Security On Smart Grid: Denial Of Service Attack Study on the PLC PRIME Standard

JORGE ERUSTES DE LA VEGA

uc3m | Universidad Carlos III de Madrid

Bachelor Thesis

Bachelor's Degree in Telecommunication Technologies Engineering
Universidad Carlos III Madrid

September 2018 – version 3.0

Jorge Erustes de la Vega: *Security On Smart Grid: Denial Of Service Attack Study on the PLC PRIME Standard* , Bachelor Thesis, © September 2018
SUPERVISOR:
Daniel Diaz-Sanchez

Monkey killing monkey killing monkey
Over pieces of the ground.
Silly monkeys give them thumbs,
They forge a blade,
And where there's one
They're bound to divide it,
Right in two.

— Manyard James Keenan,

Ohana means family.
Family means nobody gets left behind, or forgotten.

— Lilo & Stitch

ABSTRACT

The "smartization" of our world brings obvious benefits but it also conveys great risks and an intelligent network for electricity distribution, as Smart Grid poses, needs a solid security infrastructure. The Internet of Things has proven itself useful in many ways but at the same time questionable security-wise, and the PRIME implementation of a Smart Grid infrastructure is no exception.

The intrusion and malicious control of an electrical system could mean from the consumption reports being eavesdropped to perhaps the electrical service for an entire building being cut off.

Following the study of the PRIME standard I will prove the existence of exploitable vulnerabilities in its most widespread implementation. Focusing on a Denial of Service attack design, these weaknesses will be shown dangerous when the attack is detailed for a real-life scenario of the network.

With all this, I hope to shed light on some of the most critical points concerning this implementation of a technology with still potential for improvement as the Smart Grid itself.

RESUMEN

La "smartización"de nuestro mundo trae consigo beneficios obvios pero a su vez conlleva grandes riesgos y una red inteligente para distribución eléctrica, tal y como trae la Smart Grid, necesita una sólida infraestructura de seguridad. El Internet de las Cosas ha demostrado ser útil pero al mismo tiempo cuestionable en lo respectivo a la seguridad, y la implementación PRIME de Smart Grid no es una excepción.

Las intrusiones y el control malicioso de un sistema eléctrico podría suponer desde el espionaje de informes de consumo hasta el corte de suministro eléctrico de un edificio entero.

Siguiendo el estudio del estándar PRIME demostraré la existencia de vulnerabilidades explotables en su versión más extendida. Centrándome en el diseño de un ataque de Denegación de Servicio, estas debilidades se tornarán peligrosas cuando el ataque quede detallado en un escenario realista de la red

Con todo esto, espero arrojar luz sobre algunos de los puntos críticos relativos a esta implementación de una tecnología con margen de mejora como es Smart Grid.

We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.

— Donald E. Knuth [9]

ACKNOWLEDGMENTS

My sincere thanks to Dr. Daniel Diaz-Sanchez, this thesis' supervisor, for his valuable guidance on the direction of the project. Everything I have learnt along the course of this research could not have been possible without him, his advice and availability, even 2600 km away.

Immense thanks go to Marcos Álvarez Poblete for his priceless help with the design and creation of the figures in this thesis, keeping things a just a little bit more artistic.

Needless to say, forever grateful to my family and loved ones for their unconditional support along the way.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

NB-PLC Narrowband Power-Line Communications

PRIME PowerLine Intelligent Metering Evolution

DoS Denial of Service

ITU-T International Telecommunication Union - Telecommunication Standardization Sector

PHY Physical

MAC Media Access Control

FCC US' Federal Communications Commission

ARIB Japan's Association of Radio Industries and Businesses

CENELEC-A Comité Européen de Normalisation Electrotechnique A-Band

DPSK Differential Phase Shift Keying

OFDM Orthogonal Frequency Division Multiplexing

DLMS Device Language Message specification

COSEM Companion Specification for Energy Metering

AMI Advanced Metering Infrastructure

DSO Distribution System Operators

AES Advanced Encryption Standard

ECB Electronic CodeBook

EUI-48 Extended Unique Identifier (48 bits)

SNA Subnetwork Address

PNPDU Promotion Needed PDU

PNA Promotion Needed Address

GPDU General PDU

BPDU Beacon PDU

LNID Local Node Identifier

NID Node Identifier

LSID Local Switch Identifier

SID Switch Identifier

GPDU Generic MAC PDU

LCID Local Connection Identifier

MDMS Meter Data Management System

1 INTRODUCTION

This thesis is written with the goal of determining the security vulnerabilities of the PowerLine Intelligent Metering Evolution (PRIME) protocol when faced with a Denial of Service (DoS) type of attack and designing lines of attack needed to exploit the vulnerabilities found along the way. PRIME is a Narrowband Power-Line Communications (NB-PLC) technology, with currently more than 19 million deployed Smart Meters complying with it as of 2018, and likely to expand worldwide in the foreseeable future. The PRIME Standard is a creation of PRIME Alliance, led by the Spanish utility Iberdrola.

The total number of Smart Meters installed in Spain up to 2016's first semester reaches more than 17.5 million. This amount represents 62% of total existing meters at that point [1].

Another Spanish company, Gas Natural Fenosa, through its electricity distribution subsidiary Unión Fenosa Distribución, finished 2014 with more than 1.26 million domestic smart meters installed and integrated into the remote management system. This figure represents 35.2% of the company's domestic meter stock [1].

InovGrid (Portugal) is EDP Distribuicão's umbrella project for Smart Grids, pressing for increased energy efficiency. It has already installed 70k PLC PRIME meters within InovGrid's project scope. Targetted for 2015 was to reach an aggregated figure of 300k PLC PRIME meters [20] [11].

Enega Operator, based in Poland, ordered the installation of Smart Meters, with the deployment of 310,000 meters in progress. In Latin America, Neoenergia is "testing PRIME as part of an Advanced Metering field trial under real network conditions" [1] in Argentina and Brazil.

"The new PRIME Smart Meters are a key element in the roll-out of Smart Grids, which will open up new service possibilities to customers in the near future. Thanks to these, the distribution companies will be able to respond more quickly to customer requests for setting up, modifying and cancelling contracts, reconnections and readings" [1].

NB-PLC technologies are being widely employed in AMI's last mile thanks to the many advantages they present, such as the communications infrastructure being already deployed (electric cable), which means deployment costs can be seriously reduced. In fact, according to the Royal Decree 1110/2007 and the IET/290/2012, in Spain, 100% of all domestic electrical management equipment must be "smart" with telemetering capabilities by 2018. All of them use NB-PLC tech-

nologies: approximately 50% (around 14M) uses Meters&More (Endesa) and the other 50% uses PRIME (Iberdrola, Fenosa, EDP) [3].

1.1 MOTIVATION

Smart Cities are, without a doubt, a blooming concept with a vast number of ideas flourishing around it, where Smart Grid infrastructures conform a crucial part of that concept. Focusing on Smart Grid means focusing on a limited part of a greater technological vision but without which that vision could never come to play.

Ever since the birth of the Internet of Things, key in the development of tools to build Smart Cