain et al. [4] analyze some of the possible attacks against IEC 61850 protocols:

- GOOSE Poisoning Attacks: two critical parameters, status number and sequence number, can be easily manipulated by attackers by sending spoofed

GOOSE messages, like in High-Status Number Attack and Status Number Flooding. GOOSE is also vulnerable to a semantic attack where the attacker first observes normal traffic patterns and then sends multicast spoofed messages to prevent processing of legitimate messages.

- Additional attacks against GOOSE/SV: Replay and tampering attacks can be executed by capturing and retransmitting old or modified GOOSE/SV messages. It's also possible to inject false data in SV messages to cause incorrect operational decision or perform masquerade attacks where intruders can impersonate legitimate devices and send false commands.
- MMS Message Attacks: ARP spoofing can be used by attackers to initiate a MITM attack and enable secondary attacks like eavesdropping, data injection and masquerade. Since MMS operates on top of TCP/IP, the initial session establishment is vulnerable to SYN flood attacks, causing Denial of Service.
- Physical attacks: IEDs devices can be affected by fault injection attacks, introducing computation errors (using invasive or noninvasive technique) that can affect the availability or integrity of the smart grid.

3.8.2 IEC 62351

IEC 62351 is a standard that provides cybersecurity measures for many IEC TC 57 communication standards, including IEC 61850. The standard currently has 16 parts, the main ones relevant to IEC 61850 will be described below.

- **IEC 62351-3: Profiles Including TCP/IP**

It specifies measures for confidentiality, integrity and authentication at the transport layer for SCADA protocols that employ TCP/IP. In the context of IEC 61850, it directly applies to MMS protocol. Using standard TLS security, it aims to provide end-to-end security (with encryption and message authentication codes) peer authentication through X.509 certificates and key exchange.

- **IEC 62351-4: Profiles Including MMS and Derivatives**

This part of the standard specifies security requirements for MMS messages both at application and transport level. Regarding the transport level security (T-security), it defines two modes of operation: Compatibility Mode, for MMS messages operating with OSI stacks (legacy equipment), and Native mode, for MMS using IP suite (modern equipment using TCP/IP) and IEC 61850-8-2 XMPP implementations. For application level security, there are two options: peer-to-peer (or A-security) provides authentication during association establishment and no confidentiality or integrity for

the following data transfers (requires T-security at the transport level). End-to-end security (E2E security) provides end-to-end data authentication and integrity using public key signatures and optionally symmetric encryption for confidentiality.

- **62351-6: Security for IEC 61850 (GOOSE and SV)**

IEC 62351-1 indicates integrity and authenticity as two key security properties for IEC 61850 GOOSE and SV messages. Encryption, on the other hand, should not be applied because the processing overhead would be too high in IEDs to ensure the low latency up to 3 milliseconds that is required for these protocols. IEC 62351-6 proposes the use of digital signatures with RSA and SHA256, where for every GOOSE/SV message, a DS is generated and appended to the message in a dedicated Extension field. Another alternative would be using a MAC algorithm instead of a DS (like HMAC-SHA256), which would replace the DS in the Extension field. While MACs are more efficient, they cannot provide non-repudiation, or a secured channel for shared keys

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

Defines NSM data objects that monitor health and condition of devices such as IEDs, RTUs (Remote Transmission Units) and DERs (Distributed Energy Resources). The information provided can also be used by intrusion detection systems.

- **IEC 62351-8: Role-Based Access Control (RBAC)**

Implements RBAC using pre-defined roles (viewer, operator) and access rights (view, read, control) so that users can only perform the needed actions on allowed objects to perform their duties. It applies both to human users and automated systems and it can handle local or remote access sessions

- **IEC 62351-9: Cyber Security Key Management**

It specifies techniques for key management, including generation, distribution and revocation of public key certificates and handling of cryptographic keys, useful for other algorithms in different IEC 61351 parts.

3.8.3 Security Tradeoffs/Applicability

3.8.3.1 Digital Signatures vs. MAC Algorithms for GOOSE/SV:

Hussain et al. analyze in depth the comparison between MAC and Digital Signatures to provide message integrity and authentication. Even if, as already mentioned,

DS provides better security assurances such as non-repudiation, the performance overhead is not suitable for low latency messages:

Table 2: Comparison of Security Measures for GOOSE and SV

Aspect	**IEC 62351-6:2007 (RSA)**	**IEC 62351-6:2020 (MAC)**
Security Measure	RSASA-PSS based signatures	MAC based authentication
Performance	Very high computational times (0.942–10+ ms for signing with 1024-bit key), making it unsuitable for GOOSE and SV	Good performance with low computational time (0.0127 ms for HMAC-SHA256)
Extension Size	128 bytes (1024-bit key), 256 bytes (2048-bit key)	8–32 bytes
E2E Delay	Does not meet 3 ms requirement	0.0664–0.1027 ms (meets 3 ms requirement)
Key Type	Asymmetric (public-private key)	Symmetric (pre-shared keys)
Pros	Non-repudiation, no pre-shared key needed	Low computational overhead, smaller signature size
Cons	High computational time, large signature size	Requires pre-shared keys, cannot prevent compromised insider attacks

3.8.3.2 Confidentiality Considerations

As anticipated, the standard mandates that time-critical messages in GOOSE and SV shouldn't apply standard encryption algorithms since they impact the low latency requirements of 3ms. However, since IEC 61850 extended not only to substation automation systems but also to power utility automation (DER, smart meters, electric vehicles) sensitive data can be carried over IEC 61850 messages. In this case, the context of operation is at the smart metering/demand response and distributed generation levels identified by Donitzky et al. [3], where latency requirements are not as strict (order of seconds). Therefore, it's always important to evaluate the specific scenario of application, meaning that appropriate encryption algorithms can be applied when the protocol is not applied to the stringent 3ms timing requirements of substation automation. Appropriate encryption algorithms can and should be applied when:

- The protocol is not subject to the strict timing requirements of substation protection functions (< 3 ms)
- It applies to categories that are more tolerant in terms of latency (smart metering, distributed generation coordination)
- Sensitive commercial or operational data requires confidentiality protection

3.8.3.3 Performance Measurements of IEC 62351 Cipher Suites

Todeschini et al. [6] developed an experimental platform to measure the impact in performance of different IEC 62351 cipher suites on IEC 61850 MMS communications. The two performance indicators adopted were handshake time (latency from connection attempt to connection completion, that includes the initial TCP and TLS handshake, certificate exchange and session key generation) and report transmission time (latency to transmit MMS reports of 125 bytes, that includes encryption, transmission and decryption). Tests were conducted on libiec61850 v1.2 platform with TLS v2.6.0 and intel i7-6700 processor.

The key findings of the experiment are the following.

- The selection of cipher suite has a high impact on the handshake time, with TLS_RSA_WITH_AES_128_CBC_SHA256 being the fastest cipher suite (21 ms mean handshake time) and TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 the slowest (30.8 ms), with more than 1300% difference in performance.
- The key exchange algorithm is also a dominant factor: suites based on ECDHE perform much slower than the alternatives based on DHE or RSA.
- The impact of key length was minimal (less than 1% variation) between 1024 and 2048 bits, noticeable (+10%) at 4096 bits, and significant (+100%) only at 8192 bits
- Report transmission time, on the other hand, showed minimal variation across all cipher suites, probably caused by modern hardware acceleration equalizing encryption and decryption performance.

This experiment indicates that the main performance overheads happen mainly during the initial session establishment and not during data transmission, suggesting that the selection of the cipher suite should depend more on the connection pattern: faster suites for frequent reconnections, while stronger suite for long persistent sessions.

Chapter 4

Distributed Network Protocol

The Distributed Network Protocol version 3 (DNP3) is an open communication protocol for Supervisory Control and Data Acquisition (SCADA) systems in power distribution networks. It was developed in Canada in 1993 and since then it has been adopted in many power utilities, specifically by the SCADA Master Station to manage and control remote devices like Remote Terminal Unit (RTU) and Intelligent Electronic Devices (IED). DNP3 can operate over multiple physical media, including serial (RS-232/485) and Ethernet (TCP/IP), and is structured on multiple layers to ensure reliable data transmission. It's important to notice that DNP3 is not a general purpose protocol but it's specifically designed and intended for SCADA applications. A more recent protocol layer called DNP3 Secure Authentication (DNP3-SA) was later developed to address the lack of security.

4.1 DNP3 protocol stack

Because DNP3 was originally designed for serial communication where TCP/IP didn't exist, traditional DNP3 implements 3 layers of communication: DNP3 Application layer, DNP3 Transport Layer and DNP3 Data Link layer, which operate directly on top of the Serial physical layer such as RS-232.

However, in modern applications, DNP3 operates on top of TCP/IP over ethernet. In this case, the three DNP3 layers are folded into the application layer: DNP3 becomes essentially an application-layer protocol when running over TCP/IP. This double possibility of application (TCP/IP network or serial bus link) provides a good degree of flexibility that is not common to other protocols in this field.

- DNP3 Application Layer

Creates Application Protocol Data Units (APDUs), made of an Application Service

Data Unit (ASDU), the actual payload, and an Application Protocol Control Information (APCI) containing control headers. Main functions of the application layer are: data object handling, command processing, event management, time synchronization and file transfers. In DNP3-SA it also implements security features such as challenge response authentication, encryption and session key management.

- DNP3 Transport Layer

The primary functions of this layer are: message segmentation and reassembly (to break and later reconstruct APDUs into smaller Transport Protocol Data Units that fit in the Data Link layer) and ordered transmission (using fragment sequence numbers). Since it was designed for efficiency on serial links with minimal bandwidth, the overhead is minimal.

- DNP3 Data Link Layer

This layer handles physical addressing, error detection, flow control (with a request/confirmation mechanism) and frame formatting. In modern deployment where DNP3 operates over TCP/IP networks, this layer is usually removed because many of its features are already provided by TCP and would be redundant, adding useless overhead.

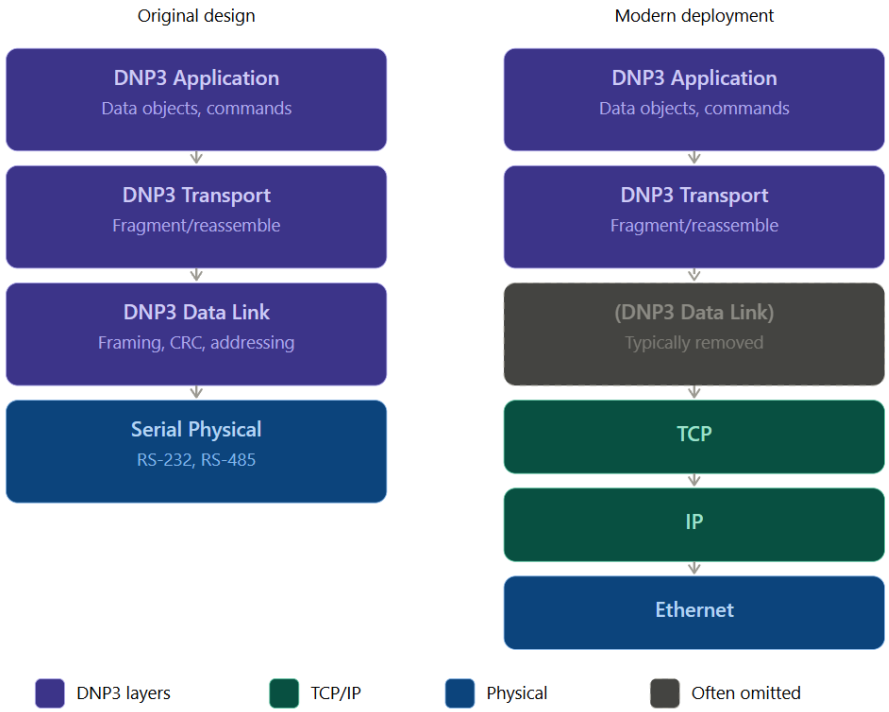

Figure 6: DNP3 Protocol Stack Deployment Options

DNP3 defines two types of endpoints that communicate with each other: a master and a remote unit. Here's a definition of each:

4.2 DNP3 Communication Architecture and Operation

DNP3 follows a master-outstation (slave) architecture where a single or multiple masters are connected with one or more remote units called outstations. Communication rules allow the master and remote unit computers to communicate over limited network bandwidth to transport data values and simple commands between two ends of the system.

The Master: provides the interface between the monitoring system and human operators. Typically, it initiates control commands to request data from or to actuate devices managed by the outstation. It stores all data from the remote units for later processing.

Outstations: remote units (RTUs and intelligent electronic devices), also called slaves. They are the interface between the master and physical devices being monitored or controlled. These remote units collect information from various field devices (such as current and voltage sensors) and transmit the data back to the master station.

4.2.1 Polling and Prioritization

Masters can use a polling mechanism to update their database with the latest system states. DNP3 defines data variables by type and behavior, allowing to specify what kinds of data the master wants to receive. Static data refers to the most recently measured or calculated data points, while event data refers to any significant activity (state changes, new information or data that has gone past a certain threshold) and can be further prioritized into classes. These rules and values are set by the master at startup via an integrity poll, which asks the remote unit to send values and states of configured points to the master. After this initial configuration process, the remote unit selectively transmits events based on whether the data has changed since the previous poll. These transmissions often occur cyclically but can also occur spontaneously through unsolicited responses when certain parameters are met or when crucial data changes need to be reported immediately.

DNP3 also includes some basic time management features, such as synchronizing outstation clocks, recording when data was collected using timestamps, and capturing snapshots of data from multiple outstations at the same moment for comparison.

4.3 DNP3 Security Mechanisms

The protocol was designed to be highly reliable, but not to withstand attacks from hackers or anyone who might want to compromise the control system of some essential infrastructure. Since smart grid applications generally assume third-party access to

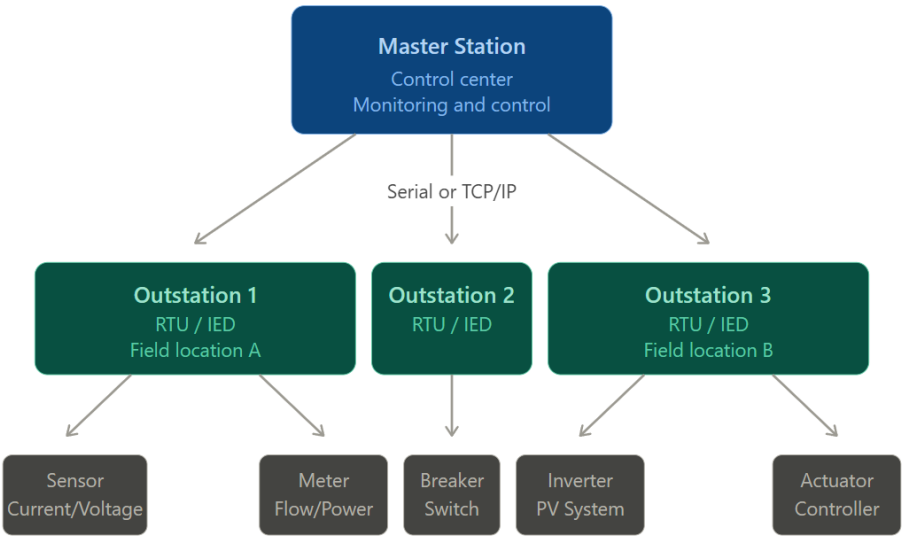

Figure 7: DNP3 Master-Slave Architecture

the network and the underlying IP-based infrastructure, unsecured standard DNP3 is vulnerable to many kinds of attacks such as eavesdropping, Man-in-the-Middle, DoS, spoofing and replay attacks. Because of this lack of security, which is now mandatory on current infrastructures, DNP3 Secure Authentication (DNP3-SA) was developed.

4.3.1 DNP3-SA architecture

DNP3-SA is introduced as a separate protocol layer between the DNP3 Application Layer and the DNP3 Transport function. It's important to notice that because DNP3-SA is a separate layer, it can also be used by other protocols than DNP3. The Authorization Management Protocol (AMP) is a complementary protocol that is used together with the DNP3 Application Layer and DNP3-SA to manage access control functions and role-based access control (RBAC).

4.3.2 Overview of DNP3-SA security features

- Entity authentication: a Challenge-Response protocol based on shared keys is used for bidirectional authentication, so that both master and outstation can verify each other. There are two possible operation modes:
- Non-Aggressive Challenge Response (NACR): traditional challenge response for critical (in terms of security) operations
- Aggressive mode (AGM): Pre-emptive authentication for faster operations and acceptable level of security

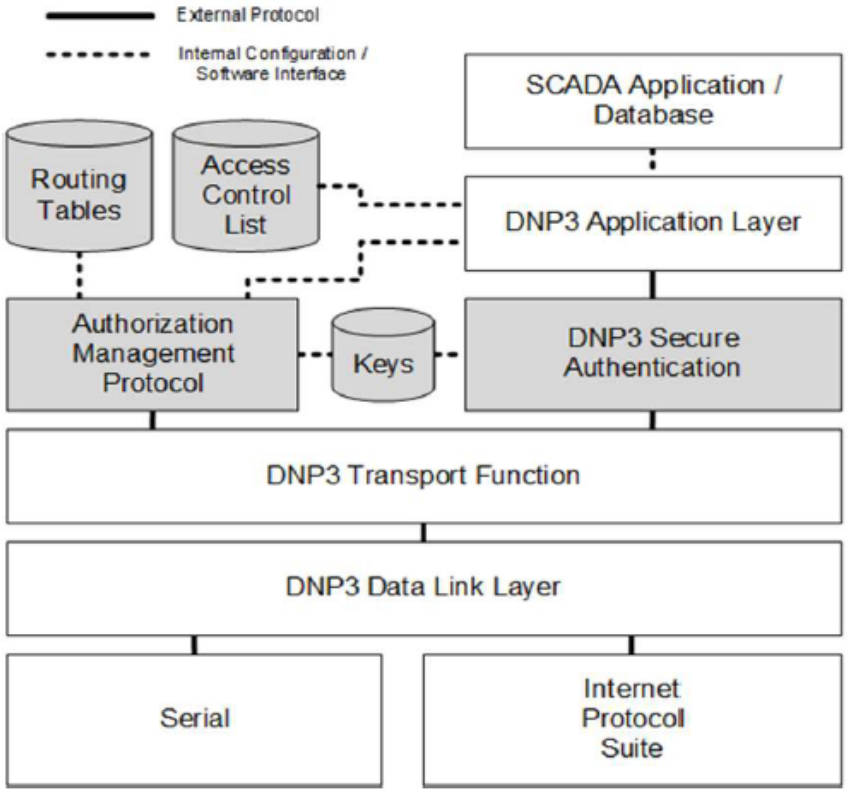

Figure 8: DNP3-SA Architecture from [15]

- Message confidentiality and integrity: traditional DNP3-SA (up to version 5) separately encrypt (optional) and compute a Message Authentication Code (MAC) to protect exchanged messages, while the updated version 6 applies Authenticated Encryption with Associated Data (AEAD) with AES-256-GCM, to provide both confidentiality and integrity in a single operation.
- Key management: key distribution and device enrollment (initial key establishment) are automated, without requiring manual key management burden. There are three possible types of keys, depending on the specific application:
- Update Keys: long-term keys for key distribution
- Session Keys: medium-term keys for operational periods
- Control Session Keys: short-term keys for specific control operations
- Anti-replay protection: timestamps for time-based validation and sequence numbers for unique identification of messages prevent replay attacks. DNP3-SA version 6 also introduced Challenge Sequence Quality (CSQ) protection, that prevents CSQ manipulation attacks (TB2016-002 vulnerability discovered by Amoah et al. in 2016), making sure challenges cannot be reused
- Access control (optional): different authorization levels for different users allow to control who can execute which commands and can support organizational

policies. The AMP protocol implements role based access control (RBAC), informing outstations of corresponding permissions and roles of masters assigned by the authority. Outstation also have the possibility of configuring Access Control Lists to enforce permissions at a per-point level.

- Security statistics (optional): monitors authentication failures and tracks all security events to facilitate the detection of attack attempts.

4.3.3 DNP3 Security Architecture Alternatives

Other DNP3-SA, two other possible security architecture exist to secure DNP3 communication. Bump-in-the-Wire (BITW) leverages IPsec and utilizes network gateways at master and outstation to add security over a WAN. This architecture doesn't require to change existing OT systems, but it adds extra hardware overhead and leaves the final cable connection (between the IPsec gateway and the field device) unprotected. Another approach is Bump-in-the-Stack (BITS), which is usually implemented with TLS, integrated in the end devices and transparent to other applications. However, BITS is characterized by a difficult key management and causes an overlap of OT and IT responsibilities, since now security is embedded into devices which are traditionally managed by the OT staff. On the other hand, the integrated protocol security approach of DNP3-SA integrates security directly in the application protocol and it's independent of transport operations, working both on Ethernet or serial cables. It also introduces a lower bandwidth consumption since it secures only the application layer instead of the whole session, with the only disadvantage of not being as mature and confirmed as standard TLS or IPsec.

4.4 Security vs Efficiency Tradeoffs

4.4.1 Bandwidth Efficiency and Authentication Overhead

Rosborough et al. (2019) analyzed in [18] the performance implications of the three DNP3 security options, about bandwidth efficiency, latency addition, key management complexity. Regarding bandwidth efficiency, DNP3-SA achieves a lower bandwidth consumption compared to other wrapped protocols, because security is embedded in the application level. This is crucial for legacy systems with low bandwidth (25 kbps) and high latency (300+ ms), still present in many SCADA networks.

For what concerns the two possible Challenge Response authentication models in DNP3-SA, NACR requires two additional message exchanges per critical operation (challenge and HMAC response), but on poor communication links this can cause cascading failures when masters send retries. AGM instead can reduce the authentication overhead for scenarios which require low latency or a lower level of security.

However, as demonstrated by Amoah et al. (2016), this efficiency optimization can introduce vulnerabilities, like the CSQ manipulation vulnerability (addressed by DNP3-SA version 6).

4.4.2 Key Management Automation

A clear advantage of DNP3-SA is the automated key management feature, as The DNP Users Group emphasizes in [15]: on one hand it reduces management complexity, on the other hand it also improves security as keys can be safely updated more frequently than they would in the manual approach. Update keys are automatically updated via authority certification keys using X.509 certificates, so only these certification keys need manual management (but they are rarely updated).

4.4.3 Physical Security and Last-Mile Vulnerability

Another important consideration, when comparing DNP3-SA to BITW with IPsec, is that the 'last mile' physical vulnerability of BITW is critical as it leaves a cleartext cable between the outstation and encryption device. An attacker who is able to physically access the enclosure, specifically in large distribution network, can disconnect the cable and connect an unauthorized device. In this way the attacker can potentially gain direct access to the master control center. This vulnerable segment is eliminated by the integrated approach of DNP3-SA on the application layer.

4.4.4 Real-World Deployment Experiences

Nevertheless, the specific choice for the security architecture is not so obvious and is up to the security administrator, depending on the specific context and configuration. In fact, Rosborough et al., regarding a real-world deployment scenario coming from Exelon (American public utility headquartered in Chicago), explained how both security approaches (DNP3-SA and IPsec) had their pro and cons. DNP3-SA has proven to be highly sensitive to network conditions and performed poorly on unstable or congested networks, where even few misbehaving devices could cause denial of service for others. On the other end, Exelon's IPsec deployment was successful in securing thousands of devices, but it required network with Round Trip Time (RTT) of under 1 second: any RTT past 3 seconds on any single device caused denial of service. In any case, the authors suggest to not put all devices through one central security gateway, especially for large utilities, and to use instead multiple regional gateways to avoid bottlenecks.

4.4.5 Integration with Intrusion Detection Systems

Rosborough et al. note in [18] another important tradeoff when analyzing the BITS approach with TLS: TLS 1.3 mandates confidentiality with perfect forward secrecy, which makes it impossible to recover data for purposes of analysis by intrusion detection systems. DNP3-SA can instead be configured to apply only authentication but not encryption (which is optional), meaning it has the additional option to integrate well with IDS/IPS systems.

4.4.6 Security vs latency requirements

The security mechanisms implemented in DNP3-SA align well with the latency constraints of the contexts where DNP3 operates. As Donitzky et al. explains, DNP3 is located at the distribution level, where the latency requirements for communication are in the range of 50-100 milliseconds. These requirements align with the overhead introduced by DNP3-SA's security mechanisms: challenge-response exchanges add some modest delay, but not as much as, for example, asymmetric key exchanges with certificates for public key validation (as other protocols that operate in other levels of a SES can afford).

DNP3-SA mandates only core security functionalities like message integrity, entity authentication and anti-replay protection, while many features can be configured as wanted by the utility, to balance security with bandwidth and latency constraints depending on the scenario. For instance, secondary substations with 50-100ms latency requirement can choose to use lightweight protection (AGM, no encryption...), while distributed generation coordination with 5-60 second windows can tolerate higher security without impacting its operations.

4.4.7 Conclusion

To conclude this discussion, as research evidence shows, no single security architecture is optimal for every single SCADA deployment. However, DNP3-SA surely adapts better than the other configurations thanks to its flexibility in configurations: it might not be the best for any case, but it can be tuned for numerous environments optimally.

Chapter 5

DLMS/COSEM

5.1 Overview

DLMS/COSEM (IEC 62056, EN 13757-1) is the global standard for smart energy metering, control, and management. It specifies an application layer protocol with an object oriented data model and media-specific communication profiles. This standard can be applied across all utility and energy types, market segments and application, over virtually any communication mean.

The core of this standard is the object oriented model, providing a standardized way to represent devices, data, and functions. This methodology guarantees extensibility and interoperability over all types of metering systems.

5.2 Comparison with Other Standards and advantages

Historically, to model metering data and enable communication, proprietary solutions like ANSI C12.19 utility tables were used by utilities. Many of these solutions were dependent on specific lower level communication means and are therefore limited in portability, flexibility and scalability, especially in the current scenario where utility demands are always more complex**.**

DLMS/COSEM approach is different as it is an international and open standard that supports a wide range of solutions. Differently from proprietary approaches, DLMS/COSEM supports multi vendor compatibility and reduces vendor lock-in. The protocol guarantees security properties and regulatory compliance and because of the extensive support (global community of over 300 member organizations with more than 30 years of refinement), DLMS/COSEM provides stability in the long term, which is crucial for smart metering systems that need to stay operational for 15-20 years.

5.3 Key Components

DLMS is characterized by 3 key components: Modeling (the semantics), defined by COSEM objects, Messaging (the syntax), defined by DLMS application services, and Transport (communications), determined by communication profiles.

5.3.1 Semantics: COSEM Object Model

COSEM interface objects model the semantics of the DLMS language. Every object represents a data element (registers, clock, data profiles, schedules, calendars) and is defined by a specific set of attributes and methods: attributes represent values and methods perform operations on these attributes. Data inside a COSEM object can be retrieved by reading or writing their attributes and invoking their methods.

There are two particular object types in the COSEM object model that support security functions: Security Setup Objects, which manage the security context, and Data Protection Objects, which allow the application of cryptographic protection on COSEM data, including attribute values and method invocation and return parameters.

The Object Identification System (OBIS) is the naming system of COSEM objects. It provides readable identifiers for all data objects in a meter and is present as an attribute called Logical Name in every COSEM object. Each OBIS code identifies the application domain (like electricity, gas, water, heat energy metering), the physical quantity measured (like voltage, current, energy, power, temperature), and the way that quantity is processed and stored.

5.3.2 Syntax: Messaging

Device Language Message Specification (DLMS) is the application layer protocol that is responsible for converting the information held by COSEM objects into messages. The syntax of the language, specifically the services available for accessing objects and the formatting of messages, is defined by the DLMS services.

DLMS application layer services operate on a client-server model where the client (Head End Systems or concentrators) sends the request, and the server (end devices such as smart meters) responds. Some of the main services implemented are GET and SET (to read/write an attribute), ACTION (to invoke a method), and the unified ACCESS (which can perform all of them). In addition to request-response exchanges, DLMS supports a push operation (for example with the DataNotification service), allowing servers to send unsolicited messages to clients for events or alarm notifications.

Client and server applications communicate within Application Associations (AAs), which determine the rules for the data exchange. Each AA defines three key contexts:

- the Application Context describes how attributes and methods in COSEM object are referenced and the presence/absence of ciphering
- the Authentication Context defines the mechanism for entity authentication
- the xDLMS Context specifies the COSEM services and capabilities to be used.

An Application Association also defines the list of COSEM objects visible within that AA, attributes and methods access rights, and the cryptographic protection required for the message exchange. AAs can be either explicitly established or pre-established; it is also possible to negotiate AA context, so that data exchange parameters can be customized depending on the specific communication means.

An important feature of the AA concept is that each AA can have its own security policy, giving the possibility of selective protection. This allows to reduce computation overhead when not necessary: ciphered AAs are employed for critical data while non-ciphered AAs can be used for data that doesn't require any protection.

5.3.3 Transport: communication profiles

The transport of DLMS messages is defined by communication profiles. For each communication medium, a communication profile specifies the full protocol stack and the binding between the media-specific lower layers (PHY, MAC, Link) and the network and transport layers, with the DLMS/COSEM application layer at the top.

These are the DLMS Communication Profile Stack Specifications:

1. - 3-layer HDLC;
2. - TCP-UDP/IPv4/Ipv6;
3. - S-SFK PLC, G3-PLC, PRIME PLC, High speed PLC;
4. - EN 13757 M-Bus wired and wireless;
5. - Euridis twisted pair;
6. - Mesh network;
7. - Wi-SUN, LPWAN, LoRaWAN

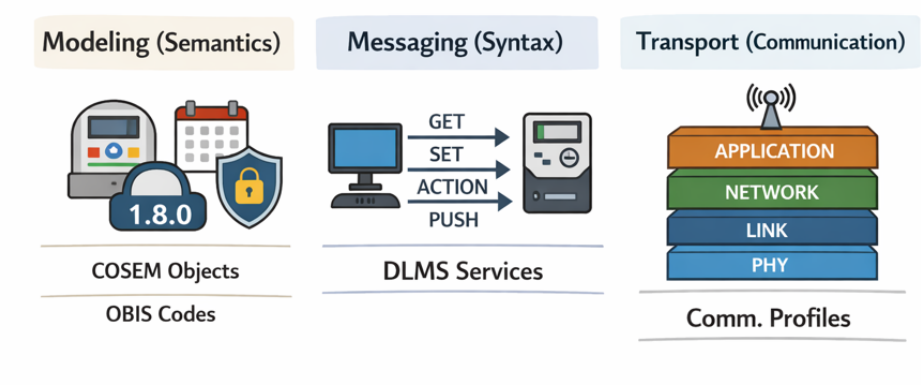

Figure 9: DLMS/COSEM architecture

5.4 Security Framework

Because smart meters serve not only as billing and consumption measurement devices, but they also manage contracts, control energy usage and provide sensitive monitoring data, their role is central in smart energy systems, and they require a high level of protection.

The DLMS security framework operates at the application level and offers end-to-end, application-to-application security: protection is applied at the source application and verified and removed only at the destination application, making it completely independent of the number of hops, the intermediate entities involved, or the communication media used. In multi-hop scenarios, common with DLMS, there are intermediate devices that need to unwrap message protection to read the routing information and determine how to forward the packet. Since messages can carry critical data, an additional layer of protection is added at the object's level, using COSEM Data Protection Object. In this way, even when the concentrator strips and re-applies the message (transport level) protection, the payload inside remains secured.

5.4.1 Security Mechanisms

The DLMS security framework is built on the three fundamental security services of Confidentiality (achieved through encryption mechanisms), Integrity (enforced through authentication and digital signature mechanisms) and Availability (through mechanisms such as firmware updates, traffic monitoring and anomaly detection, security logging and analysis)

The DLMS security framework is based on seven security mechanisms that together ensure a high level of protection:

- Entity Authentication: a challenge-response mechanism is used during the

Application Association establishment in order to authenticate the communicating parties, before exchanging any data. There exist three possible security levels: No Security (no authentication); Low-Level-Security (based on shared password), shown by research to carry implementation level vulnerabilities [27]; High-Level-Security (full mutual challenge-response authentication), which supports methods such as SHA2 and ECDSA, the latest being the most robust one with asymmetric, non-repudiable authentication.

- Role-Based Access Control (RBAC): implemented with Application Associations, it restricts access to COSEM object attributes and methods based on the client's role and defines which operations are permitted (read, write, invoke), and the respective level of cryptographic protection.
- Message Protection: ensures that COSEM object data is exchanged only through secured messages, by mapping the access rights of each object/method to the required protection type: symmetric encryption for confidentiality, authentication for integrity, or asymmetric ECDSA digital signatures for non-repudiation.
- Data Protection: to enable end-to-end confidentiality even in multi-hop scenarios, it protects COSEM data at the object level, independently of the message-level protection that carries it. This can be essential, for example, in scenarios where a third party wants to write confidential tariff data that only the meter should decrypt.
- Secure Image Transfer: it provides secure firmware upgrades on servers using the Image Transfer COSEM object, making sure that tampered images are installed only after successful verification. The firmware image is first digitally signed by the manufacturer, transferred by the client and finally checked by the server with a cryptographic integrity check.
- Communication Port Protection**:** enables port security using the Communication Port Protection object. It defends against brute force or replay attacks by setting a limit on the maximum number of connection attempts within a time window or disabling temporarily a port when suspicious traffics are detected.
- Security Logs: event records are stored in Profile Generic Objects and allow monitoring and auditing of secure message exchanges to detect anomalies and support incident response. Examples of log entries are timestamps of security errors, counts of correctly and incorrectly ciphered messages and invocation counter values.

5.4.2 Security Suites

Security suites define the set of cryptographic algorithms and allow separation of the security mechanisms from the actual algorithm selection. Currently, DLMS defines

three security suites (0, 1, and 2) plus a reserved slot for future use.

Suite	Auth. Encryption	Digital Signature	Key Agreement	Hash	Key Transport	Compression
0	AES-GCM-128	—	—	—	AES-128 Key-Wrap	—
1	AES-GCM-128	ECDSA / P-256	ECDH / P-256	SHA-256	AES-128 Key-Wrap	V.44
2	AES-GCM-256	ECDSA / P-384	ECDH / P-384	SHA-384	AES-256 Key-Wrap	V.44
Reserved	—	—	—	—	—	—

Table 3: Security Suites in DLMS/COSEM

5.4.3 Security Keys

DLMS/COSEM supports a structured taxonomy of key types, classified by their cryptographic role and lifespan. For authenticated encryption and key transport, symmetric key algorithms are used, requiring both parties to share the same key. Symmetric keys can be established by out-of-band provisioning, key transfer during AA establishment (using key wrapping) or key agreement with Diffie Hellman.

For digital signatures and key agreements, public key cryptography is used. Public keys are managed through Public Key Certificates, which bind the public key to the entity. Public key cryptography requires a Public Key Infrastructure (PKI) that includes at least a Root Certification Authority and may involve a chain of Certification Authorities.

5.4.4 Deployment gap and architectural bottlenecks

In real world contexts, there is an important gap with the security specifications: the majority of meters deployed, either because of lack of configuration or hardware constraints, relies on security suite 0, which as demonstrated by research (eFuzz from Dantas et al. [30], ValiDLMS from Mendes et al. [31]) is highly vulnerable.

Another severe issue is that many legacy profiles of the standard rely on the HDLC protocol (IEC 62056-46), which by default limits the Maximum Information Field size to only 128 bytes. While this can be only a configuration issue, many hardware constrained meters are not able to support larger payloads. For example, the Schneider Electric ION series often hardcode the Maximum Information Field to 128 bytes and lock the Transmit/Receive window size to 1 [30]. As demonstrated by Biswas et al., when utilities try to perform continuous pull requests through these default settings, the protocol overhead can create communication bottlenecks [28].

A possible solution of this issue, when caused by hardware constraints, is the integra-

tion of Hardware Security Modules (HSMs). Research (such as Huang et al. [27]) has demonstrated that offloading ECDSA, ECDH, and SHA-256 operations to dedicated processors can allow resource constrained devices to achieve a level of security in accordance with Security Suite 1.

5.4.5 Security and Efficiency

DLMS/COSEM security features add many non-negligible computational delays related to encryption, digital signature and integrity check, and adds a security header that represents an overhead to the original message. To minimize the impact of this overhead and making the security addition more efficient, DLMS support various techniques.

One of them is the presence of aggregated services that allow multiple reads and writes of attributes and methods invocation in a single exchange. Another core technique is compression, available starting from security suite 1, which can reduce message size before applying encryption, mainly used in the security header of encrypted messages. Research by Biswas et al. demonstrated that applying an adaptive multivariate data compression algorithm to metering data before DLMS transmission, can reduce the volume of data transmitted up to 85% [28]. This study also found that the DLMS protocol overhead becomes negligible at larger sizes, confirming that the protocol framing is not a limiting factor at operational scale.

Considering the context of application of DLMS/COSEM and the high latency requirements at the macro level (smart metering and demand response) pointed out by Donitzky et al. [3], this latency profile is completely justified. Features like authenticated encryption, asymmetric encryption and certificate validation, are therefore all viable in the smart metering context without disrupting its operation, since this scenario requires latency in the order of seconds to minutes.

Chapter 6

The Concept of Demand Response (DR) and Demand Flexibility (DF)

Traditionally, the paradigm adopted by power grids was "supply follows demand" where, when consumers demanded more electricity, utility companies burned more fossil fuel in centralized power plants to meet that demand. Since grids utilize renewable energy sources, which are highly volatile, this model isn't viable today both economically and physically. Therefore, the old "supply follows demand" is replaced by a "demand follows supply" paradigm called Demand Response (DR).

The U.S. Federal Energy Regulatory Commission (FERC) and the Department of Energy (DOE) define in [32] Demand Response as "changes in electric usage by end-use customers from their normal consumption patterns in response to changes in the price of electricity over time, or to incentive payments designed to induce lower electricity use at times of high wholesale market prices or when system reliability is jeopardized". In practice, this means that during peak hours (for example in hot summer when air conditioning use spikes) the grid operator can ask a building to decrease temporarily its load on the grid by switching to local battery storage or reducing HVAC and lighting usage. The advantages of this load shifting are:

- Improved grid stability, preventing blackouts
- Reduced need for peak power plants that are inefficient and highly polluting
- Cost saving for utilities (no need for expensive peak power plants) and for consumers (shifting usage to off-peak hours when prices are lower)
- It integrates well with renewable energy sources because it allows the demand to adjust dynamically depending on the current availability

- Less reliance on fossil fuels

DR originally employed event based programs where customers agreed to reduce load during specific emergencies, typically a few times a year. However, as explained by Nordman et al. [34], traditional event based DR is quickly falling out of use. As already mentioned, the intermittent nature of renewable energy sources requires continuous balancing, and static events are not suitable anymore. Instead, the industry is moving to the paradigm of "Demand Flexibility" (DF), where the interaction between buildings and grids is continuous and operates every hour of every day.

(Driven by Virtual Power Plants (VPPs)—third-party "aggregators" that bundle and manage thousands of individual customer loads to sell bulk flexibility back to the grid—and highly dynamic, continuous pricing)

6.1 Automated Demand Response (Auto-DR)

Early on, Demand Response was commonly a manual process, where grid operators would contact a facility manager, and he would manually turn off equipment through a control panel. This approach was obviously slow and unfeasible for modern smart energy systems.

According to Piette et al. at the Lawrence Berkeley National Laboratory (LBNL), auto DR consists of completely automated signaling from a utility to provide connectivity to end user control systems. The objective is to allow industrial control systems and building energy management systems (BEMS) to be programmed in advance so that demand response events can be executed without human intervention.

OpenADR (Open Automated Demand Response) is an open standard protocol that satisfies this necessity and it's one of the most used solutions.

6.2 Information Structures

Modern protocols for Demand Flexibility like OpenADR use two main structures called Events and Reports. Both of these message types are organized into time intervals, allowing them to manage real time actions and plans for future needs.

- Events (Grid to Customer): Event messages are sent downward from the utility (or smart energy management system) to end users. In traditional Demand Response, an event was just a one-time emergency signal. In the newer Demand Flexibility approach, which is largely price based, events are sent regularly and include continuous time intervals with updated information, such as the

current electricity price, the price for exporting excess solar energy back to the grid, and signals about greenhouse gas emissions to encourage more sustainable energy use. Event messages for critical grid emergencies are still used today but less frequently.

- Reports (Customer to Grid): Report messages are sent upward from customer's edge devices back to the grid operator and contain important telemetry data that gives the utility real time information on how the prosumer is using and managing energy. In systems with highly dynamic pricing, reporting back is often not required as users simply respond to price signals. On the other side, in aggregator based schemes (where an intermediate aggregator device, such as Virtual Power Plants (VPPs), collects and coordinates the energy demand of multiple users as if it were one large unit) these reports are essential to confirm that actions have taken place.

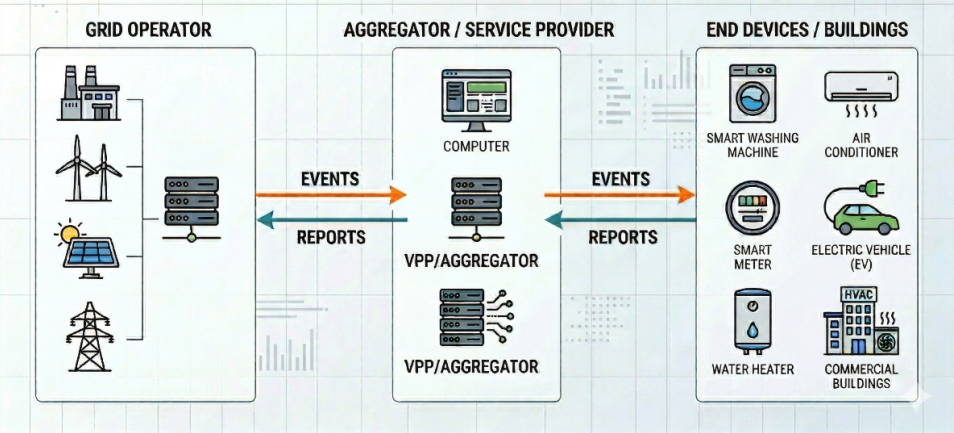

Figure 10: Demand Response Communication Flow

Chapter 7

OpenADR

7.1 General Description

Open Automated Demand Response (OpenADR) is a non-proprietary smart grid standard that enables the automation of demand response (DR) exchanges between energy providers and consumers. It provides a standardized way for the grid to broadcast pricing signals, emergency alerts or load reduction requests to building control systems, which evaluate the receiving requests and execute changes automatically. Standardization is one of the most relevant features because it allows a common way to communicate for heterogenous grids and prevent vendor lock-in,

7.2 Overview of the IT Stack

OpenADR operates at the application layer on top of conventional internet protocols (TCP/IP), allowing it to traverse fiber and cellular LTE networks. Historically, in versions 2.0a and 2.0b, the OpenADR IT stack was based on XML (Extensible Markup Language) to define data payloads. While OpenADR 2.0a served as a lite profile (only simplified event messages for constrained devices), the 2.0b version consisted of the full specification. OpenADR 2.0 relied on simple HTTP or XMPP to transport these XML payloads.

OpenADR 3.0 completely updated this IT stack to meet with modern features and constraints. Version 3.0 abandons heavy XML and employs lightweight JSON payloads and standard RESTful APIs. Nordman et al. [34] demonstrated how the payload size and complexity of OpenADR 2.0b XML structure was largely reduced in version 3.0 JSON payloads, reducing bandwidth overhead and processing overhead (which is essential for edge devices). Another novelty is the introduction of Webhooks, that allow to send push notifications via subscriptions instead of forcing clients to poll the server continuously for updates.

7.3 The Separation of Protocol and Program

OpenADR separates the communication protocol with the underlying business logic, so that a single client gateway can participate in different types of contracts using the same Event and Report message structure. The actual messages that traverse the network are just standardized Events and Reports. These messages are then encapsulated into a specific Program, that represents the actual contract/business model between utility and prosumer. For example, the OpenADR Alliance defines standard templates for various Programs, such as Critical Peak Pricing (CPP), Direct Load Control (DLC), and Distributed Energy Resource (DER) Integration.

7.4 Architecture: VTNs and VENs

OpenADR operates on a hierarchical architecture based on two types of nodes, a client and a server, called respectively Virtual End Node (VEN) and Virtual Top Node (VTN). Every entity in an OpenADR network is usually either a VTN or a VEN, but they can also be the same device. For example, a DR aggregation server can act both as a VEN for a utility DR signal, and as a VTN for end devices.

- Virtual Top Node (VTN): The VTN acts as the central server and it's typically hosted in the cloud by the utility company, a Distribution System Operator (DSO), or a third-party Demand Response aggregator. The VTN manages DR programs, calculates grid loads, generates pricing signals and event requests sent down to the edge nodes and collects reports. OpenADR 3.0 architecture also allows for the separation of complex business logic from the VTN server itself (which in this case only handles routing of OpenADR messages), making the core server much simpler to deploy and maintain.
- Virtual End Node (VEN): The VEN acts as the client at the micro-level and represents a site, building, or aggregated set of devices. It connects to the VTN, receives the DR signals, translates those signals into local physical actions, and sends telemetry data (reports) back to the VTN. Based on the scale of the energy consumers, VENs can be deployed as Cloud VENs (managing thousands of devices) or on-site VENs. This structure avoids that every single end device (like a thermostat or lightbulb) communicates directly with the grid, leaving this job to the VEN, that acts as a gateway towards the operator. The local communication between VENs and edge devices and the way events are translated into physical actions is out of the scope of OpenADR and it's handled by local BEMS protocols like BACnet.

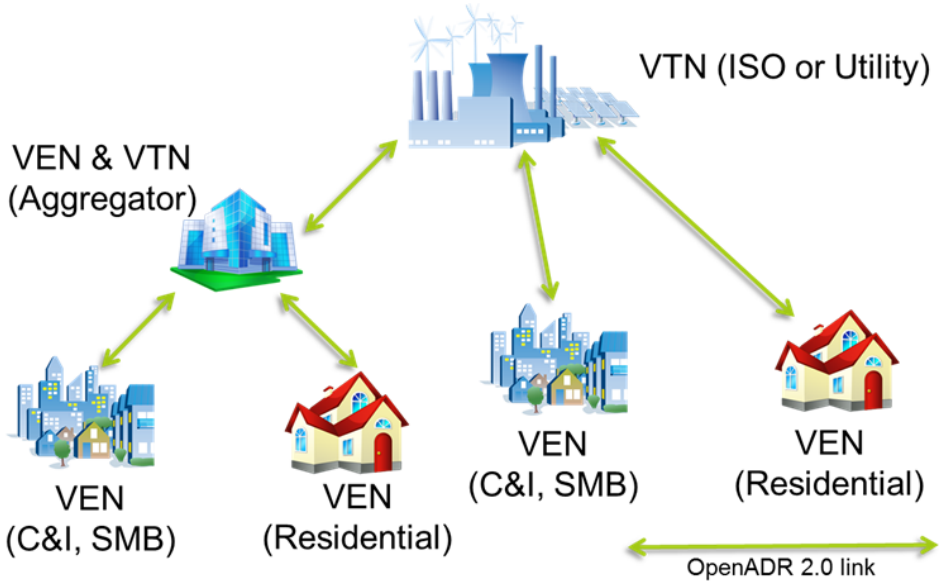

Figure 11: OpenADR architecture, from OpenADR Alliance [39]

7.5 Core Services

OpenADR is based on specific services that define the interaction between VTN and VEN. The two most important ones, that map exactly to the Events and Reports of Demand Response paradigm, are Event Service (EiEvent) and Report Service (EiReport). The protocol also includes an Opt Service (EiOpt) which provides a mechanism to communicate temporary availability or to "opt out" of a DR event if it is not desirable at a specific time.

7.6 Capacity Management in OpenADR 3.0

Because the adoption of Distributed Energy Resources introduced some physical capacity constraints at the local distribution level, OpenADR 3.0 introduces specific capacity management mechanisms (limit-based and permission based). In limit-based mechanisms grid operators can restrict how much excess solar power can be exported to the grid by a prosumer, to prevent voltage overloads. In permission-based mechanisms, specifically designed for EV-charge, a VEN must send an automated digital request to the utility to obtain additional capacity before starting an EV charge, that can be granted or denied based on real-time conditions.

7.7 OpenADR Security

Since OpenADR doesn't use a dedicated connection network but leverages public (unprotected) IP networks, many attacks like MITM, spoofing and DoS are possible. These kinds of attacks can be catastrophic: an MITM or spoofing attack could alter pricing or load-shedding signals during peak demand and cause physical damage to the grid; a DoS attack against a VTN would cause the operator to lose the ability to trigger demand flexibility events, which during a grid emergency would be forced to perform a rolling blackout to prevent a total collapse.

Security in OpenADR has evolved significantly from its early versions. OpenADR 1.0 had some basic web security but did not employ any standard cryptographic certification for authentication. OpenADR 2.0 introduced a mandatory Mutual TLS (mTLS), which requires that both the VTN and VEN prove their identities before establishing a secure tunnel, by presenting X.509 digital certificates cryptographically signed with RSA or ECC. This solution is highly secure but introduced a severe overhead to manage the Public Key Infrastructure and certificate lifecycle, mostly for constrained IoT devices.

OpenADR 3.0 version achieved instead a better trade-off between security and complexity, mantaining high protection standards while simplifying the security model. Version 3.0 replaces the heavy mutual TLS with a standard server-side TLS 1.2+ for encryption and server authentication, while for client authentication and authorization it employs Oauth 2.0 with JWT bearer tokens. This approach is more efficient, making it computationally lighter to deploy secure VENs.

7.8 Privacy and authorization limitations in OpenADR 3.0

OpenADR 3.0, with the transition to modern OAuth 2.0 and JWT bearer tokens for authentication, secures the authentication phase while preserving a good degree of efficiency. However, there are some issues related to privacy and authorization at the application level. By default, OpenADR 3.0 uses a public by default assumption, meaning that programs and events generated by VTN are visible to all authenticated VENs on that network. While it is very useful to broadcast general grid prices, it represents a privacy vulnerability in modern Demand Response events, where many events that target specific groups (like charging sequences for a localized group of electric vehicles) can be directly correlated with human behavior or physical devices, and fall therefore under the definition of Personally Identifiable Information (PII).

A compromised VEN could use its authenticated status to query the VTN and read events sent to other intended clients, creating a possible data leak. "OpenADR 3 to

Matter Interworking Reference Specification" [41] evidences this issue and proposes a solution to apply some filter server side (with minimal changes to the protocol): OpenADR 3 includes a mandatory property clientName for subscription and report objects and it can be used to filter reads of these objects. It further proposes the introduction of a read-only unique client identifier to all objects created by VEN clients. This identifier, generated by the VTN at object creation, creates an association between every data object (including events related to it) and a specific VEN client account.

7.9 Security vs. Efficiency Trade-off

In the context of security versus efficiency, OpenADR represents a protocol where security is prioritized over low latency efficiency. The cryptographic operations that OpenADR requires, both for version 2.0 (parsing XML payloads, TLS handshake and x.509 certificate validation) and 3.0 (OAuth tokens), plus the encryption overhead, introduce a delay (hundreds of milliseconds) that wouldn't be acceptable for constrained edge devices. However, this is not the case as OpenADR operates between a VTN and a VEN, with the VEN performing these cryptographic operations in the place of edge devices. Therefore, we can say that the trade-off is completely justified by the context of operation of Demand Response.

As explained by Donitzky et al. [3], DR and DER integration operations can take place under high (loose) latency constraints, accepting delays between 1 second and 15 minutes. Because the physical execution of a DR event (for example cooling a building before a peak pricing event) occurs over the span of hours, a cryptographic network delay of half a second is entirely negligible. Grid operators can apply OpenADR 3.0 without risking the operational efficiency of the grid, while remaining compatible with standard web security and IT stack connectivity,

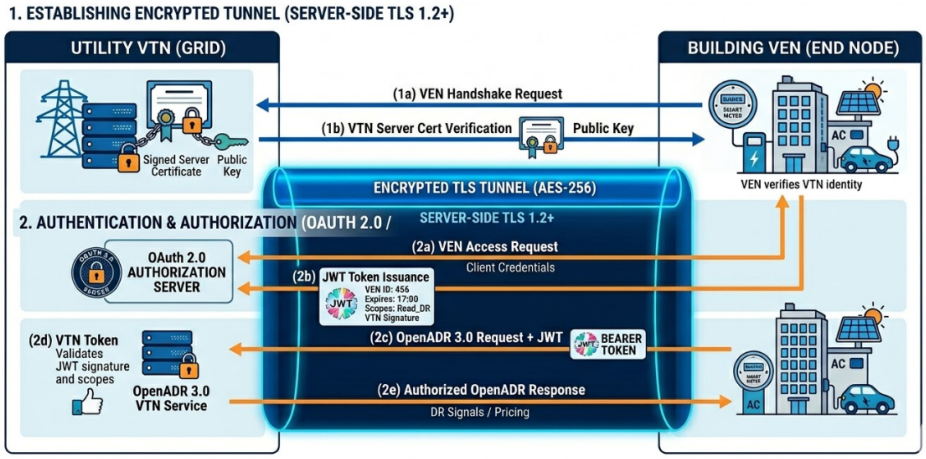

Figure 12: OpenADR 3.0 VTN-VEN security establishment schema

Chapter 8

BACnet

BACnet is a data communication protocol for building automation and control networks developed by the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). The committee's main objective was to create a protocol that would allow building systems from different manufacturers to interoperate.

8.1 Core Architecture and Interoperability

To achieve interoperability across a wide spectrum of equipment, the BACnet specification employs the concepts of:

- objects, for representing any type of building automation equipment in a standard way
- services, messages that can be sent across a computer network to monitor and control such equipment.

8.2 BACnet objects

BACnet provides a standard way of representing the functions of any device, such as analog and binary inputs and outputs, schedules, control loops, and alarms, by defining collections of related information called "objects," each of which has a set of "properties" that further characterize it. Once devices have common "appearances" on the networking terms of their objects and properties, it is then possible to define messages that can manipulate this information in a standard way.

8.3 BACnet services

BACnet defines 35 message types called "services," that are divided into 5 classes. A common one is the "ReadProperty" service request. This message causes the server machine to locate the requested property of the requested object and send its value back to the client. Other classes of services deal with alarms and events; file uploading and downloading; managing the operation of remote devices; and virtual terminal functions.

The BACnet object and services model has the advantage of being easily extensible: it allows vendors to add new properties to existing object types and create new object types (accessed in the same way as the ones defined in the standard) or new services.

8.4 BACnet Topologies

BACnet was initially designed for isolated Operational Technology (OT) networks, before the convergence of OT with IT systems. As the building automation industry embraced digitalization, it created a demand for BACnet/IP networks, which expanded communication among OT systems. The traditional BACnet topologies consisted in:

- **BACnet MS/TP:** A serial connection over RS-485 using token-passing. It is robust but limited in speed and node capacity.
- **BACnet/IP:** Created to leverage standard Ethernet infrastructure, allowing for faster communication and remote access.
- **BACnet/SC:** a new BACnet data link extension that secures BACnet/IP by leveraging the widely used and accepted TLS 1.3 security

Modern BEMS topologies usually employ BACnet MS/TP, BACnet/IP, and BACnet/SC simultaneously at different layers.

BACnet MS/TP at the edge, where simple low-power field devices (VAV boxes, actuators, wall thermostats) rely on serial cabling and lack the hardware capacity to support an IP stack (especially the cryptography required by secure protocols).

Classic BACnet/IP remains relevant on isolated VLANs where devices can utilize lightweight, connectionless UDP for rapid, real-time control. Many legacy IP controllers can't properly manage the overhead to administrate X.509 certificates required by BACnet/SC.

Finally, BAS Workstations employ BACnet/SC to communicate securely with the IT network through a secure hub.

This coexistence is possible thanks to the backward compatibility of BACnet/SC which can encapsulate MS/TP and BACnet/IP traffic.

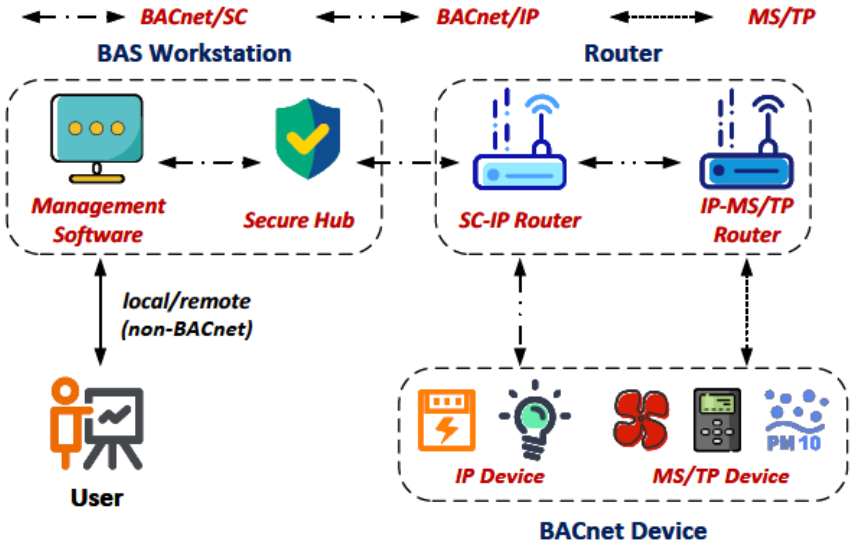

Figure 13: Example of BACnet network topology, from: [45]

8.5 Security Vulnerabilities of Classic BACnet

When BACnet was created and OT systems were isolated from IT ones, security was not a primary concern because field devices were not exposed to public networks or external threats. BACnet/IP networks, which expanded communication among OT systems, introduced new risks as IT and OT systems became increasingly blended.

The convergence of IT and OT exposed severe IT-incompatibility issues within BACnet/IP. It relies heavily on unsecure UDP ports and network broadcasts (Who-Is/I-Am messages) for device discovery. To traverse subnets, it requires BACnet Broadcast Management Devices (BBMDs), which generate excess traffic, require complex manual IT configurations, and struggle with NAT (Network Address Translation) and standard firewall rules.

Classic BACnet/IP, by its very design, lacks built-in security mechanisms. The most relevant vulnerabilities are the following:

- Plaintext Transmission: sensitive data is vulnerable to Man-in-the-Middle (MITM) attacks and sniffing because it's transmitted in clear
- Lack of Authentication: since any device in the network can issue commands without authentication, malicious actors can impersonate legitimate devices
- Broadcast Exploitation: The network is extremely susceptible to broadcast storm and denial-of-service (DoS) attacks due to its reliance on UDP broadcasts.

- No Native Auditing: no built-in logging features are in place to monitor unauthorized access.

In addition to lacking built-in security functionality, BACnet/IP networks faced issues of IT friendliness and acceptance. For example, the use of unsecure UDP ports, BACnet Broadcast Management Device (BBMD) broadcasts, and dedicated static IP addresses were not accepted by IT departments because they were incompatible with standard IT management and expectations.

8.6 Traditional security mitigations

Before BACnet/SC adoption, to mitigate the security risks, traditional OT security relied on external infrastructural defenses such as:

VLAN segregation: the OT networks that host automation devices are logically separated from the standard corporate IT networks. This limits the attack surface by isolating breaches coming from the IT segment.

VPNs (Virtual Private Networks): secure tunnels to protect the connection between remote building sites and central systems. They are useful for protecting BACnet traffic from external attackers, but they require a significant effort to configure and do not secure the actual device level payload.

Firewalls**:** they employ application specific rules to restrict traffic to specific UDP ports and prevent unauthorized external connections from reaching the internal OT network

8.7 Security transition in BACnet

BACnet Secure Connect (BACnet/SC) is an extension of native BACnet to address its main security concerns. It was developed by the ASHRAE IT Working Group in order to create a highly secure BA infrastructure, using standard internet protocol and widely used security methods. BACnet/SC is a new BACnet data link that operates at the application layer on the stack and runs over TLS to provide its security features.

The Key features of BACnet/SC are the following:

- Encryption: all the traffic is encrypted end-to-end by leveraging Secure WebSockets, an extension to HTTPS running over TLS, preventing MITM attacks
- Authentication: any device is authenticated through X.509 digital certificates before joining the network, protecting against unauthorized access.

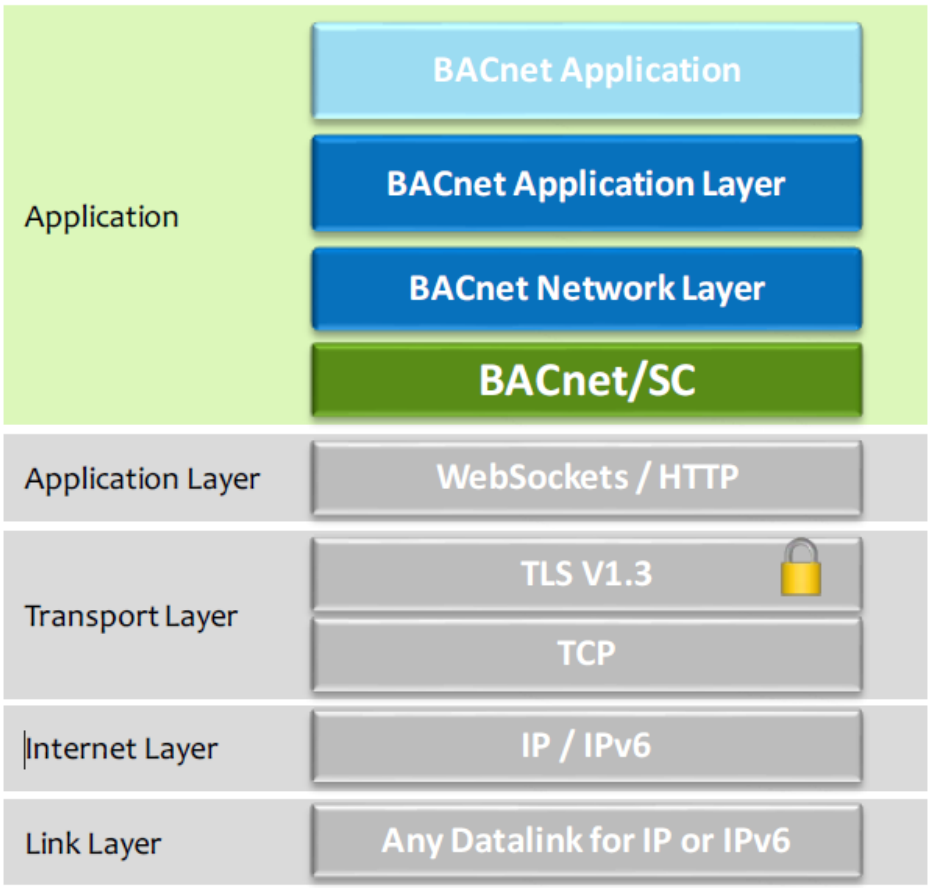

Figure 14: BACnet/SC stack

- Legacy device support: backward compatibility with older BACnet devices is guaranteed: existing BACnet IP and BACnet MS/TP networks can be encrypted into a BACnet/SC network by BACnet routing. This allows a gradual migration for organizations without disrupting operations
- IT Compatibility: BACnet/SC can traverse NATs and firewalls using outbound HTTPS connections and does not require static IP addresses or VPNs.
- Elimination of BBMDs**:** BACnet/SC does not depend on network broadcast messaging, eliminating the management and configuration of BACnet/IP Broadcast Management Devices (BBMDs)

Feature	**BACnet/IP**	**BACnet/SC**
Network Connection Type	UDP (Connectionless)	TCP (Connection-oriented)
Encryption	None	TLS v1.3 Secured WebSockets (End-to-End)
Authentication	None	X.509 Certificate-based (Mutual)
Network Traffic	Dependent on Broadcasts	Eliminates Network Broadcasts
Firewall Compatibility	Complex (Requires BBMD for subnets)	Friendly (Works through NAT/Firewalls)

Table 4: Comparison between traditional BACnet and BACnet/SC

8.8 Hub-and-Spoke Topology

While BACnet/IP was based on peer-to-peer and broadcast messages, BACnet/SC uses a hub-and-spoke topology.

One of the BACnet/SC devices is configured as a hub (primary hub) and manages the communication and data transfer of the encrypted BACnet objects. All other BACnet/SC network nodes are configured as simple nodes and communicate to the Primary Hub.

Because the hub acts as a single point of failure, BACnet/SC nodes support a failover mechanism to ensure the system remains viable even when the hub fails or is taken offline for maintenance or upgrade.

Once authenticated, nodes also have the option of communicating directly to other nodes in addition to going through a hub. This allows to lighten the load on the hub and better distribute communication.

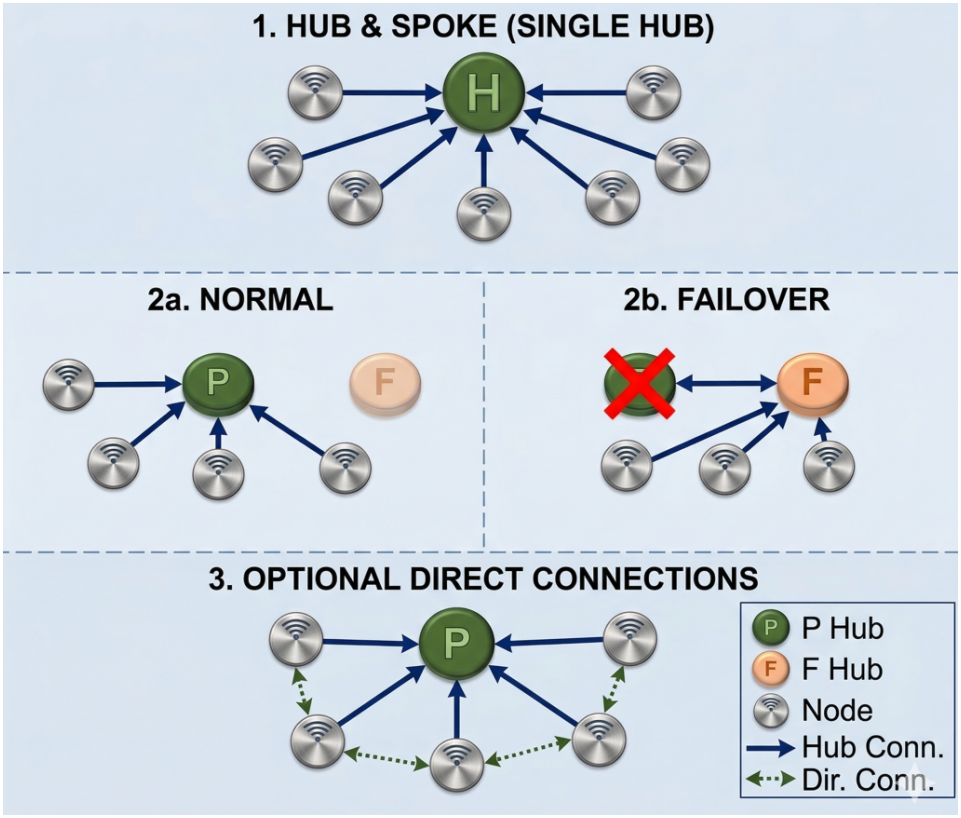

Figure 15: BACnet hub and spoke topology

8.9 Security Trade-offs

8.9.1 Baseline trade-off: security vs pure performance

The most direct comparison is with the traditional BACnet/IP without any security protection, which uses lightweight UDP packets and doesn't require cryptographic overhead. Because of this, it obviously achieves better raw performance, but it is completely impractical, at least as a standalone protocol, in the current BEMS networks with the growing cyber risks and the convergence between IT and OT.

The concern of the assessment should instead focus on comparing BACnet/SC with other alternative security solutions applied to BACnet networks: the older native BACnet/IP security model and conventional IT perimeter defenses.

8.9.2 Overhead comparison: native protocol security vs BACnet/SC

Before the adoption of BACnet/SC, it was possible to include secure protection (confidentiality and integrity) via a native BACnet security architecture, (Annex O). In this model, the BACnet devices themselves encrypted the payload and signed the messages directly at the BACnet Network Layer.

According to Nast et al's analysis [43], under this native security model, the cryptographic overhead is continuous and degrades largely system performance. In a scenario with message signing (HMAC with SHA-256) and encryption (AES-128), the network experiences a substantial increase in Round Trip Time (RTT) from 300 ms to 500 ms and the server jumps from 50 ms to approximately 150 ms just to process a single request and send a response.

In the context of BEMS, where many (often low power) edge controllers need to poll hundreds of devices, this scale of latency is not acceptable. This is probably the reason why the industry abandoned this solution.

BACnet/SC, on the other side, leverages TLS 1.3 encryption that makes per-packet transmission much more efficient. The initial session establishment is however very heavy. Catoni et al. [44] demonstrated that a TLS 1.3 handshake requires transmitting over 2 kB of certificate data and consumes approximately 2.5 ms of processing time just for certificate validation. It's important to note that this initial handshake delay affects the transmission only one time per session.

8.9.3 Operational and administrative trade-offs

It's important to note that migrating to BACnet/SC can bring some additional administrative and operational burdens such as managing and distributing X.509 certificates or upgrading hardware to handle TLS workloads, especially for organization with legacy architectures. In this context, BACnet/SC backward compatibility is great property to allow a gradual migration and support older devices and protocol.

However, as noted in the NDSS "BACnet or BADnet" paper by Zhang *et al* [45], integrating slower legacy edge networks (like MS/TP) can expose the system to local DoS vulnerabilities.

8.9.4 Conclusion: a justifiable and layered defense

BACnet/SC remains as a highly justifiable protocol choice:

- The heavy initial performance penalty is only a one-time, per-session cost incurred only when a device connects to the hub.
- Unlike legacy BACnet security models, BACnet/SC leverages TLS highly optimized symmetric encryption, ensuring that real-time operational efficiency.
- BACnet/SC provides operational advantages over traditional IT security workarounds, which required complex perimeter defenses.

Table 5: Protocol summary

Protocol	Function	Level	Latency	Security Req.	Key Security Features	Issues / Tradeoffs
IEC 61850 + IEC 62351	Substation automation, protection & control	Generation (macro)	GOOSE/SV < 3 ms; MMS 1–4 s	High	HMAC-SHA256 authentication on GOOSE/SV; TLS + X.509 on MMS; RBAC; key management (62351-9)	Encryption incompatible with < 3 ms; RSA DS too slow; HMAC lacks non-repudiation; cipher suite choice dominates MMS handshake
DNP3 / DNP3-SA	SCADA master ↔ RTU/IED monitoring & control	Distribution (macro–meso)	50–100 ms	Med–High	Challenge-response HMAC; AEAD AES-256-GCM (v6); anti-replay; optional RBAC + AMP	NACR overhead on lossy links; AGM CSQ vulnerability (fixed v6); BITW last-mile gap; TLS blocks IDS analysis
DLMS/COSEM (IEC 62056)	Smart meter data exchange (electricity, gas, water)	Distribution / Metering	Seconds–minutes	High	AES-GCM authenticated encryption; ECDSA signatures; RBAC; protection objects; secure firmware	Most meters use Suite 0 (vulnerable); HDLC 128-byte bottleneck; PKI overhead for constrained devices
BACnet / BACnet/SC	Building automation: HVAC, lighting, BEMS	End-user (micro)	100–500 ms (edge polling)	Low–High	BACnet/SC; TLS 1.3 WebSocket; X.509 mutual authentication; hub-and-spoke; backward compatibility with MS/TP and IP	Classic BACnet: plaintext, no authentication; TLS handshake cost (one-time); certificate management burden; hub SPOF; MS/TP DoS risk
OpenADR 3.0	Automated demand response signals grid ↔ buildings	Management (control layer)	1 s–15 min	High	Server TLS 1.2+; OAuth 2.0 + JWT bearer tokens; VTN/VEN gateway shields edge devices	v3.0 public-by-default exposes PII; compromised VEN can read events of others; mTLS (v2.0) heavy for IoT devices

Chapter 9

Threat Modeling Framework and Methodology

This chapter introduces the main concepts and methodologies of threat modeling, giving a theoretical base which will be later applied to the context of smart energy systems in the next chapters. The topics that will be discussed are fundamental definitions, advantages, common threat modeling approaches and methodologies

9.1 Concepts and terminology

9.1.1 Core security concepts

Before diving into threat modeling, it's essential to define the following security concepts: threat, threat action, threat agent, threat source and threat actor.

Threat: the potential cause of an incident that may harm a system, including both accidental events and malicious activities.

Threat action: an occurrence where a vulnerability is exploited through an intentional act or accidental event; it represents the realization of a threat.

Threat agent/source: The entity performing a threat action is called threat agent, often used interchangeably with threat source. Threat sources can be categorized into three areas: environmental threats (natural disasters, infrastructure failures), business resource threats (resource limitations, supply chain disruptions), and hostile actors (threat actors).

Threat actor: a threat actor is a malicious entity that carries malicious activities against a system for a defined purpose. It includes cybercriminals, insider threats, hacktivists, terrorist organizations. These actors differ largely in their capabilities,

resources and motivation. A threat actor becomes a relevant menace only when three elements coexist: intent to cause harm, capability (technical skills and tools) and opportunity (access to attack surface). Removing any one of these elements decreases significantly the likelihood of success of an attack.

9.1.2 Assets, interfaces, and vulnerabilities

A critical first step in threat analysis is documenting and determining values for an organization's assets. An asset is anything of value to the business that requires protection. Assets can be categorized into different types: hardware assets (physical computing equipment, control systems), software assets (applications, operating systems, firmware), information assets (databases, documents, intellectual property), personal/privacy assets (personally identifiable information, health records), technical assets (cryptographic keys, certificates), service assets (network services, cloud resources), and organizational/process assets (business processes, procedures, workflows).

An interface represents a point of interaction between assets or between the system and the external environment. Interfaces include network connections, APIs, user access points, physical ports, and data exchange mechanisms. The relationship between assets and interfaces is essential: assets are the targets of attacks, while interfaces are the entry points: attacks exploit interfaces to compromise assets, so the identification and protection of interfaces is important to consider.

To protect assets effectively, organizations create an asset register, that documents all security related information for each asset. This register typically includes name and description, asset type classification, location, ownership and custodianship, dependencies on other assets, and security requirements specific to that asset,

A vulnerability is, according to NIST, a "weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source." Vulnerabilities can be classified depending on the domain in: technical vulnerabilities (software bugs, misconfigurations, weak cryptography), human vulnerabilities (insufficient training, social engineering susceptibility), physical/environmental vulnerabilities (inadequate physical access controls, environmental hazards), operational vulnerabilities (process weaknesses, inadequate monitoring), and business continuity/compliance vulnerabilities (lack of disaster recovery plans, regulatory non-compliance).

The National Vulnerability Database (NVD) is a repository of known vulnerabilities maintained by the U.S. government. The NVD includes Common Vulnerabilities and Exposures (CVE) identifiers, detailed vulnerability descriptions, severity scores using the Common Vulnerability Scoring System (CVSS), impact metrics, and references to remediation guidance. CVSS constitutes a standard method to assess the severity

of a vulnerability on a scale from 0 to 10, giving organizations the possibility to prioritize its efforts for remediation.

9.2 Introduction to threat modeling

9.2.1 Definition and purpose

Threat modeling is a structured process for identifying, analyzing, and prioritizing potential threats to a system. The Threat Modeling Manifesto, developed in 2020 by a group of researchers, provides the following definition: "Threat modeling is analyzing representations of a system to highlight concerns about security and privacy characteristics" [52]. This definition emphasizes that threat modeling works with system representations (like architectural diagrams, data flow models, component specifications) instead of directly with the implemented system itself.

As described by OWASP, a threat model is essentially a structured representation of all information that affects the security of an application, a view of the application and its environment through a security glasses [52]. The process of threat modeling involves capturing and analyzing this information to make decisions about security risks in the system. Threat modeling outputs a list of all the identified security threats, prioritized, and when appropriate a list of mitigation to reduce the risks they create.

Threat modeling is based on the analysis of five key components: assets (what needs protection), architecture of the system, attack surfaces (where the system is exposed), threat actors and attack paths (how attacks can succeed). Instead of being a reactive activity, it's a proactive process where security issues are anticipated and prevented before they can happen.

Even if there are many kinds of threat events, threat modeling focuses mainly on intentional threat events, attacks by threat actors exploiting vulnerabilities to impact assets. Instead, other security processes such as risk assessment consider all types of threat events, including accidental and environmental ones. This feature is important to distinguish threat modeling from the broader program of risk management.

9.2.2 Benefits and strategic value

Threat modeling is extremely useful because it brings many strategic benefits when integrated within a system life cycle. According to OWASP, when executed properly, threat modeling provides visibility across a project that helps justify and support security efforts [52]. The threat model allows to make rational security decisions based on complete information, differently from ad-hoc decisions not supported by

data, scenarios or outcomes. This process can serve as an assurance element to justify the security posture of an application.

Fortinet identifies several core advantages [54]:

- Improved security architecture design: because threat modeling identifies architectural vulnerabilities early in the design process, before systems are built or deployed, it's possible to implement effective controls from the beginning, anticipate attacker behavior and reduce the attack surface.
- Enhanced risk management and communication: the documentation of assets and attack vectors (with respective mitigations) given by the threat modeling process facilitates the communication of risks and supports informed security investment decisions
- Prioritized remediation: teams can prioritize threats by ranking them according to likelihood and potential impact, directing the limited budget where it matters the most.
- Cross-functional collaboration: threat modeling requires collaboration from different areas of expertise: security, compliance, business and engineering. Therefore, it helps build a shared ownership of risks and improves decision-making.
- Compliance and incident readiness: well-documented threat models constitute a source of evidence for audits and certifications. Moreover, they prepare incident response teams standard detection and response techniques for the most common attacks

9.2.3 Lifecycle integration and timing

Threat modeling is not performed as a one-time event but rather as a lifecycle activity. OWASP emphasizes that it is applied in the best way when it's a continuous process during the software development, and it can be done at any resolution depending on the specific project [52]. Essentially, at different levels of abstraction the process remains the same while information becomes always more granular.

Ideally, a high-level threat model should be defined during the design phase and then refined throughout the lifecycle, since adding more details to the system means also creating new vulnerabilities that can be exploited.

Threat modeling should be performed during the following key phases:

- Design phase: To identify architectural weaknesses early before implementation begins

- Architecture review: During major changes that introduce new attack surfaces or modify existing ones
- Compliance security reviews: As part of regulatory or certification requirements
- Post-incident analysis: After security incidents to validate and update threat models

The OWASP community supports the principle: "Threat modeling: the sooner the better, but never too late" [52]. This underlines how performing threat modeling is extremely useful at any stage in the lifecycle, but it brings its maximum benefit when done at an early stage.

9.3 The four questions framework

The Threat Modeling Manifesto proposed the Four Questions Framework as the core framework to conduct threat modeling [52]. This framework, originally developed by Adam Shostack, provides a structured approach that most threat modeling methodologies answer through their technical steps. The four questions are:

9.3.1 What are we building?

This question is important to define the scope of the threat model, and it means identifying key components, defining system boundaries and documenting the architecture. Many techniques can be employed and include architecture diagrams, mapping data flows, identifying trust boundaries (where data crosses from one security domain to another) and determining critical/sensitive data.

This phase also requires gathering people from different roles with sufficient technical and risk awareness to agree on the framework to be used during the threat modeling exercise.

9.3.2 What can go wrong?

This is a “research” activity in which you want to find the main threats that apply to your application. There are many ways to approach the question, including brainstorming or using a structure to help think it through. Structures that can help include STRIDE, Kill Chains, CAPEC and others.

This is the part that involves finding the main threats appliable to the system under analysis. There are many possible approaches such as brainstorming or using

structured methodologies like MITRE ATT&CK, CAPEC (Common Attack Pattern Enumeration and Classification) or STRIDE.

9.3.3 What are we going to do about that?

In this phase findings from previous phases are transformed into specific actions. This means determining an appropriate response for each threat identified, that includes implementing a technical control, establishing or updating security procedures (monitoring, logging, incident response), transferring the risk to a third party or accepting the risk.

Remediation actions against threat are prioritized based on threat severity, likelihood, impact, and available resources, making sure that the most critical vulnerabilities receive attention first.

9.3.4 Did we do a good enough job?

Finally, it's needed to assess retrospectively the threat modeling work to check feasibility, quality and completeness. This includes reviewing if the threat model was accurate, identified threats are comprehensive and mitigations proposed are effective and practical. This phase can become a mean to improve next threat model approaches and identify lessons learned.

9.4 Threat modeling methodologies

There are many possible methodologies for threat modeling, each with a specific focus and application. There isn't a universally accepted one or a best one, the selection depends on the system under evaluation, context, available budget and security objectives.

9.4.1 STRIDE

STRIDE, developed by Microsoft in 1999, provides a scheme to categorize common threats in software systems and operating systems [53]. The acronym represents six threat categories:

- Spoofing: attacks employing falsification of identity in different forms.
- Tampering: unauthorized modification or manipulation of data, in transit or at rest, violating the principles of integrity and availability.

- Repudiation: the ability to deny having performed an action. This threat allows attackers to maintain plausible deniability.
- Information Disclosure: giving away confidential or controlled information to external or unauthorized entities.
- Denial of Service (DoS): attacks impacting the availability of a service, increasing latency or completely preventing authorized use of a resource.
- Elevation of Privilege (EoP): transforming a user account that already has access to the system into one with higher privileges, causing unauthorized access.

STRIDE is an effective methodology for threat identification during the design phase, but it is weaker for modeling real attacker behavior.

9.4.2 PASTA (Process for Attack Simulation and Threat Analysis)

PASTA is an attacker-centric methodology that includes seven different phases, with the purpose of providing a dynamic process for identifying and evaluating threats. When choosing countermeasures for protecting assets, PASTA considers business objectives other than technical and compliance requirements.

The seven phases are: (1) definition of objectives, (2) definition of technical scope, (3) application decomposition, (4) analysis of threats, (5) analysis of vulnerabilities and weaknesses, (6) attack modeling, and (7) analysis of the risk and impact on business. This approach makes sure that threat modeling aligns with business goals while maintaining technical rigor.

9.4.3 VAST (Visual, Agile, and Simple Threat Modeling)

VAST stands for Visual, Agile and Simple Threat modeling. It is a methodology based on the principle of Agile programming, an approach to software creation based on fast work cycles called sprints. This methodology integrates threat and risk management in Agile development programs on a scalable basis. VAST is founded on the principle that threat modeling is useful only when it covers the entire software development lifecycle, including three pillars: automation, integration, and collaboration. It is designed to deny the fact that threat modeling is necessarily a heavyweight activity and it can fit into Agile development processes.

9.4.4 Trike

Trike is an open-source framework focused on defending the system rather than replicating possible attacks by threat actors. A model of the application to be defended is made and then analyzed using the CRUD acronym to see who can create, read, update or delete data. This study is usually performed with data flow diagrams and examines threats that include elevations of privileges or denials of service.

9.4.5 DREAD

DREAD is a risk assessment model focused on quantifying threats and is often used together with other threat identification methodologies like STRIDE, to prioritize identified threats based on their risk. The acronym stands for five evaluation criteria: damage potential (harm caused by a negative event), reproducibility (how easy the attack can be replicated), exploitability (how easy the vulnerability can be exploited), affected users (percentage of users impacted), and discoverability (how easy can the vulnerability be located).

9.5 Threat modeling process and techniques

9.5.1 System decomposition and modeling

A critical step in threat modeling is decomposing the element or application under analysis. As Fortinet explains, decomposition allows understanding of how the application works and how it communicates with other elements in the system and external entities that could be a threat. This requires analyzing the behavior of the system across different contexts in terms of user access levels, network architectures and data types that are processed.

Threat modeling can be divided into two activities that are closely related: system modeling and threat elicitation. Because recognition is the basis for effective threat elicitation, it's very important that all threat model participants recognize the system under analysis in whatever representation is chosen (diagram, formal mode, textual description).

One commonly used type of diagram for representation is data flow diagram. Data flow diagrams represent visually how data flows into, through and out of a system and show how data is transformed at different stages. This technique eases the identification of trust boundaries, where data must be validated before entering an entity that will use it.

9.5.2 Threat identification approaches

Because many different kinds of threats exist, it's essential to identify those that are relevant to the context. There are three major approaches for threat identification, depending on the central element they revolve around:

- Asset-centric approach: starting with the evaluation of assets, then identifying threats connected to each one.
- Attacker-centric approach: identifying potential attackers and determining threats based on their objectives, putting the focus on assets that can be relevant to them.
- Software-centric approach: for organizations developing software, considering potential threats that could affect the software during its lifecycle

9.5.3 Best practices

Fortinet identifies several proven practices to strengthen security through threat modeling [54]:

- Define the scope: Limit the threat modeling exercise to a specific application, system, or process rather than the entire IT infrastructure. A well-defined and smaller scope makes the analysis manageable and focused.
- Implement Visualization: Use diagrams, such as network maps or attack trees, to represent threats and relationships visually. This approach helps identify potential risks that may be overlooked in text form.
- Use modeling frameworks: Implement a structured threat modeling framework to guide the process with targeted questions. Frameworks ensure a clear view of risks compared to unstructured discussions.
- Try attacker profiling: Build detailed profiles of potential attackers based on motives, capabilities, and methods. This helps anticipate different threat scenarios and implement the right strategies.
- Prioritize identified risks: Rank risks according to their likelihood and potential impact on the organization. This helps to address the most critical vulnerabilities first.

9.6 Threat analysis and security configuration automation

Threat analysis should not be viewed only as listing potential attacks. The real purpose of threat analysis is to identify threats for which it's possible to select and configure proper countermeasures. This is especially true in the case of modern network systems, because many countermeasures are not implemented manually anymore, but are enforced using automated methods. Network security features like firewalls, VPNs, isolation techniques, and Others can be distributed and configured automatically based on current security requirements. This means that the quality of threat analysis influences directly the quality of the countermeasures that can be applied afterward.

The importance of the automation in security configuration was discussed extensively in the literature. Bringhenti et al. provide a state-of-the-art review on the topic of automated network security where they illustrate how manual configuration of network security features is prone to error and not appropriate for large, complex and virtualized networks [58]. Other works then address firewall automation. Automation of firewall configuration in virtual networks is studied in [59], where the allocation and creation of firewall rules are solved using optimization techniques with guarantees of formal correctness. Another contribution is presented in [60], which introduces traffic flow models that support formal verification and policy refinement in network security management. Some other works extend this part of research by including sustainability as one of their goals: GreenShield is proposed in [61] to optimize firewall configuration to reduce power consuption while satisfying the required security properties, while GreenShield-E2C extends this approach to edge-to-cloud continuum scenarios in [62].

Automation has also been applied to more reactive scenarios. In [63], reconfiguration of firewalls is optimized so that an already deployed network can be updated without recomputing the whole configuration from scratch. This idea is extended in [64], where the process for mitigating threats is automated so that IDS logs are translated into new requirements and firewall configurations are recomputed to react to current Attacks. To ensure a correct configuration and reduce the risk of conflicts or redundancies, [65] proposes an approach based on atomic predicates to analyze and resolve firewall policy anomalies

This trend towards automation is not limited only to packet filtering firewalls. [66] proposes a method for the automatic configuration of VPNs with optimization objectives and formal correctness, while [67] proposes an intent-driven approach for network isolation in the Cloud, where high level security properties are translated into concrete isolation policies, such as Kubernetes Network Policies.

These contributions show that the countermeasure side of network security is becoming

always more automated. Still, there is one more interesting problem that is related to automation: automated configuration solutions usually assume that the security requirements are already known, for example which communications must be blocked, isolated, protected or reconfigured. The missing link is a threat model and an analysis algorithm that converts the knowledge about threats, assets and attack paths into security requirements. In other words, threat modeling acts as a bridge between the questions "what can go wrong?" and "what should be configured to prevent it?". This link is important in Smart Energy Systems, where cyber, physical and human elements interact. A hybrid threat model such as TAMELESS can perform the task of such mediation by generating attack paths and security states that can be used to select concrete countermeasures.

9.7 TAMELESS: A hybrid threat model for cyber-physical systems

The previous methodologies presented were developed for traditional IT systems. Cyber-Physical systems like smart grids and Building Energy Management Systems (BEMS) present some additional challenges that go beyond the scope of traditional threat modeling. These systems combine cyber, physical, and human aspects that can interact with each other and create attack surfaces that are not addressed properly by traditional threat modeling approaches.

9.7.1 The need for hybrid threat modeling

As Valenza et al. explain, current threat models and analyses typically consider only one or two components among the human, cyber, or physical aspects of the system under analysis [56]. There is a lack of approaches that consider all three components together and how they can impact each other in adding new vulnerabilities or protection measures. Because many attacks on cyber-physical systems exploit this very interaction, this gap becomes significant. An attacker can for example gain physical access to a device to compromise cyber system and then causing physical consequences. That's why these hybrid systems must be analyzed not from one perspective alone but considering physical context, cyber resources and connections and humans that operate or use it. Traditional approaches separate physical security, cybersecurity and human security into different domains and fail to capture the combination between them, that can be exploited in a single attack. This is also true from a defensive point of view: they can help protect each other and so they must be leveraged together to respond to threats.

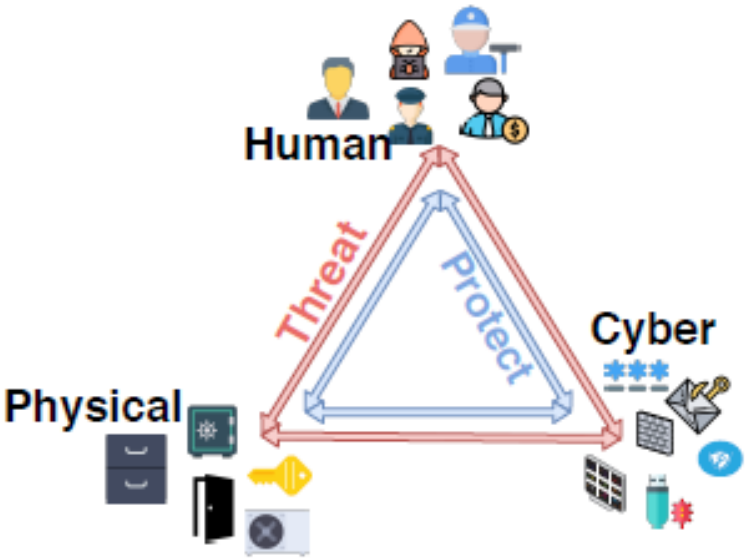

Figure 16: The main components of a hybrid system threat model, from [56]

9.7.2 TAMELESS overview

TAMELESS (Threat & Attack ModEL Smart System) is the first threat model that is able to describe and derive the security state, relations and properties of smart systems' entities while considering their cyber, physical, and human aspects simultaneously. This framework was developed by Valenza, Karafili, Steiner, and Lupu to address the challenge of hybrid threats in cyber-physical systems. The TAMELESS framework is based on three main elements:

- Hybrid Threat Model: TAMELESS introduces a new threat model that can represent the relations and security properties of system components by considering their cyber, physical, and human natures. As shown in Figure 16, the model captures how human, physical, and cyber elements can threaten or protect each other. These are not necessarily one-to-one relations, and components may be used in combination.
- Threat Analysis Method: The framework uses a set of derivation rules that allow to understand the properties of components and the overall security state. These rules automatically derive new security properties based on initial relations and properties, identifying attack paths that spread across the different aspects of the system.
- Automated Tool Implementation: TAMELESS is implemented as an automated tool using XSB Prolog, which can automatically analyze threats, verify security properties and generate attack graph representations useful for security architects to identify solutions for prevention and mitigation.

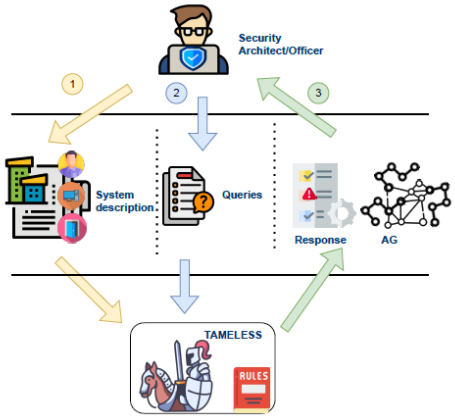

Figure 17: TAMELESS architecture and workflow, from [56]

9.7.3 System components and security properties

TAMELESS uses two component types to model systems:

1. Entities: Any system or system component that can be of cyber, physical, or human nature. Entities represent the elements that compose the system under analysis. Importantly, some entities can have more than one nature: for example, a smart lock has both physical and cyber aspects.
2. Threats: One or a sequence of actions that directly or indirectly changes a property that can alter the security state of an entity. Threats in TAMELESS can exploit vulnerabilities across the cyber, physical, and human domains, and can propagate between these domains.

TAMELESS distinguishes between basic properties, auxiliary properties, and high-level properties:

Basic properties:

- Compromised (Comp): Entity has been compromised by a threat
- Malfunctioned (Malfun): Entity is malfunctioning and one or more functionalities are not working as expected
- Vulnerable (Vul): Entity has a known vulnerability that makes it vulnerable to a specific threat

Auxiliary properties:

- Detected (Det): It has been detected that an entity has been compromised by a threat

- Restored (Rest): Control over an entity is restored, usually after compromise
- Fixed (Fix): The functionality of an entity is repaired, usually after malfunction

The framework distinguishes between assumed properties (declared explicitly as part of security assumptions) and derived properties (obtained by applying derivation rules to known properties). Assumed properties represent initial knowledge, while derived properties represent security properties that can become true when attackers exploit vulnerabilities.

High-Level properties: TAMELESS also defines high-level properties so that users can understand the security state of a system more easily:

- Valid (Val): Entity has not been compromised and is not malfunctioning
- Defended (Def): Entity is defended from a threat when another entity protects it, and that protecting entity is valid
- Safe (Safe): Entity is not vulnerable to a threat or can be defended from it
- Monitored (Mon): Entity is monitored for a threat by another valid entity
- Checked (Che): Entity's functionality is checked by another valid entity
- Replicated (Rep): At least one valid entity replicates this entity

9.7.4 Relations between components

TAMELESS represents both structural relations between entities and relations between entities and threats:

Structural relations between entities:

- Contain: Represents system composition (e.g., a room contains a server)
- Control: One entity controls another (e.g., a person controls their identification badge)
- Connect: One entity connects two others (e.g., a door connects a room with a hall)
- Depend: One entity's functionality depends on another
- Check: One entity checks that another is functioning normally
- Replicate: One entity is a replica of another

Relations between entities and threats:

- Protect: An entity protects another from a specific threat
- Monitor: An entity monitors another for a specific threat (enabling detection even when prevention is not possible)
- Spread: An entity can propagate a threat
- Potentially Vulnerable: An entity can become vulnerable to a threat under certain circumstances

These relations allow TAMELESS to represent how attacks propagate across the different aspects of a system and how different aspects can protect each other.

9.7.5 Derivation rules and threat analysis

The threat analysis performed by TAMELESS operates through a set of derivation rules that infer automatically new security properties from known relations and properties. Key derivation rules include:

- Compromise Propagation Rules: Rules that determine when an entity can be compromised through control relationships, connections, or containment. For example, if entity A controls entity B, A can be compromised, and A spreads a threat T, and B is not safe from T, then B can be compromised by T.
- Malfunction Propagation Rules: Rules that determine when compromise leads to malfunction, or when malfunction propagates through dependency relationships.
- Vulnerability Derivation Rules: Rules that determine when malfunctioning entities become vulnerable to specific threats.
- Detection and Restoration Rules: Rules that determine when compromises can be detected (based on monitoring relationships) and when entities can be restored (based on replication and detection).

The application of these rules creates logical attack graphs that show not only the possible attack paths but also the application of specific derivation rules, making the threat analysis process transparent and auditable.

9.7.6 Attack graph generation

TAMELESS performs an automatic generation of attack graphs that represent:

- Assumed and derived properties of entities

- Relationships between entities (shown as dashed edges)
- Connections created by application of derivation rules (shown as solid edges)

These attack graphs are a powerful tool for representing visually how attackers can exploit vulnerabilities across human, physical and cyber aspects of a system. Thanks to that, they enable to analyze a priori possible attack paths and support informed decision making for countermeasures.

9.7.7 TAMELESS workflow

The TAMELESS workflow, as implemented in the tool, follows these steps:

1. System Description Input: Users specify the system components (entities and threats), relations between components, and security assumptions.
2. Automated Threat Analysis: TAMELESS applies derivation rules to compute basic and auxiliary properties (compromised, malfunctioning, vulnerable, detected, restored, fixed) for all entities.
3. Query and Response: Users can query TAMELESS to display predefined security properties of system components. This allows to perform consistency checks between expected and derived states.
4. Graphical Output: TAMELESS generates attack graphs showing threat propagation paths and the derivation process.
5. Iterative Refinement: after adding new components or countermeasures, the analysis can be repeated again to observe how the threat model changes. This iteration can be repeated until the threat model is considered acceptable.

9.8 Recent extensions to TAMELESS

In [57] Sanfilippo extended TAMELESS to make it "more exhaustive, efficient, and accessible" through formal verification, technological modernization, and integration with large language models. These extensions address some of the limitations of the original framework and can be differentiated in four areas: formal verification of the threat model rules, technological infrastructure upgrades, development of a graphical user interface, and integration with machine learning techniques.

9.8.1 Formal verification with PRISM

Properties and relations in the original TAMELESS threat model allow to express derivation rules, but these rules were not formally verified and were abstracted from concrete experiences instead of formal axioms. As [57] notes, the system lacked the demonstration that it is sound, meaning that every property that can be derived from its rules is actually true with respect to the semantics of the model. This informal interpretation of relations may create ambiguities that could affect the reliability of the results of threat analysis.

To address these issues, TAMELESS derivation rules were formally modeled with PRISM, a model checker for formal modeling and analysis of systems with probabilistic behavior. PRISM performs a state exploration to verify the correctness and consistency of the derivation system. The verification process modeled TAMELESS rules as a Discrete-Time Markov Chain (DTMC), where each state represents a configuration of properties and relationships, and transitions represent the application of derivation rules. The model has local variables that represent all properties and relationships with initial values chosen to avoid triggering any rules, then explores all possible state transitions systematically.

9.8.2 Verification results

The formal verification process validated several critical properties of the TAMELESS rule system:

- Rule Activation: Each rule was verified to activate correctly without creating logical conflicts with other rules.
- Consistency and Resilience: The verification confirmed that restored or fixed entities cannot be immediately compromised again, ensuring system liveness and resilience.
- Multiple Compromise-Restoration Cycles: The system was verified to handle multiple cycles of compromise and restoration, confirming nonzero probability that a compromised system can be restored and subsequently compromised again.
- Relationship Analysis: The verification process revealed that all TAMELESS relationships are applicable independently of the nature of the entities involved, with one exception: the "Contain" relationship is not applicable to the human domain.

The formal verification demonstrates that TAMELESS is robust, consistent, and capable of handling critical state transitions without deadlocks or logical inconsistencies,

significantly strengthening confidence in its application to real-world systems.

9.8.3 Neo4j integration

The original implementation of TAMELESS relied on an older version of Neo4j graph database that used REST API endpoints for CRUD operations. Sanfilippo updated it to support versions 4.x and later of Neo4j, which include:

- Authentication: simple authentication implemented with HTTP Basic Authentication (credentials sent in Base64 encoding inside the Authorization header of requests).
- Unified Transaction Endpoint: transition from multiple specialized endpoints (/node, /relationships, /schema) to a single endpoint called /tx/commit, which runs Cypher queries in atomic transactions and performs automatically commits and rollbacks
- Improved Transaction Management: This new architecture takes care of transaction atomicity automatically, so there is no need for manual transaction management and rollback management like in previous versions.

The changes applied make sure that TAMELESS is compatible with current Neo4j versions and improve the efficiency and security of database communication.

9.8.4 Graphical User Interface development

Another essential addition was the addition of a GUI based on React and Tailwind CSS frameworks. Previously, TAMELESS could be accessed only through API, with results returned in JSON format, resulting in a usability barrier especially for non-technical users.

The interface comprises two main components:

GraphEditor component: This component allows intuitive modeling of the system via:

- Nodes being inserted through forms with labels and types selected (Human, Cyber, Physical)
- Edge created with selection of source and target node from existing nodes
- Test Case Library that comes pre-loaded to allow visualization of known attacks
- Graph Visualization using the Cytoscape library to arrange nodes dynamically and automatically

- Click-to-edit functionality to modify relationships and security properties
- Visual representation where blue nodes represent entities, edges display relationship labels, and red nodes indicate security properties

GraphResults component: This component provides full visualization of the results:

- Initial display of TAMELESS formal analysis results with coloring representing the security level of each node
- Option to integrate MITRE ATT&CK results with LLM analysis
- Graph reduction showing only entities present in both result sets
- Display of quantitative metrics like graph reduction percentage, nodes added by ATT&CK analysis, and entity identification
- Node-level details including compromise probability, MITRE ID of each technique identified, exploit probability, official MITRE description, rationale for exploitability, detection methods used, link to MITRE documentation, and mitigation methods

9.8.5 LLM integration

One of the key extensions to TAMELESS in [57] is the integration of Large Language Model analysis using LLaMA 3.1 with Retrieval-Augmented Generation (RAG), which provides threat intelligence from the MITRE ATT&CK database. This addition solves an important limitation of the original TAMELESS: it could determine if a node may be compromised, but it could not identify the specific attack techniques.

The workflow is the following:

1. Parallel Analysis Initiation: When analysis begins, the orchestration program (graph_processor.py) sends the system graph to both the TAMELESS analyzer for formal analysis and to the LLM module for analysis of threats.
2. Per-Node LLM Analysis: For each node in the system graph:

- Generate a partial context string of all the elements and security properties related to that node
- Create a RAG context containing the node's security properties
- Send this information to the LLM "identify" endpoint

1. RAG-based Threat Identification Enhancement: The LLM module:

- Uses FAISS (Facebook AI Similarity Search) index created from MITRE ATT&CK dataset
- Uses Sentence Transformer model (all-MiniLM-L6-v2) for semantic embedding of tactics, techniques, descriptions, and MITRE IDs
- Relevant attacks are found using a semantic search based on node properties.
- Sends node information, system context, and retrieved ATT&CK techniques to LLaMA 3.1

1. Threat Validation: For every technique detected:

- Context about the whole system, covering all nodes, edges, and properties, is sent to a "validate" endpoint
- The LLM determines if the technique is actually exploitable given the complete system configuration
- The LLM provides reasoning explaining why the technique applies or does not apply

1. Result Integration: Results from both TAMELESS formal analysis and LLM analysis are merged, and users have the option to view formal results alone or combined results with a reduced graph that shows only confirmed threat paths.

The following optimizations were made to make LLM analysis computationally feasible:

- Model Quantization: replacing typical 16 or 32-bits quantization with a 4-bit quantization using BitsAndBytesConfig library, reducing memory requirements while maintaining acceptable accuracy with specialized quantization algorithms.
- Asynchronous Processing: LLM execution runs in separate threads to avoid blocking HTTP connections when performing long analysis operations.
- Efficient Embedding: Semantic embedding of MITRE ATT&CK dataset allows for fast similarity searches instead of exhaustive comparisons.

9.8.6 Probability and risk calculation

9.8.6.1 Node compromise probability

The extended TAMELESS framework calculates quantitative probabilities of node compromise based on internal factors from formal analysis. The probability is maximum (1.0) if a node is confirmed to be compromised with the formal analysis.

For other security conditions, probability is calculated using weighted factors:

- Vulnerability State (Vuln): 1 if node is vulnerable, 0.6 if is malfunctioning, 0 otherwise
- Compromised Connected Nodes (CompConn): number of connected nodes in compromised state.
- Threat Spread Capability (ThreatSpread): number of adjacent nodes capable of spreading threats.
- Control and Dependency Relations (RelControl, RelDepend): number of adjacent nodes that control the target node or depend on it, and are compromised
- Protection Mechanisms (Protection): number of nodes providing protection to the entity.

The overall formula for probability of compromise is:

$$P_{\text{compromise}} = w_1 \cdot \text{Vuln} + w_2 \cdot \text{CompConn} + w_3 \cdot \text{ThreatSpread} + w_4 \cdot \text{RelControl} + w_5 \cdot \text{RelDepend} - w_6 \cdot \text{Prot}$$

where $\sum_i w_i \leq 1$ (weights sum to at most 1.0)

9.8.6.2 Attack technique probability

For every attack technique that the LLM identified as exploitable is assigned a probability that considers topological factors and specific indicators from MITRE ATT&CK:

- Permission Requirements (Extracted from "x_mitre_permissions_required" field): Value = 0.4 for "Root" or "Administrator" privileges, 0.6 for "User" privileges and 0.8 if no specific privileges are required.
- System Prerequisites (SystemReq): Value = 0.8 if prerequisite list is empty, 0.6 if ≤ 2 prerequisites required, 0.3 otherwise.
- Kill Chain Complexity (KillChainScore): Value = 0.5 for complex stages (Privilege Escalation, Persistence, Defense Evasion), 0.8 for other stages.

The formula for the attack technique probability is:

$$P_{\text{Technique}} = w_1 \cdot \text{Vuln} + w_2 \cdot \text{CompConn} + w_3 \cdot \text{ThreatSpread} + w_4 \cdot \text{RelControl} + w_5 \cdot \text{RelDepend} - w_6 \cdot \text{Protec}$$

where $\sum_i w_i \leq 1$

9.8.6.3 Risk assessment

The Risk associated with each identified technique is calculated quantitatively by taking into account the impact that would result from the event and the likelihood of the event occurring, following NIST guidelines. The component for likelihood is the $P_{\text{Technique}}$ previously calculated, the probability that the technique can be successfully exploited. The component for the impact metric is determined by multiple factors that influence how critical a node is: structural importance (calculated from count of entities controlled by or depending on this node and count of direct connections to other nodes), security property criticality (indicating if node compromise affects confidentiality, integrity and availability) and cascading impact (evaluates cascading effects based on dependency chains and control relationships).

The quantitative risk metric allows to prioritize mitigation measures by directing resources to techniques with the highest combination of likelihood and impact.

9.9 Proactive vs reactive approaches

Security considerations must happen from the early stages of system design and during the whole lifecycle. This principle is at the base of Microsoft's SD3+C, which stands for the strategies Secure by Design, Secure by Default, Secure in Deployment and Communication. As anticipated, threat modeling is a proactive process. However, only a proactive approach is not sufficient, because not all threats can be anticipated at design phase. This means that it's necessary to combine reactive approaches to address unexpected problems. Reactive strategies are in fact the basis for penetration testing, fuzzing, code review, that are able to discover threats and further improve the security of a product.

OWASP underlines that after security incidents, reviewing and updating threat models is an important process [52]. This retrospective threat modeling helps organizations understand how attacks succeeded, identify gaps in the original threat model, and improve future threat models based on data coming from real world attacks.

Therefore, proactive threat modeling during design and development should also combine reactive validation (testing, retrospective analysis) to be as effective as possible and address the entire lifecycle continuously.

9.10 Application of TAMELESS to smart energy systems

The first methodologies discussed (STRIDE, PASTA, VAST, Trike, and others) were developed mainly for traditional information technology systems where the primary objectives were data confidentiality, integrity, and availability. However, smart energy infrastructures present additional complexities compared to the conventional IT threat modeling. Smart grids and Building Energy Management Systems (BEMS) combine cyber systems (control networks, protocols) with physical infrastructure (power lines, HVAC systems) and human operators, meaning that hybrid threat modeling becomes essential. Traditional threat models that focus only on cyber or physical security cannot represent adequately the attack surfaces and cascading vulnerabilities present in these systems.

TAMELESS is particularly relevant to smart energy infrastructures because it aligns with the inherent characteristics of these systems. The main challenges specific to smart energy system that the framework addresses are:

- Critical safety operations: The capability of TAMELESS to simulate malfunction propagation is especially beneficial when dealing with energy infrastructures, in which availability prevails over confidentiality, and where cascading failures can have severe consequences for public safety and infrastructure resilience.
- Formal verification confidence: the formal verification of derivation rules, based on PRISM, gives mathematical proof that derived security properties are sound. This is essential for safety-critical energy systems where an incorrect threat model could cause serious consequences like service disruptions or physical harm.
- Accessibility for diverse stakeholders: the GUI makes it easier for specialists like grid operators, energy engineers and facility managers, that often have high expertise in power systems or automation but limited cybersecurity knowledge, to understand the threat modeling process and participate in security decision making.
- MITRE ATT&CK integration: The MITRE framework includes specific matrices for Industrial Control Systems (ICS), providing threat intelligence directly relevant to SCADA systems, smart meters, building controllers, and other energy infrastructure components. In this way TAMELESS is connected to a continuously updated repository for threat
- Quantitative risk metrics: risk calculations allow to prioritize security investments, which is extremely important for energy utilities that must balance security/operational/regulatory requirements and costs.

- Graph reduction for complex systems: smart grids and BEMs can contain hundreds to thousands of connected components, including meters, sensors, controllers, gateways, servers and user interfaces. The capability of automatic graph reduction can draw the attention of security professionals only on relevant attack paths instead of overwhelming them with the complexity of the full system.

Chapter 10

Methodology and setup

This chapter will outline the practical TAMELESS setup, specifying the core integrations and modifications (from the extended version by Sanfilippo in [57]), to adapt the tool for the purpose of this thesis. Next, it will present the methodology used to compare SESs protocols and their tradeoffs employing this tool. Finally, results from analysis performed on defined use cases will be shown and evaluated, linking everything to the comparative protocol analysis performed on the previous chapters.

10.1 Limitation of the original codebase

The original codebase depended on the infrastructure of the University of Southampton. In [57], Sanfilippo designed the AI component to run a localized instance of the model Llama 3.1 8B-Instruct.

Running an LLM locally can be computationally heavy. In fact, [57] applied 4-bit quantization to the model to make the process lighter and avoid saturating computational resources; the system also relied on the IRIDIS High Performance Computing Facility. However, as he explains in section 8.2 of his thesis, the IridisX architecture had the limitation that the computing nodes were isolated and prevented from downloading directly larger and more capable models, forcing the model to be small enough.

Therefore, the framework code was based specifically on the IRIDIS HPC cluster and it contained hardcoded directory paths pointing to specific user folders on the Iridis supercomputer, meaning that the original code (related to the MITRE ATT&CK LLM analysis) could not be run on a standard local machine. Since the LLM could not be executed locally without the HPC cluster, the possibility of performing a dynamic analysis was disabled for local users.

- The multi-threading logic responsible for triggering the live LLM analysis was

commented out.

- Instead of evaluating new graphs dynamically, the backend was hardcoded to avoid the AI analysis. The author acknowledges this limitation directly in the project notes: ″Due to the absence of the LLaMA analysis, please test only with the provided examples that have available results. With any other graph, the test will fail...″. The system simply loaded JSON files precomputed on the Iridis HPC cluster (like results_20250203_203926_light.json) that contained the LLM analysis, and sent it to the frontend.
- Users could draw new topologies and compute the formal TAMELESS logic, but the MITRE ATT&CK could not be computed, unless one of the test use cases was used, for which the precomputed JSONs results where available.

10.2 Architectural enhancements & cloud integration

To allow the live LLM analysis on any machine, the dependency on the local LLM was replaced with cloud-based Groq APIs. Instead of relying on the model in local memory, llama_vulnerability_analyzer.py now acts as a lightweight API client. The orchestrator now integrates Groq Python library and sends prompts over the internet to Groq's LPUs (Language Processing Units), achieving the same results in real time without local hardware constraints.

Since free Groq's APIs have some limitations on max tokens per day and tokens per minute, submitting the entire graph of the system under analysis can go past these free-tier rate limits. To address the rate limit issue and avoid the orchestrator from crashing when exceeding the Tokens Per Minute speed limit, the API calls were executed in a try/except block where they would be repeated after a sleep timer in case of an HTTP 429 response (too many requests). The exponential backoff timer makes the python script pause to allow the token bucket to refill before resuming the analysis of the graph without dropping node data.

10.3 TAMELESS architecture & components

The system extended by [57] is a modular, three-tier web application.

1. The Frontend (React GUI)

Main elements:

- GraphEditor.jsx: The input interface where the security architect can define the system. Users can add nodes (Human, Cyber, Physical), create edges (rela-

tionships like Control, Protect, Connect), and assign initial security properties like Compromised, Vulnerable.

- GraphResults.jsx: The output interface powered by Cytoscape. It shows the final attack graph with colored node states and displays specific MITRE ATT&CK techniques, risk scores, and mitigation strategies when a node is clicked.

2. The Backend Orchestrator (Python/Flask)

- graph_processor.py: when an analysis starts, this server receives the graph JSON and splits the workload, triggering the TAMELESS (Java) and LLaMA (Python) analyses to run in parallel. It also merges the results at the end.
- The Neo4j database is used immediately after the Python backend receives the JSON. The Python backend doesn't analyze the JSON itself; instead, it uses the TAMELESS API to write that JSON data directly into Neo4j.

3. The Formal Logic Engine (TAMELESS Core)

- Neo4j Database: it stores the complex graph of nodes, properties, and relationships. It uses Cypher queries to efficiently filter and retrieve the network topology. After each analysis, a delete all endpoint in the Java backend is used to execute a delete call and removes all nodes and relationships that are currently in the graph, preparing it for the next analysis.
- Java Bridge (Neo4jDB.java & MyNeo4jClient.java): Acts as the Data Access Object (DAO) and orchestrator for the formal analysis. It extracts the relevant graph from Neo4j and translates it into a .pl file that Prolog can read.
- SWI-Prolog Engine: It applies derivation rules (for example, Rule 6: If A controls B, and A is compromised, B is compromised) to mathematically prove the security state of every node in the network.

4. The Probabilistic Intelligence Engine (LLaMA 3.1 + RAG)

- llama_vulnerability_analyzer.py: a secondary Flask server acting as a lightweight API client that queries the GroqCloud API (using the LLaMA 3.1 8B model) to serve as a virtual cybersecurity expert.
- Retrieval-Augmented Generation (RAG) Module: uses a FAISS vector database to search the official MITRE ATT&CK dataset. It injects real-world threat data (tactics, techniques, mitigations) into the LLM's prompt.

- /identify Endpoint: looks at a node's type and context, then asks the LLM to find potential MITRE ATT&CK vulnerabilities.
- /validate Endpoint: takes the found vulnerabilities and asks the LLM to double-check them against the entire system graph to ensure they are actually logically exploitable.

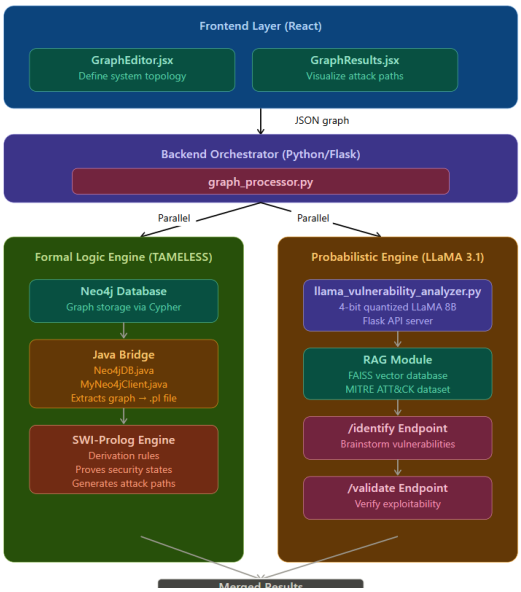

Figure 18: Overview of TAMELESS architecture and components

10.4 Workflow representation diagram

1. User creates a graph (nodes, edges, properties) in the React frontend
2. Python backend translates JSON to XML API calls
3. Java backend writes entities/threats to Neo4j database
4. Cypher queries extract graph topology
5. Java builds Prolog rules file (.pl)
6. SWI-Prolog executes derivation rules to prove attack paths
7. Parallel split: Formal results + LLaMA analysis begin simultaneously
8. LLaMA identifies MITRE ATT&CK vulnerabilities using RAG
9. LLaMA validates exploitability against graph context
10. Probabilistic results include tactics, techniques, and mitigations.
11. Both analyses merge into final interactive visualization

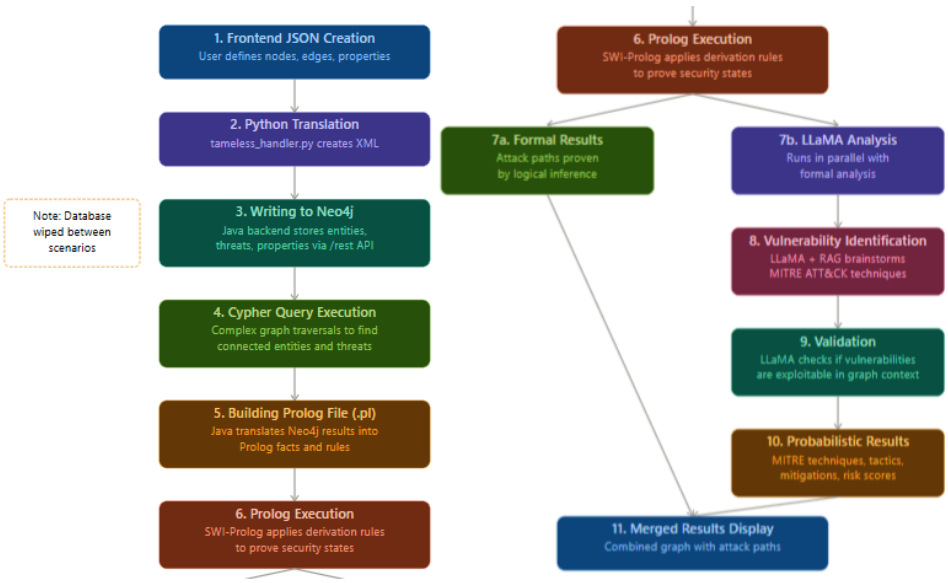

Figure 19: TAMELESS workflow diagram

10.5 Methodology for SES protocol analysis

TAMELESS analysis, thanks to its integration of cyber, physical and human factors, adapts perfectly in the context of Smart Energy Systems. The methodology used in this thesis will start with the mapping of the SES ecosystem to TAMELESS topology definition: facility managers, operators and attackers are modelled human nodes, physical equipment such as HVAC, smart meters and sensors are modeled as physical devices and all the remaining digital devices or entities, like protocols, gateways, servers, are modeled using cypher nodes. For each node, the specific properties are included (like vulnerable, compromised) and the relations between them, according to TAMELESS relations between components (connect, depend, contain, control...).

After the specific scenario is mapped into the TAMELESS topology, both the formal analysis by Prolog and the LLM analysis for MITRE ATT&CK are used to define possible attack paths. The core stage of the methodology is performing a comparison between the insecure and secure version of a system, where the supposed security state depends on which specific communication protocols is present between SESs entities (e.g. Bacnet/IP vs Bacnet/SC in a smart building).

First, a baseline graph is drawn representing a generic SES using an insecure, plaintext protocol, and it will run through TAMELESS and LLaMA analysis to generate an attack graph. This will clearly demonstrate how easily the attacker can compromise the unsecured system when no secure protocol is used to protect the communication, and then cause physical damage (like manipulating a HVAC system). Next, the same graph will be drawn, but this time adding secure versions of the protocols present. As TAMELESS requires, a security feature of the secure protocol version is

modeled as a distinct cyber node. This node does not represent a physical device, but the cryptographic or logical mechanism itself. It is then connected to the vulnerable node via a protect edge, which specifies the exact threat it mitigates. For example, if node A is vulnerable to networkEavesdropping, a security node is introduced representing the encryption mechanism (such as tls_encryption_tunnel). The tls_encryption_tunnel has a protect relationship targeting node A, specifically mitigating networkEavesdropping.

It's expected that the formal TAMELESS logic will show the attack path breaking, and LLaMA will confirm that the previous MITRE techniques are not exploitable anymore. In this way the final metrics of the unsecured and secured versions will be compared, and it will be clear how a secure protocol is able to protect a SES, or a completely unsecured infrastructure is vulnerable to all kinds of attacks shown in the tool attack graph output.

Other than a completely unsecure and a fully secure scenario, middle-way scenarios will be used as it is useful to show the tradeoffs of securing only a certain path of the communication (or integrating only some optional security properties) because of performance or timing requirements, or legacy infrastructure.

10.6 Scope and limitations of the evaluation

The evaluation performed is focused on the determination of the attack graph, specifically how an attacker can compromise the system by following a certain path, exploiting vulnerabilities of nodes in the SES infrastructure. However, while TAMELESS can prove security vulnerabilities, it has no awareness of other important metrics like latency, computational overhead or battery usage. Many of these metrics were evaluated separately in the previous protocol analysis chapter, highlighting the implications of security when applied in scenarios which require low latency or overhead. This tool can serve as a threat modeling approach to help a security architect in identifying the best solution to protect against certain attack paths, while separately taking into account the other tradeoffs of each secure protocol addition.

10.7 Use cases description

Since the main communication protocols employed in Smart Energy Systems operate at different levels, the use cases employed can be divided into distinct attack domains, so that the specific vulnerabilities of each protocol can be highlighted. Lastly, a comprehensive scenario will be analyzed, including all the different levels and protocols of a SES.

1. The micro level (building automation)

- Protocols: BACnet (Unsecured) vs. BACnet/SC (Secure Connect).
- Scenario: An attacker gains access to a corporate network and attempts to manipulate the HVAC system or lighting to cause physical damage or a localized blackout.
- Expected outcome: traditional BACnet (which lacks native encryption and authentication) allows lateral movement and spoofing, while the secured version halts the attack path.

2. The edge / prosumer level (smart metering & Demand Response)

- Protocols: DLMS/COSEM (Meters) and OpenADR (Demand Response).
- Scenario: A localized attack where a threat actor compromises a smart meter or intercepts OpenADR pricing signals to a smart home/factory, tricking the system into drawing massive power during peak hours.
- Expected outcome: the unsecured version allows the attacker to spoof Demand Response signals or perform false data Injection on the meter. The secured alternative (TLS for OpenADR and High-Level Security with cryptographic authentication for DLMS/COSEM) nullifies the spoofed commands, protecting the physical load.

3. The macro level (substation & transmission)

- Protocols: DNP3 and IEC 61850.
- Scenario: An attacker sends malicious DNP3 commands to the outstation (altering voltage sensor readings) and malicious MMS commands to the IED (triggering a spoofed GOOSE message to trip the breaker)
- Expected outcome: while the unsecured unencrypted version permits spoofing and the secured alternative, with DNP3-SA and IEC 62351, blocks the attack path by ensuring entity and message authentication and message integrity.

4. Complete SES example infrastructure

- Protocols: all the previous protocols from micro, macro and edge level.
- Scenario: attackers breach the corporate IT network at the Micro level and access an unsecured BACnet system. From there, they laterally move to the Edge level compromising the OpenADR gateway and the DLMS/COSEM smart meter. Then, they use the compromised meter to inject false data to the macro

level. The SCADA server (running unsecured DNP3), blindly trusts this data and sends wrong trip command to IEC 61850 IED, shutting down the physical circuit breaker and causing a blackout.

- Expected outcome: shows how a breach in the Micro level can cause a cascading failure that extends up to the transmission and macro level, potentially causing critical consequences on a large scale

10.8 How to interpret TAMELESS attack graphs

In the following results evaluations, the attack graphs generated automatically by TAMELESS will be presented. It's therefore necessary to define how to visually interpret them in a proper way. The final visualizations produced by TAMELESS after the Prolog analysis, distinguish different node states using a color-based taxonomy, according to the following legend:

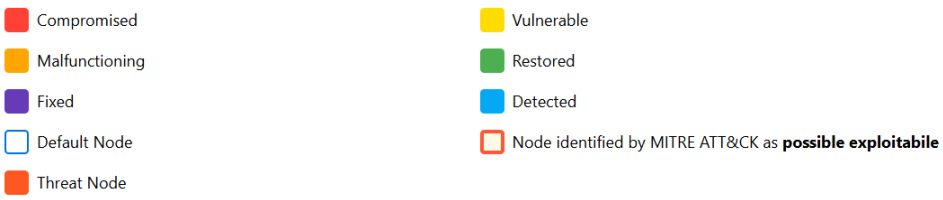

Figure 20: TAMELESS results visualization legend

10.9 Evaluation of the results

10.9.1 Micro level (building automation)

10.9.1.1 The Baseline: BACnet Unsecured

The first scenario analyzed is a totally unsecured BACnet/IP architecture. The attacker gains remote access to the corporate network and, because everything is in plaintext, compromises easily the management_software, the ip_sc_router, the bacnet_ip_controller, and finally the hvac_system.

TAMELESS analysis:

In the unsecured scenario, the TAMELESS formal logic models a permissive and flat architecture that lacks cryptographic boundaries. The attack chain starts with a human attacker, who is compromised by malicious intent and acts as the source to spread a network intrusion. The formal mathematical progression develops in the following way:

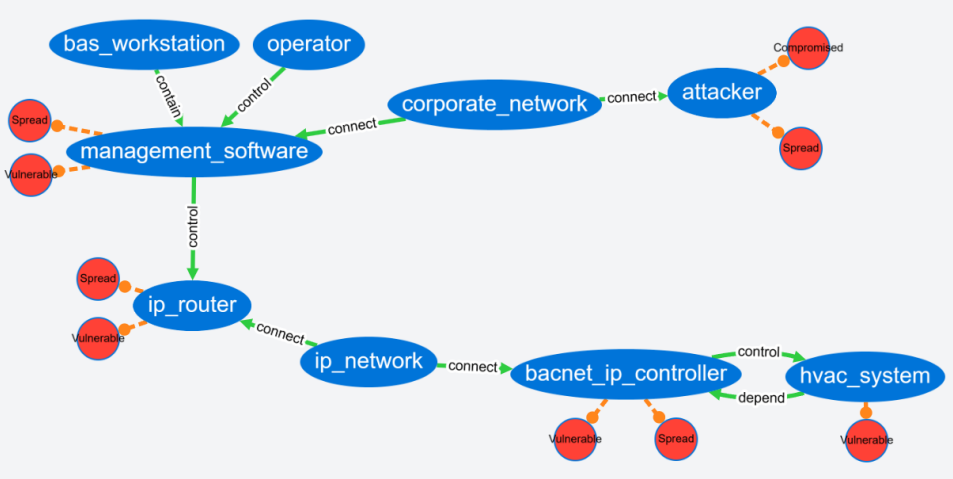

Figure 21: Graph representation of the unsecured BACnet/IP testcase

- The network intrusion travels through the corporate network and reaches the management software, which is vulnerable to network intrusions.
- Once compromised, the management software spreads the threat via lateral movement. Because it controls the IP router, the latter will also be vulnerable to this lateral movement
- The IP router acts as a bridge via the IP network to connect to the edge device, the BACnet IP controller.
- Because the BACnet IP controller is vulnerable to lateral movement, it becomes compromised and spreads HVAC manipulation down the chain.
- Finally, since the compromised BACnet controller controls the physical HVAC system, which is vulnerable to HVAC manipulation, the physical asset is compromised too.

LLM analysis:

The LLM analysis finds several MITRE ATT&CK techniques that can be applied, giving a realistic view of how a real attacker could exploit this unsecured scenario.

- Initial access: The LLM validates that the human attacker can initiate the breach using Spearphishing Attachment (T1566).
- Lateral pivot: As the attack moves to the management software, the LLM flags the node as exploitable for Connection Proxy (T0884) and Exploitation of Remote Services (T1210), justifying it with the node's vulnerability to network intrusion.
- Network exploitation: because the BACnet IP controller is vulnerable to lateral movement, the LLM flags it as exploitable for network exploitation techniques

like Exploitation of Remote Services (T1210), Exfiltration Over Other Network Medium (T1011), and Exploitation for Privilege Escalation (T1068).

- Physical impact: at the final physical layer, the LLM correctly identifies the HVAC system as exploitable for Physical Device Manipulation (T1135) and Physical Device Access (T1115) due to the active HVAC manipulation threat.

This baseline scenario proves that legacy building automation without any security offers no resistance against network intrusion and physical manipulation

10.9.1.2 BACnet partially secured (legacy BACnet/IP)

This scenario describes a hybrid architecture where BACnet/SC is deployed at the supervisory level and protects the management_software, but legacy BACnet/IP still remains at the edge. Two attackers are present:

- remote_attacker: attempts a top-down attack from the corporate network but is blocked by the bacnet_sc_hub.
- local_attacker: connects directly to the unsecured ip_network at the edge and is able to compromise the bacnet_ip_controller and the hvac_system.

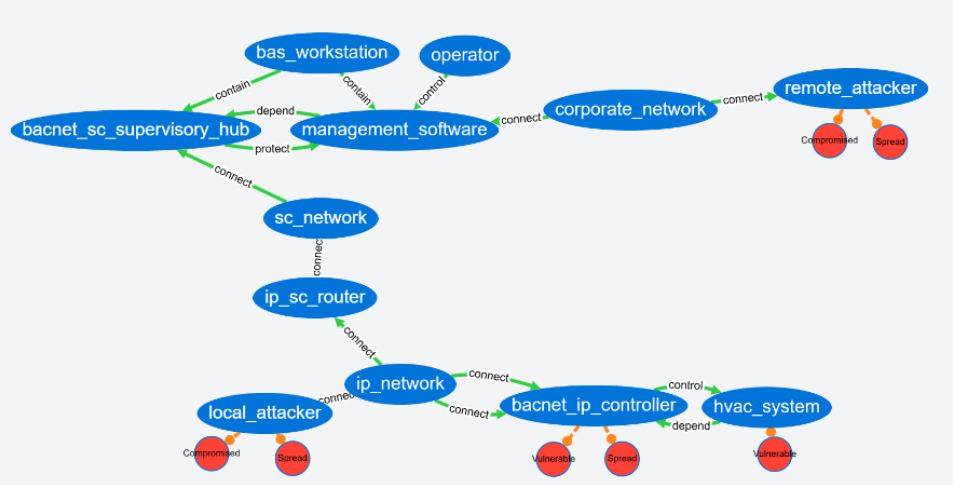

Figure 22: Graph representation of the partially secured BACnet testcase

TAMELESS analysis:

In the TAMELESS logic, the two attackers are modeled respectively as remote_attacker, spreading networkIntrusion, and local_attacker, spreading localNetworkIntrusion. The formal mathematical analysis outputs two main results:

- Top-Down attack: like in the previous scenario, the remote_attacker tries to target the management_software via the corporate_network. However, the management_software is explicitly protected by the bacnet_sc_supervisory_hub

against networkIntrusion, meaning that the top-down network attack path is blocked. The management_software remains vulnerable to the threat but is formally prevented from becoming compromised.

- Bottom-Up attack**:** On the other hand, the local_attacker targets the unprotected edge layer by connecting to the bacnet_ip_controller via the ip_network. Because there is no security node that protects the controller at this level, it is subject to the localNetworkIntrusion and becomes compromised. The compromised controller then spreads the hvacManipulation threat in the physical level, and the hvac_system is formally evaluated as compromised because the local attack stays unmitigated.

LLM analysis:

The generative LLM analysis outputs an interesting contrast by looking for logical bypasses: it captures both a physical breach at the edge and theoretical bypasses at the management layer.

- Bypassing formal protection: Even if TAMELESS blocks the networkIntrusion mathematically at the management layer, the LLM flags the management_software as exploitable for Connection Proxy (T0884) and Exploitation of Remote Services (T1210). The LLM states that while bacnet_sc_supervisory_hub protects successfully against network intrusion, it does not prevent an attacker from using a connection proxy attack if the host itself is compromised.
- Local edge exploitation: At the edge level where the local attack succeeds, the LLM maps the bacnet_ip_controller vulnerabilities to Exploit Public-Facing Application (T1190) and System Network Connections Discovery (T1049). The LLM validates these techniques because the controller is vulnerable to local network intrusions.
- Physical impact: Finally, following the formal TAMELESS logic that identified the HVAC system as compromised, the LLM flags the hvac_system for Physical Device Access (T1110) and Physical Device Tampering (T1135). The LLM confirms that the physical level can be exploited because of its vulnerability to physical manipulation originating from the compromise of the upper level.

This scenario confirms the importance of defense in depth. While BACnet/SC secures the corporate network from direct spoofing and manipulation, the lack of protection at the edge level allows local attackers to get full physical control. The LLM underlines that even remote attackers can bypass the defenses in the corporate network if there is a lack of host-level security, other than the communication-level security provided by BACnet/SC.

10.9.1.3 Full BACnet/SC architecture

This scenario represents an example of modern architecture where every component, from workstation down to the edge, supports BACnet/SC. The same two attackers are present:

- remote_attacker: Blocked by the supervisory bacnet_sc_hub.
- local_attacker: Attempts to connect to the edge network but is blocked because the edge now has a BACnet/SC edge hub that requires valid TLS certificates

However, the generative analysis still highlights some critical vulnerabilities that are related to host-level security.

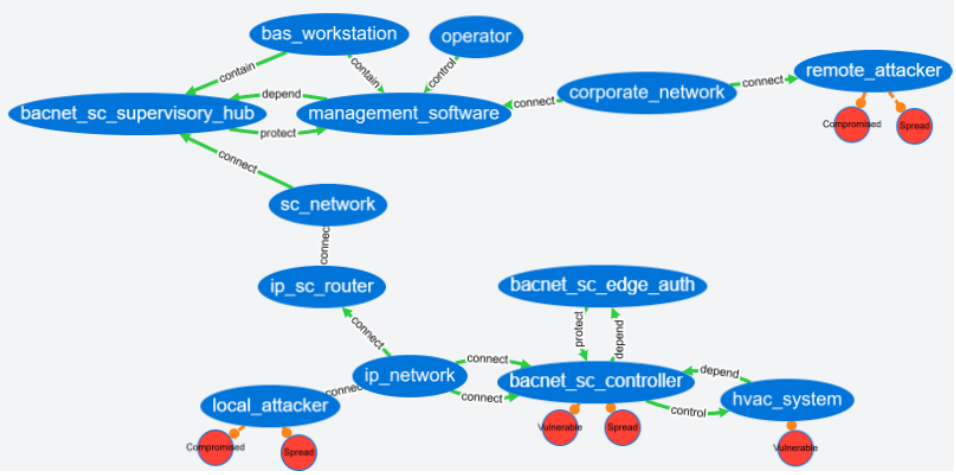

Figure 23: Graph representation of the fully secured BACnet testcase

TAMELESS analysis:

The formal analysis by TAMELESS evaluates the topology as secure to defend against both the remote_attacker (spreading networkIntrusion) and the local_attacker (spreading localNetworkIntrusion):

- Remote attack blocked: as observed in the previous case, the bacnet_sc_supervisory_hub applies a protect edge to the management_software, stopping the remote_attacker to perform a network intrusion via the corporate_network
- Local attack blocked: the bacnet_sc_edge_auth node directly protects the bacnet_sc_controller against the localNetworkIntrusion threat.

Therefore, according to the Prolog formal analysis, the network is safe because both top-down and bottom-up network paths are blocked. The management_software and bacnet_sc_controller, the main points of compromise in the unsecured architecture,

are now protected (as it's possible to see in Figure 23, where the yellow color indicates that they are potentially vulnerable but not compromised anymore) and the latter cannot spread hvacManipulation threat, leaving the hvac_system node uncompromised.

LLM analysis:

While the formal analysis proves the efficacy of the BACnet/SC cryptography against network intrusions, the LLM exposes once again host-level risks which can still compromise this architecture:

- The Proxy Bypass persists: even with the bacnet_sc_supervisory_hub, the LLM keeps flagging the management_software as exploitable for Connection Proxy (T0884) and Exploitation of Remote Services (T1210). The LLM reasons that while the hub protects the software from spoofing at the network layer, it does not prevent an adversary from exploiting an underlying OS vulnerability or using a connection proxy directly on the host machine to tunnel through the protections.
- Local infiltration risk: despite the presence of the bacnet_sc_edge_auth protecting the controller, the LLM flags the authentication node itself for Local Network Infiltration. A local attacker, because of the physical proximity, could attack the bacnet_sc_edge_augh node itself and physically tamper it. Moreover, the LLM flags the bacnet_sc_controller for System Network Connections Discovery (T1049), because even if the node cannot be compromised to execute commands, an attacker could still map its connections.
- Persistent physical risk: although TAMELESS proves the hvac_system is safe, the LLM flags it for Physical Device Access (T1110) and Physical Device Manipulation (T1135), because a local attacker can physically tamper with them regardless of protections at the network layer.

This fully secured scenario proves that while cryptographic standards like BACnet/SC can eliminate the threat of unauthorized network intrusions, they cannot substitute host-level security, endpoint security, and physical access controls.

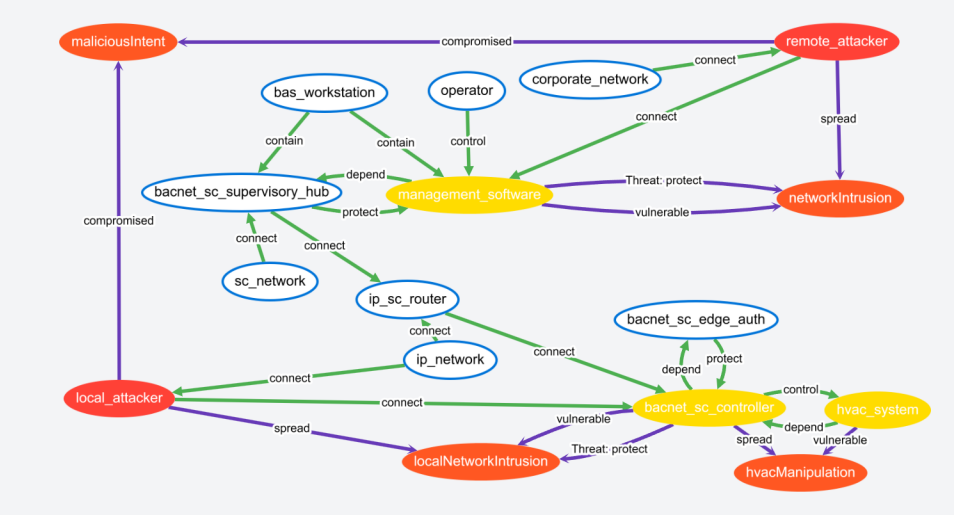

Figure 24: TAMELESS attack graph from formal analysis for the full BACnet/SC testcase

10.9.1.4 Full BACnet/SC with phishing bypass

This final scenario for the micro level shows how the previous security state reached by a fully secured BACnet/SC architecture falls off if an adversary exploits human vulnerabilities, completely bypassing the defenses at network layer. The attacker targets via phishing the human operator and obtains valid credentials, gaining authorized but malicious access to the network that goes unchecked.

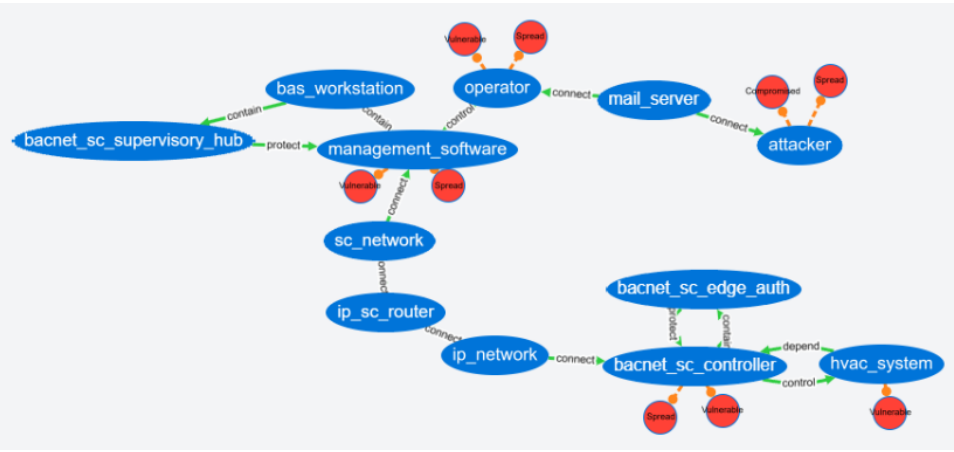

Figure 25: Graph representation of the fully secured BACnet with phishing bypass testcase

TAMELESS analysis:

TAMELESS logic models an attack that originates from an external email server, separate from the corporate network. The formal analysis finds the following:

- Human breach: the attacker executes successfully a phishingAttack via the

mail_server against the human operator. Because the operator is vulnerable to this threat, the node becomes formally compromised.

- Credential abuse: The compromised operator has legitimate control privileges over the management_software. The attacker uses this access to spread the credentialAbuse threat. Because the management_software is vulnerable to credential abuse, it becomes compromised.

- Cryptographic bypass: once the management_software is compromised using valid credentials, it spreads the authorizedMaliciousCommand threat to the edge level. The bacnet_sc_supervisory_hub works against networkIntrusion but it has no formal protect relation against an authorizedMaliciousCommand. The threat can therefore pass cleanly through the sc_network to the ip_sc_router and reach the bacnet_sc_controller.

- **Physical compromise:** since both the router and edge controller are vulnerable to authorized malicious commands, they become compromised and spread hvacManipulation to the hvac_system, compromising the physical level.

LLM analysis:

The LLM analysis aligns with the formal model and identifies social engineering risks, other than the host-level security bypasses of the other scenarios, which remains still.

- Initial vector: the LLM identifies the attacker as executing Phishing for Information (T1598) and Phishing (T1566). It then validates that the operator is susceptible to Phishing: Spear-Phishing Attachment (T1566.001) and User Execution: Malware (T1204), explaining it with the node's vulnerability to phishing attacks.

- Host and credential abuse: at the management layer, the LLM flags the management_software for identity-based techniques, such as Cloud Administration Command (T1651) and Windows Credential Manager (T1555.004). The LLM notes that compromising the software through credential abuse allows an adversary to gain or add additional credentials to maintain a persistent authorized access.

- Authorized downstream execution: since the attack moves downward, the LLM flags the ip_sc_router and bacnet_sc_controller for Authorized Malicious Command (T1218) and Command and Control (T1204).

- Physical impact: finally, matching outcome of the formal analysis, the LLM flags the hvac_system for Physical Device Access (T1110) and Physical Device Manipulation (T1133).

BACnet/SC provides cryptographic security against external network attacks, but it cannot defend against human compromise, which can cause the very same encrypted tunnels to be used for the delivery of malicious data using valid stolen credentials. This closes the micro-level analysis and proves once again how network defenses can't work alone but must be paired with host-level security and a zero-trust architecture to secure the Building Automation field.

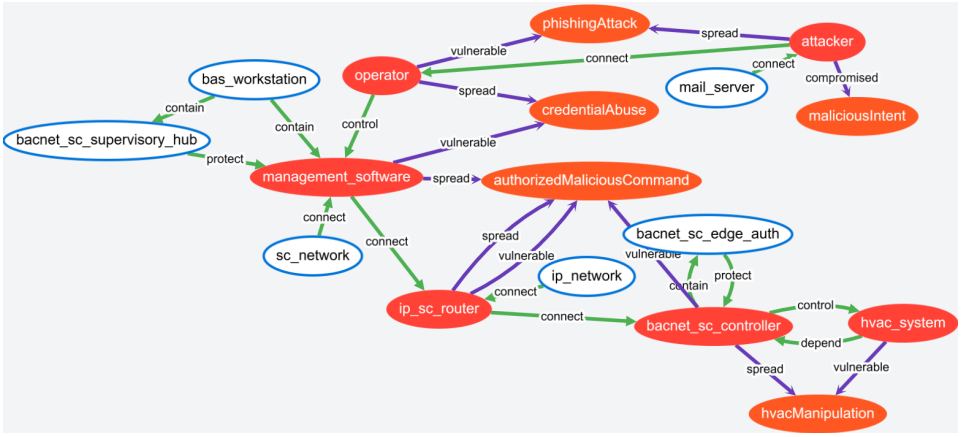

Figure 26: TAMELESS attack graph from formal analysis for the BACnet/SC phishing bypass testcase

10.9.2 Edge / prosumer level (smart metering & Demand Response)

In this level the focus goes on the interaction between Demand Response signals (OpenADR) and Smart metering infrastructure (DLMS/COSEM). The utility VTN sends pricing or load shedding signals to the building's VEN over the internet. The VEN then coordinates with the local smart meter to execute DR events and report data back to the utility. Two possible attack scenarios were evaluated for this architecture:

1. Remote spoofing (OpenADR): A remote attacker intercepts the traffic between VTN and VEN. Using a rogue VTN, the attacker sends spoofed Demand Response signals, such as a "price drop" signal to the VEN during peak hours, causing a power surge (high load devices are turned on).

2. Local tampering (DLMS/COSEM): A local attacker physically connects to the port of the smart meter (or intercepts the Home Area Network) to perform a false data injection, altering consumption data.

10.9.2.1 The baseline: unsecured edge architecture

OpenADR signals are sent over plaintext HTTP, and DLMS/COSEM uses "Lowest Level Security" (no password). A remote attacker spoofs the VTN and causes the

VEN to send a malicious load command to the meter.

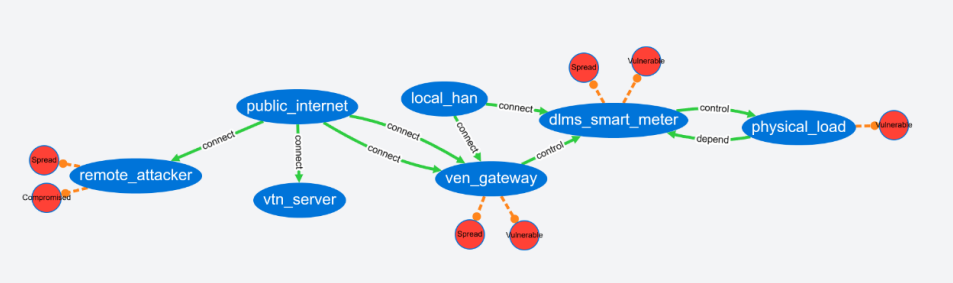

Figure 27: Graph representation of the unsecured edge level testcase

TAMELESS analysis:

TAMELESS identifies an attack path caused by this open architecture where VTN and VEN communicate without TLS encryption, and the smart meter lacks any form of authentication:

- External injection: the attack starts from remote_attacker, compromised by maliciousIntent, that spreads the spoofedDrSignal threat via the public_internet.
- Gateway compromise: the ven_gateway is completely vulnerable to the spoofed signals because he lacks entity authentication or encryption of the payload, so it accepts the fake pricing/demand parameters and becomes formally compromised.
- Lateral movement to metering: the compromised ven_gateway spreads maliciousLoadControl threat across the Home Area Network local_han to the dlms_smart_meter. The smart meter, also vulnerable to malicious control, becomes compromised because of its lack of security controls.
- Physical impact: the compromised smart meter now operates on false data, originating a gridSurge threat to the physical_load node, which is evaluated as compromised.

LLM analysis:

The LLM validates the attack path of the formal model and maps the vulnerabilities to specific ICS attack techniques:

- Internet exploitation: the LLM identifies correctly the public_internet as exploitable, flagging it for Exploit Public-Facing Application (T0819) and the abuse of standard Web Service (T1102) and Application Layer Protocol (T1071) channels. This aligns with the real world mechanics of OpenADR, which utilizes JSON payloads and RESTful APIs.

- Signal spoofing and malicious control: the LLM maps the spoofedDrSignal threat in the formal model to corresponding MITRE techniques. The ven_gateway is flagged for Spoofing (T1036) and Malicious Load Control (T1134), justified by the fact that the gateway completely lacks any control to recognize these threats.
- Smart meter manipulation: the attack reaches the dlms_smart_mater, flagged for Trusted Developer Utilities Proxy Execution (T1127) and confirmed to be exposed to the grid surge effect.
- Physical impact: finally, the physical_load is correctly flagged for Physical Device Compromise (T1115), because it's controlled by the compromised smart meter

This baseline scenario proves as expected that without authentication and TLS tunnels, edge gateways trust blindly any signal over the internet and allow remote attackers to cause a power surge without needing to breach a physical perimeter or a corporate IT network.

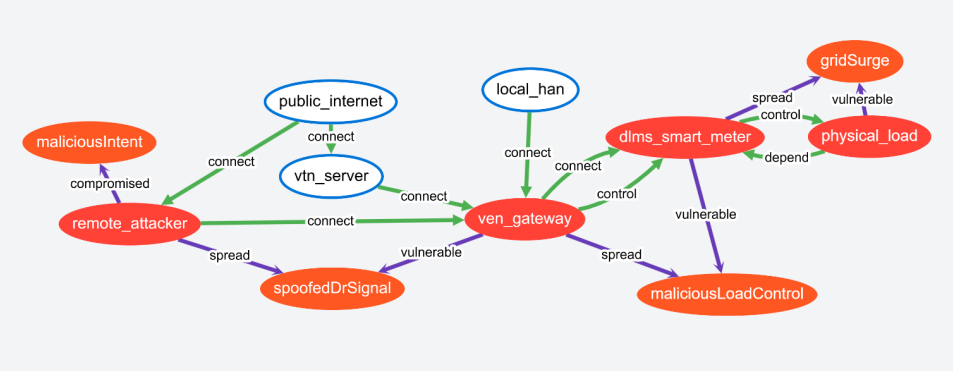

Figure 28: TAMELESS attack graph from formal analysis for the unsecured edge level testcase

10.9.2.2 Partially secured edge architecture

This time, the utility enforces TLS tunnels and client/server authentication, securing the connection between VTN and VEN. However, if the local DLMS connection stays unprotected, a local attacker can bypass the internet and attack the smart meter directly, or through the local HAN.

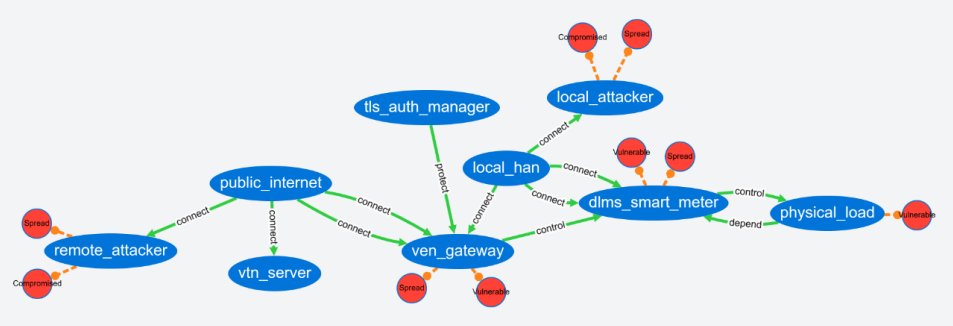

Figure 29: Graph representation of the partially secured edge level testcase

TAMELESS analysis:

In this scenario there is the addition of the tls_auth_manager cyber node which protects the VTN-VEN link, and the local_attacker node which performs the local attack.

- Remote spoofing stopped: the remote_attacker tries to inject a spoofedDrSignal via the public_internet like in the previous case, but the tls_auth_manager generates an explicit protect edge defending the ven_gateway against this threat. The Prolog engine mathematically stops the attack and the ven_gateway remains vulnerable but is not compromised.
- Local false data injection: at the same time, a local_attacker (for example a rogue prosumer trying to manipulate billing or grid physics) acts on the local_han and injects the falseDataInjection threat directly into the dlms_smart_meter. Because the smart meter lacks protection, the formal engine evaluates it as compromised.
- Physical impact: The compromised meter spreads the gridSurge threat downwards, causing the physical_load to be formally compromised.

LLM analysis:

The LLM confirms the results of formal analysis but also identifies additional issues in the cryptographic implementation itself:

- TLS blind spot: the LLM flags the tls_auth_manager node itself for Private Keys (T1145) and Digital Certificates (T1588.004). The LLM notes that while TLS protects the network connection, the certificate manager is a cyber node that can be exploited (for example with stolen or forged certificates) to bypass the cryptography.
- False data injection: the LLM flags the local_attacker for Masquerading (T1036), meaning that it correctly identifies that the local adversary is spoofing

or impersonating legitimate meter traffic on the Home Area Network to execute the False Data Injection. The dlms_smart_meter is then flagged for GridSurge (T1483.001) and FalseDataInjection (T1212). It also notes that the meter can be manipulated using Remote Services (T1210), proving that the lack of local authentication leaves the device exposed completely in the locan HAN.

- Physical access: finally, the LLM flags the physical_load for Physical Device Compromise (T1115), meaning that this partially secure architecture still has the same final result of the previous unsecured case.

This scenario underlines the principle of the weakest link in Smart Energy Systems. Failing to secure the prosumer's smart meter can cause localized physical damage, even when the utility signals for Demand Response with OpenADR are cryptographically secure.

10.9.2.3 Fully secured edge architecture

Like the previous case, OpenADR is secured with TLS, but this time also the smart meter is protected using DLMS/COSEM High Level Security (HLS) with entity authentication and payload authenticated encryption.

TAMELESS analysis:

The topology evaluated by the formal analysis is protected by the tls_auth_manager node at the utility level and dlms_hls_crypto node at the prosumer level. Prolog concludes that this architecture contains both local and remote attacks:

- Remote spoofing blocked: as observed previously, the remote_attacker tries to inject a spoofedDrSignal via the public_internet. The ls_auth_manager generates a protect edge targeting the ven_gateway and avoiding this threat. The gateway remains vulnerable but does not become compromised, stopping the top-down attack.
- Local FDI blocked: the local_attacker attempts a falseDataInjection attack against the dlms_smart_meter on the local_han. This time, the dlms_hls_crypto node provides a direct protect relation that specifically mitigates False Data Injection. The formal engine evaluates that the meter cannot be compromised.
- Physical asset Safe: because both the ven_gateway and dlms_smart_meter are not in a compromised state, no gridSurge threat is generated, and the formal logic evaluates the physical_load as safe.

LLM analysis:

As in the fully secured BACnet/SC scenario, while the formal results evaluate this architecture as completely secure, the LLM analysis exposes additional host-level risks that are not solved by network layer security:

- Cryptographic exploitation: as in the previous case, the tls_auth_manager is flagged for Private Keys (T1145) and Digital Certificates (T1588.004). Similarly, the dlms_hls_crypto node is flagged for Use Alternate Authentication Material (T1550.003) and Standard Cryptographic Protocol (T1032), noting that adversaries can attempt to bypass or abuse cryptographic modules locally instead of breaking the math.
- Gateway targeting: the LLM flags the ven_gateway for deep-network techniques like Multi-hop Proxy (T1090.003), Internal Proxy (T1090.001), and Web Shell (T1508.001). It reasons that while OpenADR signals are protected by TLS, the operating system hosting the gateway remains a vulnerable target connected to the internet.
- Smart meter compromise: the LLM explicitly flags the dlms_smart_meter as exploitable for host-level execution techniques like Execution (Scheduled Task/Job T1053), Persistence (Boot or Logon Autostart Execution T1547.001), Lateral Movement (Remote Services T1021).
- Physical asset compromise (LLM hallucination): even if the smart meter is flagged for host-level compromises, the LLM evaluates all the techniques against the physical_load, including Physical Device Compromise (T1115), as completely unexploitable. The reasoning of the LLM was the following: "no direct relationships in the system graph that enable a cyber-based attack vector" on the physical node. This is an example of logical hallucination, because the smart meter (that can be compromised by host-level vulnerabilities as noted in the previous point) has direct physical control over the physical load. The LLM identified successfully the host flaws on the smart meter but failed to propagate that risk to the physical level.

10.9.3 Macro level (substation & transmission)

In this scenario, the architecture consists of a central scada_master connected to the public or corporate network via a scada_wan. From the SCADA master, the operational technology separates into two different paths: the DNP3 path, where the master directly controls a dnp3_outstation, which controls a physical voltage_sensor, and the IEC 61850 Path, where the master connects via a local substation_lan to an iec61850_ied (Intelligent Electronic Device), which controls a high-voltage circuit_breaker.

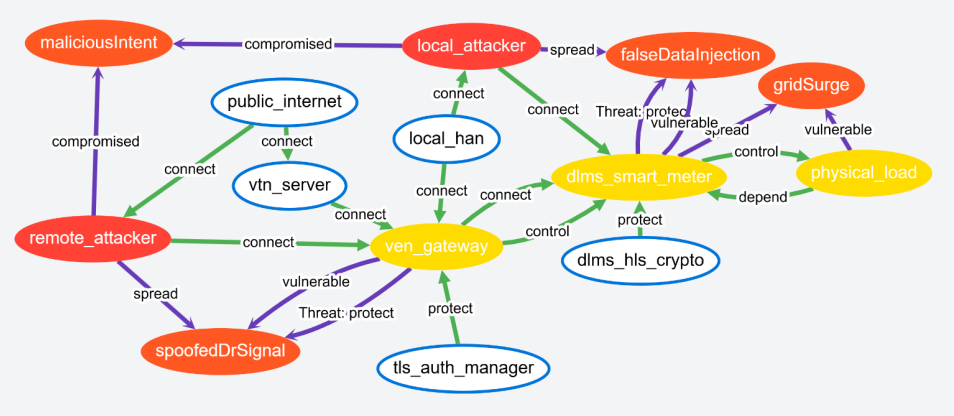

Figure 30: TAMELESS attack graph from formal analysis for the fully secured edge level testcase

10.9.3.1 The baseline: unsecured macro level

This is an example of architecture with legacy infrastructure where both DNP3 at the outstation level and IEC61850 (MMS, GOOSE/SV) at the substation level, completely lack any security feature. An attacker compromises the SCADA Master, which blindly sends malicious DNP3 commands to the outstation (altering voltage sensor readings) and malicious MMS commands to the IED (triggering a spoofed GOOSE message to trip the breaker).

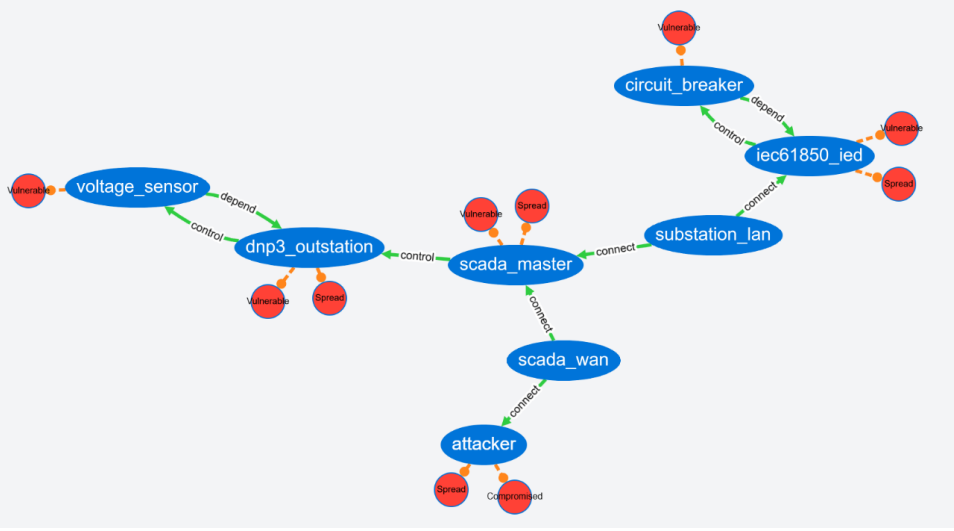

Figure 31: Graph representation of the unsecured macro level testcase

TAMELESS analysis:

In the unsecured state, the TAMELESS formal logic correctly identifies a cascading compromise in both transmission paths:

- SCADA master compromise: the attack starts from a human attacker compromised by maliciousIntent. Going through the scada_wan, the attacker spreads

a networkIntrusion threat to the central scada_master. Because the master lacks formal protection against network intrusions, it enters a compromised state.

- Path 1 - DNP3 outstation compromise: because the scada_master controls the dnp3_outstation, it spreads the maliciousDnp3Command threat down to the outstation, which becomes compromised. The compromised dnp3_outstation then spreads sensorManipulation to the physical voltage sensor.
- Path 2 - IEC 61850 substation compromise: the scada_master injects a maliciousMmsCommand into the iec61850_ied through the substation_lan. The IED becomes compromised and executes maliciousGooseMessage, which in turn causes malicious SV commands to the circuit_breaker, that becomes compromised.

LLM analysis:

The LLM output confirms the attack path of the formal model and translates it into specific MITRE ATT&CK techniques:

- WAN/Master breach: the LLM flags the scada_wan and scada_master for Exploit Public-Facing Application (T1190), Connection Proxy (T0884), and Remote Services (T1210). The LLM explicitly notes that the network intrusion makes the central master vulnerable to lateral movement and software exploitation.
- DNP3 path compromise: for the DNP3 path, the LLM flags the dnp3_outstation for deep host level control, including Scheduled Task/Job (T1053) and Scripting (T1064). Spreading to the lower level as physical impact, the LLM flags the voltage_sensor for Physical Device Manipulation (T1130) and Physical Device Access (T1110), validating that the compromised cyber nodes allow physical tampering.
- IEC 61850 path compromise: on the substation automation path, the LLM flags the iec61850_ied for Modify System Configuration (T1531) and Loss of Control (T0827). It understands that injecting malicious MMS/GOOSE commands takes away control of the substation from the legitimate operators. As a result, the circuit_breaker is flagged for Physical Device Tampering (T1110).

This baseline scenario underlines the insecurity of legacy OT environments where DNP3 and IEC 61850 endpoints trust blindly commands coming from the SCADA master. A breach at the OT layer allows control over physical equipment like sensors and substation circuit breakers.

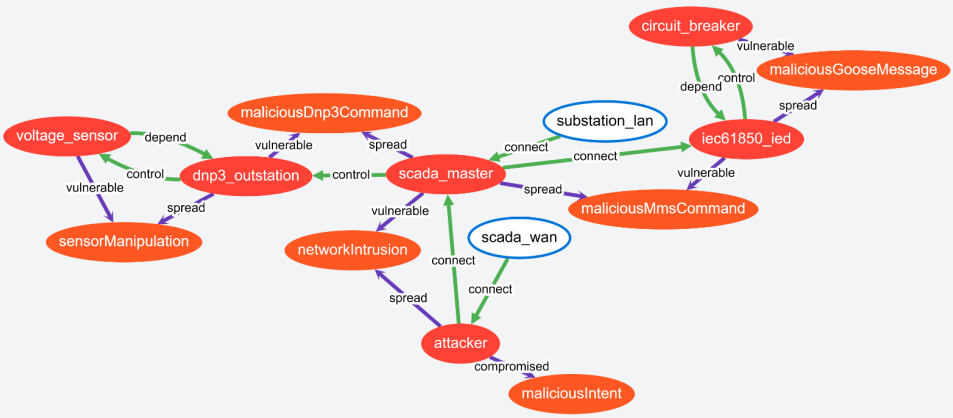

Figure 32: TAMELESS attack graph from formal analysis for the unsecured macro level testcase

10.9.3.2 Partially secured macro level

This is a realistic industrial scenario where remote connections are secured by the utility, using DNP3-SA and IEC 62351 MMS TLS. While the remote attacker of the previous case is blocked, because GOOSE and SV messages require strict milliseconds latency, the utility left the communication between IEDs and between IED and physical level unprotected. A local attacker injects a spoofed GOOSE message directly into the IED, bypassing MMS security.

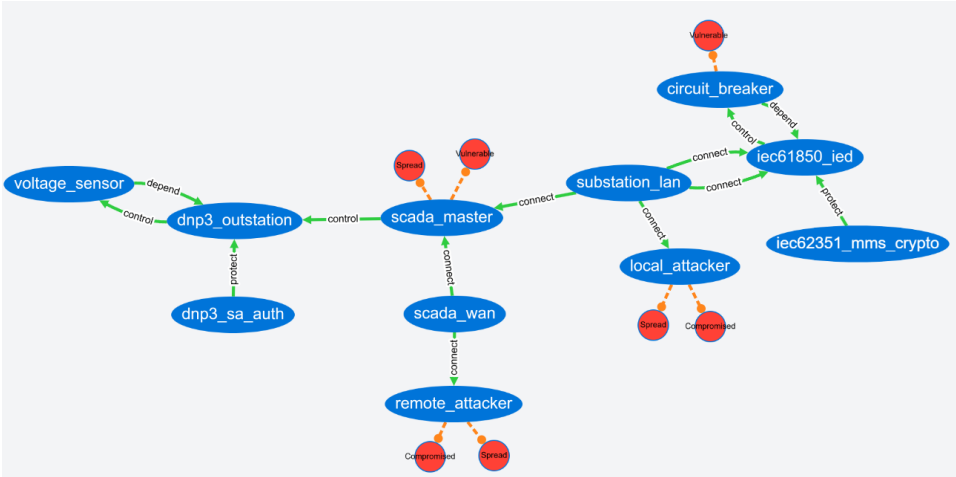

Figure 33: Graph representation of the partially secured macro level testcase

TAMELESS analysis:

The formal model correctly proves that the security posture is asymmetric. Even if the remote attack is contained, it identifies that the physical grid remains exposed to local exploitation:

- Remote attack blocked: the remote_attacker breaches successfully the scada_wan

and compromises the central scada_master. From here, the master attempts to spread both a maliciousDnp3Command and a maliciousMmsCommand. However, the TAMELESS engine evaluates the explicit protect edges generated by the dnp3_sa_auth and iec62351_mms_crypto nodes. The remote attack is blocked on both paths, the dnp3_outstation and the connected voltage_sensor remain uncompromised.

- Local attack: a local_attacker operates directly on the substation_lan and bypasses the SCADA master. This attacker injects a gooseSpoofing threat against the iec61850_ied. Because the IED lacks GOOSE authentication, it is formally evaluated as vulnerable and becomes compromised by the local spoofed traffic.
- Physical impact: because of its control relationship with the circuit breaker, the compromised iec61850_ied spreads maliciousGooseMessage and the circuit_breaker node is evaluated as compromised.

LLM analysis:

The LLM supplements the formal logic by mapping this specific local bypass to real-world MITRE ICS techniques:

- IED exploitation via unauthenticated protocols: it flags the iec61850_ied for execution and manipulation vulnerabilities, including Program Organization Units (T0844), Malicious Code (T1204), and Code Injection (T1129). The generative AI deduces that because the IED accepts unauthenticated GOOSE messages, a local adversary can inject malicious code or manipulate its internal logic blocks as if they were a trusted peer relay.
- Accurate physical consequence: the LLM flags the circuit_breaker for Device Restart/Shutdown (T0816), because the property maliciousGooseMessage creates an exploitable relationship that allows the adversary to trip forcefully the physical breaker.
- Host level vulnerabilities: even if TAMELESS proved the MMS cryptography blocked successfully the remote attack, the LLM continues to identify for host-level vulnerabilities. It flags the iec62351_mms_crypto node itself for Encrypted Channel (T1573) and Modify System Configuration (T1136), noting that an adversary who has breached the local network could attempt to abuse the encryption algorithms or modify the configuration files of the security modules.

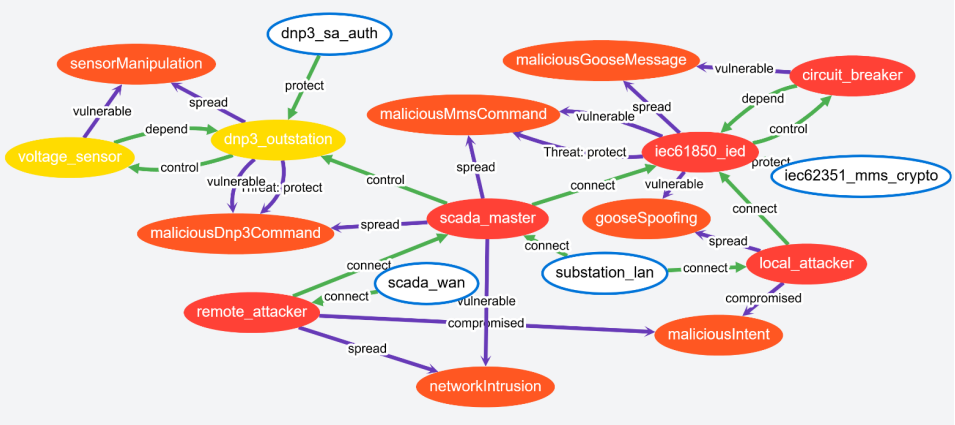

Figure 34: TAMELESS attack graph from formal analysis for the partially secured macro level testcase

10.9.3.3 Fully secured macro level

This is an example of industrial scenario with complete communication protocol security. The utility secured remote connections using DNP3-SA and IEC 62351 MMS TLS, blocking the remote attacker like in the previous case. Additionally, the utility implemented IEC 62351-6 to authenticate local Layer 2 GOOSE messages. In this way, if a local attacker tries to inject spoofed traffic, the unauthenticated GOOSE messages are rejected instantly by the IED.

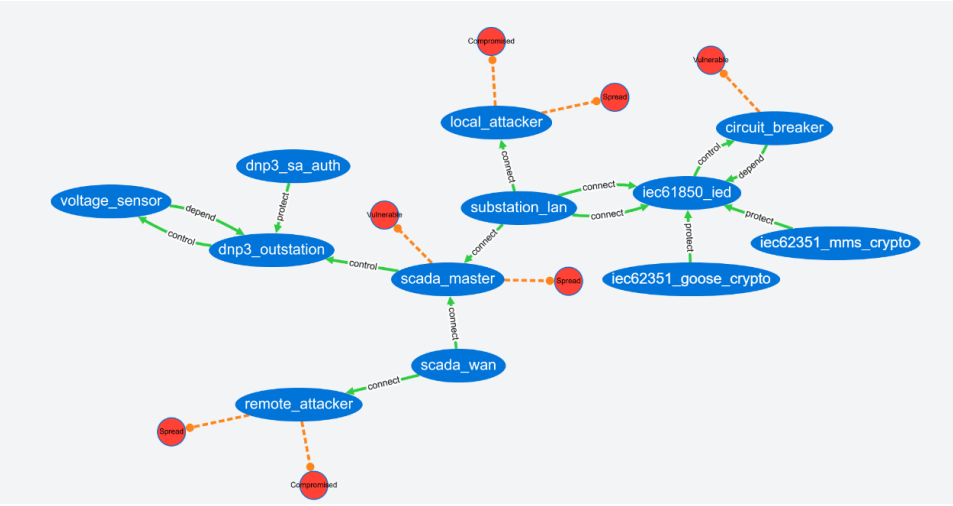

Figure 35: Graph representation of the fully secured macro level testcase

TAMELESS analysis:

The formal analysis proves that both remote and local attack paths are stopped, ensuring that the physical circuit breaker remains safe:

- Central breach: the remote_attacker goes through the scada_wan and compromises the central scada_master via a networkIntrusion.

- Remote attacks on DNP3 and IEC 61850 blocked: exactly like in the previous scenario, both the remote attack against the dnp3_outstation and against the IEC 61850 substation are blocked.
- Local attack blocked: differently from the partially secured scenario, when the local_attacker attempts to inject unauthenticated gooseSpoofing traffic into the iec61850_ied, the new iec62351_goose_crypto node protects the IED and stops the spreading of the threat, leaving the IED secure.
- Physical safety: because all cyber attack paths to the dnp3_outstation and iec61850_ied are stopped, no malicious commands are passed to the physical devices. The formal engine evaluates the circuit_breaker and voltage_sensor as safe from physical compromise.

LLM analysis:

Once again, while the formal model proves that the OT communication level is mathematically secure, the LLM identifies host level vulnerabilities related to the cryptography nodes, similarly to the previous scenario:

- Cryptographic hosts attacks: the LLM flags the cryptographic nodes themselves as primary targets, both iec62351_mms_crypto and iec62351_goose_crypto for Encrypted Channel (T1573), Standard Cryptographic Protocol (T1032), and Modify System Configuration (T1136).
- Control center vulnerability: the LLM underlines that while the physical grid is safe, the control center is not. It flags the compromised scada_master for network breaches, including Exfiltration Over Alternative Protocol (T1438), Data Encrypted (T1022), and Denial of Service (T1490). Even in a fully secured OT environment, a compromised SCADA master is critical and needs host level defences to prevent data loss or DoS.
- Physical consequence: the LLM evaluates the circuit_breaker as possibly susceptible to Device Restart/Shutdown (T0816), because of the host-level vulnerabilities of the first point that make still possible to trigger malicious-GooseMessage, in that case.

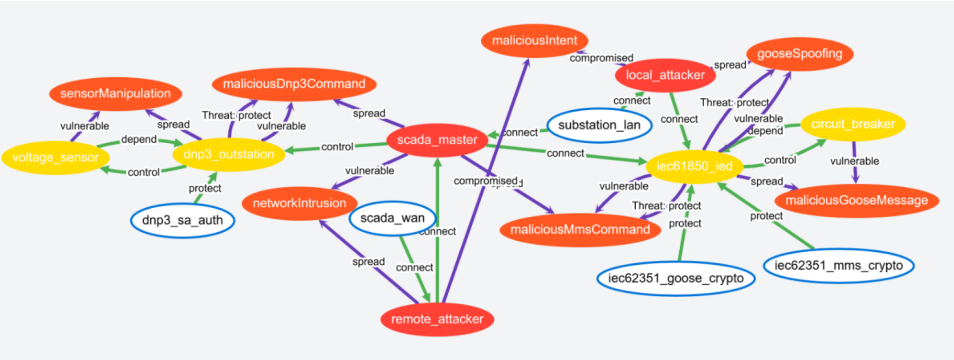

Figure 36: TAMELESS attack graph from formal analysis for the fully secured macro level testcase

10.10 Extension of the framework: hybrid consequence audit

10.10.1 Motivation

The evaluations of the SES use cases showed a limitation when comparing the formal Prolog analysis with the LLM MITRE ATT&CK one. The LLM identified effectively different realistic adversarial techniques, especially at the host level, but in some cases, it was less reliable when the reasoning required multi-hop propagation across the topology. This is a relevant limitation in SESs, where many security consequences are not located at the same node where the initial vulnerability appears.

10.10.2 Objective

For this reason, an additional component was introduced in the extended framework: the hybrid consequence audit. The objective of this extension is to compare the deterministic results produced by TAMELESS with the vulnerability assessment produced by the LLM.

The audit does not replace these components, but instead it acts as a consistency check layer between them, using the graph structure to verify if the LLM missed any important physical consequence.

10.10.3 Classification of inconsistencies

The audit identifies three main types of inconsistencies between the formal and generative analyses.

A missed physical consequence takes place when the LLM identifies a node as exploitable, but then it fails to recognize the possible impact on a physical node

downstream.

A formal-only impact happens when the formal analysis evaluates a node as compromised or malfunctioning, but the LLM fails to identify any exploitable technique for that node.

An LLM-only exploitability occurs when the LLM identifies a real-world vulnerability that is not recognized during the formal analysis. This can suggest that the formal model should be enriched with additional threats, vulnerabilities, or protection relations.

10.10.4 Application to the evaluation

This audit is particularly useful for the interpretation of the fully secured scenarios. In these cases, TAMELESS correctly proves that a specific attack path at the protocol level is blocked by a cryptographic mechanism. However, the LLM may still identify, on the same architecture, risks at the host level, such as certificate theft, abuse of authentication material, or compromise of the endpoint implementing the secure protocol.

The audit helps distinguish between these two levels of reasoning. It checks if a host-level weakness detected by the LLM could have a physical consequence downstream according to the topology.

A clear example is the edge level scenario, where in the fully secured architecture, the formal analysis evaluates that the spoofed demand-response signals and the local false data injection attack are blocked. However, the LLM identifies possible host-level vulnerabilities on the smart meter or cryptographic components. In the original output of the LLM, the physical load was classified as not exploitable because the model failed to propagate the smart meter risk to the physical layer. The hybrid consequence audit solves this contradiction by flagging the finding as a possible missed consequence, so that it can be evaluated from an analyst properly.

10.10.5 Interpretation of the results

This final audit provides a sort of warning mechanism indicating if, according to the topology, there is an inconsistency between the two analyses, and a propagation path should be further considered. In this sense, the audit improves the explainability of the hybrid analysis: it shows not only which nodes are vulnerable, but also where the formal and generative analyses disagree.

10.11 Conclusions of the evaluation

10.11.1 Need for comprehensive approach

One of the key findings in common with all the analyzed use cases is the need for a comprehensive approach that covers both local and remote vulnerabilities. At all levels, Micro, Edge, and Macro layers share a distributed architecture where a traditional approach that secures only the perimeter isn't enough. The formal evaluation demonstrated that implementing only top-down security at the control/management level is not sufficient when the localized controllers and physical layer remain unprotected. Securing only the IT/OT boundary does not protect the physical grid from localized and bottom-up attacks, like an unauthenticated layer 2 GOOSE message forcing a circuit breaker to trip, or a spoofed smart meter triggering a localized grid surge.

Furthermore, because these architectures are fundamentally linked, a localized failure can trigger a cascading event. The concept of defense in depth becomes critical, especially in this context where a threat breaching the central SCADA master could in theory start an advanced attack spanning from transmission substation down to building automation.

10.11.2 Limitations of generative AI

While the LLM assessment proved to be extremely useful in translating formal vulnerabilities into real world MITRE ATT&CK techniques, this evaluation also exposed some clear limitations. One of them was the logical propagation, and the most clear example was in the fully secured edge testcase evaluation. The LLM identified many host level vulnerabilities on the smart meter but, when evaluating the physical load downwards, the model concluded erroneously that the physical asset was perfectly safe because there was no"direct cyber-based attack vector."

This logical hallucination is certainly heavily dependent on the choice of the specific model, which was in the case of this thesis a Groq-optimized serving version (llama-3.1-8b-instant) and chosen for its very low latency and overall cost and token efficiency, solving tasks cheaply while maintaining acceptable quality. This limitation was mainly forced by the use of a cloud based free API which imposes restrictions on tokens per minute and daily total tokens. A more complete approach should include a more capable model, especially in terms of multi-hop deductive reasoning across complex graphs.

However, this logical hallucination exposes a flaw that is common of all generative AI and is always present in a bigger or smaller degree. Therefore, generative models, when applied to threat modeling, cannot be trusted blindly to evaluate ultimate

physical consequences in multi-tier topologies like the one of a Smart Energy System.

10.11.3 Formal verification vs. generative threat analysis

An important finding in the evaluation of the proposed scenario is the difference between formal mathematical verification provided by Prolog and the generative threat analysis of the LLM. The original TAMELESS analysis operates on a strictly deterministic logic at the network and protocol layer; when a cryptographic standard such as BACnet/SC or IEC 62351 is applied, the Prolog engine treats it as an absolute boundary, proving mathematically that unauthorized external spoofing and lateral network intrusions are blocked. However, the LLM exposes the limitations of relying only on network layer security. The LLM identifies host level bypasses that formal network logic inherently trusts, by correlating the graph architecture against the MITRE ATT&CK framework.

The difference of these results proves how cryptographic network protocols secure successfully the communication links, but they can't protect against threats caused by host level vulnerabilities, like an authorized malicious command coming from a compromised endpoint.

10.11.4 Necessity of the dual methodology

The findings of this thesis prove definitely that neither formal mathematical modeling or generative AI are sufficient on their own for modern threat modeling for ICSs; they are instead complementary. Formal engines based on graph logic like TAMELESS are deterministic and absolutely necessary to enforce logical consistency between nodes. They mathematically prove if a threat variable can pass through the topology and verify the physical state of the grid. However, formal math cannot formally model well the concept of a spear phishing campaign, a stolen digital certificate, or a zero-day web shell.

By integrating the generative LLM, the methodology maps theoretical mathematical proofs to real world threats. In particular, the LLM was able to go a step further from only protocol level security and integrated many host level vulnerabilities and flaws that are essential to evaluate in any architecture.

10.11.5 Final considerations

These are all the main findings of the evaluation performed:

1. Top-down security is insufficient: Securing wide-area networks does not protect against local, high-speed physical manipulation.

2. Cryptography is not enough: Even mathematically perfect protocols require secure host configurations and strict certificate management.

3. The Hybrid Approach is mandatory: Generative AI provides contextual vulnerability mapping (MITRE ICS), while formal logic (TAMELESS) is required to enforce multi-hop consistency and prove ultimate physical consequences.

Chapter 11

Conclusions and Future Developments

11.1 Conclusions

The transition from conventional power networks to Smart Energy Systems caused the creation of an infrastructure that is more distributed, two-way, and digitalized. Such features enable better flexibility and efficient integration of renewable energy resources. On the other hand, this shift implies an increase in the cyber-physical attack vector, as communication protocols become an integral part of the processes that impact the operation of the physical layer of the grid.

This thesis reviewed the most prevalent communication protocols used at different levels of a Smart Energy System: IEC 61850, DNP3, DLMS/COSEM, OpenADR, and BACnet/BACnet Secure Connect. It was found that all five protocols work within certain limitations that differ significantly from each other. For instance, the operation of IEC 61850 depends on extremely low latency in order to provide substation protection functionality, while DNP3 should be highly effective in the context of SCADA and distribution systems. At the same time, DLMS/COSEM should provide maximum security for constrained smart meters, OpenADR can leverage existing IT security tools because of more flexible timings, and BACnet/BACnet SC requires securing building automation systems while maintaining compatibility with legacy deployments.

A core contribution of the thesis is that the cybersecurity mechanisms should be tuned to fit the specifics of operation of the given protocol. Sometimes more secure cryptographic algorithms will not be applicable for time-sensitive communication, whereas lighter ones will be unable to secure communication between endpoints that exchange private and commercially sensitive information. As a result, the balance between security and performance considerations will have to include factors such as

latency, computational costs, bandwidth usage, and limitations on devices. Security cannot be applied consistently throughout the entire Smart Energy System.

Threat modeling and evaluation using the TAMELESS framework, Large Language Models, and the MITRE ATT&CK for ICS revealed that protocol-level security measures are necessary to limit cyber-physical attack vectors. Such secure versions as IEC 62351, DNP3 Secure Authentication, DLMS/COSEM security suites, OpenADR secured with secure web protocols, and BACnet/SC would help to reduce attacks such as spoofing, replay, manipulations, and unauthorized control. However, it was also shown that cryptography alone wouldn't be sufficient in creating resilient smart energy systems: secure endpoint configuration, certificate management, network segmentation, intrusion detection, and defense-in-depth strategies remain necessary.

11.2 Future Developments

First, future research could include more performance metrics in the assessment. This thesis discussed security and efficiency trade-offs mainly from an architectural and literature point of view. Experimental benchmarking can be used to measure latency, CPU and memory usage, and bandwidth overhead for different protocols, allowing a more accurate evaluation of which protections are feasible in each operational context.

Another possible direction for future development could extend the set of analyzed protocols and technologies. For example, future studies could include other protocols such as Modbus, OPC UA, MQTT, IEC 60870-5-104, KNX, Zigbee, Wi-SUN, LoRaWAN, OCPP, and ISO 15118. This would give a more comprehensive view of Smart Energy System communication, especially in scenarios where electric vehicles and IoT devices are involved.

Finally, the methodology could be validated on real Smart Energy System scenarios. The framework would be applied to infrastructures with real smart meters, controllers and SCADA devices, making it possible to compare predictions with real cyber-physical effects.

Bibliography

[1] Henrik Lund, Poul Alberg Østergaard, David Connolly, and Brian Vad Mathiesen. *Smart energy and smart energy systems*.

[2] Yizhe Xu, Chengchu Yan, Huifang Liu, Jin Wang, Zhang Yang, and Yanlong Jiang. *Smart energy systems: A critical review on design and operation optimization*. 2020.

[3] C. Donitzky, O. Roos, S. Sauty, and J. Krueger. *Communications for smarter energy systems*. Intel Corporation, White Paper. 2015 (cit. on pp. 7, 8, 21, 37, 45).

[4] S. M. S. Hussain, T. S. Ustun, and A. Kalam. *A Review of IEC 62351 Security Mechanisms for IEC 61850 Message Exchanges*. IEEE Trans. Industrial Informatics. 2020 (cit. on p. 18).

[5] J. C. Lozano, K. Koneru, N. Ortiz, and A. A. Cardenas. "Digital Substations and IEC 61850: A Primer". In: (2023). IEEE Communications Magazine (cit. on p. 14).

[6] M. G. Todeschini, G. Dondossola, and R. Terruggia. "Impact Evaluation of IEC 62351 Cyber Security on IEC 61850 Communications Performance". In: 25th International Conference on Electricity Distribution (CIRED), Madrid, Spain. 2019 (cit. on p. 22).

[7] iGrid T&D. *IEC 61850 Standard - Smart Guide Chapter*. iGrid T&D Smart Guide. URL: https://www.igrid-td.com/smartguide/iec61850.

[8] International Electrotechnical Commission. *IEC 61850*. IEC; Accessed Mar. 31, 2026. 2026. URL: https://iec61850.dvl.iec.ch/.

[9] Neuron Team. *IEC 61850 Protocol: Features, Information Model, and Combination with MQTT*. EMQ, Nov. 30, 2023. 2023. URL: https://www.emqx.com/en/blog/iec-61850-protocol.

[10] R. M. Augusto. *Safeguarding IEC 61850 Implementation in Industrial Control Systems: A Technical Odyssey*. Industrial Cyber, Feb. 5, 2024. 2024. URL: https://industrialcyber.co/expert/safeguarding-iec-61850-implementation-in-industrial-control-systems-a-technical-odyssey/.

[11] Typhoon HIL. *IEC 61850 Sampled Values protocol*. Typhoon HIL Software Manual. URL: https://www.typhoon-hil.com/documentation/typhoon-hil-software-manual/References/iec_61850_sampled_values_protocol.html.

[12] K. Saadi, B. Bou Farraa, and R. Abbou. "Methodology of conception, implementation and validation of an IEC 61850 communication system in smart grids demonstration platform". In: in 2018 International Conference on Smart Communications in Network Technologies (SaCoNeT). 2018 (cit. on p. 17).

[13] R. Amoah, S. Camtepe, and E. Foo. "Formal modelling and analysis of DNP3 secure authentication". In: (2016). Journal of Network and Computer Applications, 59, 345-360.

[14] R. K. Belchandan and A. Akhtar. "Comparative analysis of DNP3 and IEC 61850 from architectural, data mapping, data modeling and data reporting view". In: 2023 North American Power Symposium (NAPS). 2023.

[15] DNP Users Group. *Overview of DNP3 security version 6*. 2020 (cit. on pp. 27, 29).

[16] IEEE. *IEEE standard for electric power systems communications—Distributed network protocol (DNP3) (IEEE Std 1815-2012)*. 2012.

[17] J. Pérez García, A. L. Sandoval Orozco, and L. J. García Villalba. "DNP3 protocol taxonomy". In: vol. 123. 2026, p. 29.

[18] C. Rosborough, C. Gordon, and B. Waldron. "All about Eve: Comparing DNP3 secure authentication with standard security technologies for SCADA communications". In: Proceedings of the 55th Annual Minnesota Power Systems Conference. 2019 (cit. on pp. 28, 30).

[19] National Instruments. *Introduction to DNP3*. May 30, 2025. 2025. URL: https://www.ni.com/en/shop/seamlessly-connect-to-third-party-devices-and-supervisory-system/introduction-to-dnp3.html.

[20] COPA-DATA. *DNP3 (Distributed Network Protocol) e IEC 61850*. URL: https://www.copadata.com/it/industrie/energia-infrastrutture/energy-insights/dnp3-e-iec61850/.

[21] DLMS User Association. *DLMS/COSEM Official Website*. dlms.com. URL: https://dlms.com.

[22] Itron. *DLMS/COSEM Explained: The Backbone of Interoperability for Modern Utilities*. Itron Blog. URL: https://it.itron.com/it/w/dlms-cosem-explained-the-backbone-of-interoperability-for-modern-utilities.

[23] *DLMS/COSEM Developer Resource*. dlms.dev. URL: https://dlms.dev.

[24] Reallin. *Protocollo DLMS nell'applicazione di contatori di elettricità intelligente*. Reallin News, Apr. 2025. 2025. URL: https://it.reallin.net/news/dlms-protocol-in-smart-electricity-meters-appl-17350981893858304.html.

[25] DLMS User Association. *Security Framework in DLMS/COSEM: Ensuring Data Protection and System Integrity*. White Paper. 2024.

[26] DLMS User Association. *DLMS Commissioning: Addressing Integration Costs and Easing the Deployment of New Use-Cases*. White Paper. 2024.

[27] T.-H. Huang, C.-T. Chien, C.-L. Wang, and I.-E. Liao. "The Implementation of an HSM-Based Smart Meter for Supporting DLMS/COSEM Security Suite 1". In: in Proc. 8th Int. Conf. Internet of Things, Big Data and Security (IoTBDS 2023). 2023 (cit. on pp. 35, 37).

[28] S. Biswas, P. Das, H. Sharma, and K. Saha. "Efficient Data Transfer Mechanism for DLMS/COSEM Enabled Smart Energy Metering Platform". In: (2023). IEEE Transactions on Industrial Informatics (cit. on pp. 36, 37).

[29] Schneider Electric. *Specific implementation of DLMS: ION9000, ION8650, ION7400, ION8800*. Technical Reference.

[30] H. Dantas, E. Zekeriya, C. Doerr, R. Hallie, and G. van der Bij. *EFuzz: A fuzzer for DLMS/COSEM electricity meters*. 2014 (cit. on p. 36).

[31] H. Mendes, I. Medeiros, and N. F. Neves. *Validating and Securing DLMS/COSEM Implementations with the ValiDLMS Framework*. 2018 (cit. on p. 36).

[32] U.S. Department of Energy. *Benefits of Demand Response in Electricity Markets and Recommendations for Achieving Them*. Report to the United States Congress. 2006 (cit. on p. 38).

[33] M. A. Piette, G. Ghatikar, S. Kiliccote, E. Koch, D. Hennage, P. Palensky, and C. McParland. *Open Automated Demand Response Communications Specification (Version 1.0)*. Lawrence Berkeley National Laboratory, Rep. CEC-500-2009-063. 2009.

[34] B. Nordman, L. Parker, A. K. Prakash, and M. A. Piette. *Transforming Demand Response using OpenADR 3.0*. Lawrence Berkeley National Laboratory. 2024 (cit. on pp. 39, 41).

[35] T. Donitzky et al. *IoT for Smarter Energy Systems*. Intel Corporation White Paper.

[36] B. Nordman, L. Parker, A. K. Prakash, and M. A. Piette. *Transforming Demand Response using OpenADR 3.0*. ACEEE Summer Study on Energy Efficiency in Buildings, Lawrence Berkeley National Laboratory. 2024.

[37] M. A. Piette, G. Ghatikar, S. Kiliccote, E. Koch, D. Hennage, P. Palensky, and C. McParland. *Open Automated Demand Response Communications Specification (Version 1.0)*. Lawrence Berkeley National Laboratory, Rep. CEC-500-2009-063. 2009.

[38] GridFabric. *OpenADR Overview*. GridFabric Blog. URL: `https://www.gridfabric.io/blog/openadr-overview`.

[39] OpenADR Alliance. *OpenADR: In a Nutshell*. and "Cyber Security," OpenADR Alliance (cit. on p. 43).

[40] OpenADR Alliance. *The OpenADR Primer: An introduction to Automated Demand Response and the OpenADR Standard*. URL: https://www.openadr.org/.

[41] OpenADR Alliance. *OpenADR 3 to Matter Interworking Reference Specification*. Version 1.0. 2025 (cit. on p. 45).

[42] A. Catalano. *Sicurezza e Interoperabilità nei BEMS: Valutazione del Rischio e Scenari di Protezione tra VPN, Segregazione e BACnet/SC*. 2025.

[43] M. Nast, B. Butzin, F. Golatowski, and D. Timmermann. "Performance Analysis of a Secured BACnet/IP Network". In: in IEEE International Conference on Industrial Technology (ICIT), pp. 748-753. 2019 (cit. on p. 53).

[44] L. Catoni, C. Puliafito, and G. Dini. "TLS 1.3 for Fast and Secure Edge Service Continuity in Smart Cities". In: in Proceedings of the 11th International Conference on Smart Computing (SMARTCOMP '25), pp. 1-6. 2025 (cit. on p. 53).

[45] Q. Zhang et al. "BACnet or 'BADnet'? On the (In)Security of Implicitly Reserved Fields in BACnet". In: in Network and Distributed System Security (NDSS) Symposium, San Diego, CA, USA, Feb. 23-27. 2026 (cit. on pp. 48, 53).

[46] Siemens Industry, Inc. *BACnet Secure Connect: The next generation of OT security for building operations*. Key Considerations White Paper, Order no. 153-SBT-1403. 2021.

[47] D. Fisher, B. Isler, and M. Osborne. *BACnet Secure Connect: A Secure Infrastructure for Building Automation*. ASHRAE SSPC 135 IT Working Group, White Paper. 2019.

[48] H. M. Newman. "BACnet - The New Standard Protocol". In: (1997). Electrical Contractor, vol. 62, no. 9, pp. 62-66.

[49] ASHRAE. *ANSI/ASHRAE Standard 135-2020, BACnet - A Data Communication Protocol for Building Automation and Control Networks*. 2020.

[50] Optigo Networks. *Your Introduction to BACnet Secure Connect*. Jan. 2020. 2020. URL: https://www.optigo.net/your-introduction-bacnet-secure-connect/.

[51] Optigo Networks. *What is BACnet/SC?* 2019. URL: https://www.optigo.net/what-is-bacnetsc/.

[52] OWASP Foundation. *Threat Modeling*. OWASP Community; Accessed 2025. 2025. URL: https://owasp.org/www-community/Threat_Modeling (cit. on pp. 57–59, 76).

[53] L. Mella. *Threat modeling: cos'è e quali metodologie usare per l'identificazione delle minacce*. Cybersecurity360; Accessed 2025. 2025. URL: https://www.cybersecurity360.it/soluzioni-aziendali/threat-modeling-cose-e-quali-metodologie-usare-per-lidentificazione-delle-minacce/ (cit. on p. 60).

[54] Fortinet. *What is Threat Modeling?* Fortinet Cyberglossary; Accessed 2025. 2025. URL: https://www.fortinet.com/resources/cyberglossary/threat-modeling (cit. on pp. 58, 63).

[55] Threat Modeling Manifesto. *Threat Modeling Manifesto*. 2020; Accessed 2025. 2025. URL: https://www.threatmodelingmanifesto.org/.

[56] F. Valenza, E. Karafili, R. Vieira Steiner, and E. C. Lupu. "A hybrid threat model for smart systems". In: (2015) (cit. on pp. 65–67).

[57] V. Sanfilippo. "A Hybrid Model for Threat Analysis Enhanced by LLM: Integrating MITRE ATT&CK for Cyber-Physical System Security". Master's thesis, Politecnico di Torino. 2025 (cit. on pp. 70, 71, 73, 79, 80).

[58] D. Bringhenti, G. Marchetto, R. Sisto, and F. Valenza. "Automation for Network Security Configuration: State of the Art and Research Trends". In: *ACM Computing Surveys* 56.3, 57 (2023) (cit. on p. 64).

[59] D. Bringhenti, G. Marchetto, R. Sisto, F. Valenza, and J. Yusupov. "Automated Firewall Configuration in Virtual Networks". In: *IEEE Transactions on Dependable and Secure Computing* 20.2 (2023) (cit. on p. 64).

[60] D. Bringhenti, S. Bussa, R. Sisto, and F. Valenza. "A Two-Fold Traffic Flow Model for Network Security Management". In: *IEEE Transactions on Network and Service Management* 21.4 (2024) (cit. on p. 64).

[61] D. Bringhenti and F. Valenza. "GreenShield: Optimizing Firewall Configuration for Sustainable Networks". In: *IEEE Transactions on Network and Service Management* 21.6 (2024) (cit. on p. 64).

[62] D. Bringhenti and F. Valenza. "GreenShield-E2C: A sustainable energy-aware firewall configuration mechanism for edge-to-cloud continuum". In: *Computer Networks* 282, 112279 (2026) (cit. on p. 64).

[63] F. Pizzato, D. Bringhenti, R. Sisto, and F. Valenza. "Automatic and optimized firewall reconfiguration". In: *NOMS 2024 IEEE Network Operations and Management Symposium* (May 6–10, 2024). Seoul, Republic of Korea, 2024 (cit. on p. 64).

[64] D. Bringhenti, F. Pizzato, R. Sisto, and F. Valenza. "Autonomous Attack Mitigation Through Firewall Reconfiguration". In: *International Journal of Network Management* 35.1 (2025) (cit. on p. 64).

[65] D. Bringhenti, S. Bussa, R. Sisto, and F. Valenza. "Atomizing Firewall Policies for Anomaly Analysis and Resolution". In: *IEEE Transactions on Dependable and Secure Computing* 22.3 (2025) (cit. on p. 64).

[66] D. Bringhenti, R. Sisto, and F. Valenza. "Automating VPN Configuration in Computer Networks". In: *IEEE Transactions on Dependable and Secure Computing* 22.1 (2025) (cit. on p. 64).

[67] F. Pizzato, D. Bringhenti, R. Sisto, and F. Valenza. "Intent-Driven Network Isolation for the Cloud Computing Continuum". In: *Journal of Network and Systems Management* 34, 6 (2026) (cit. on p. 64).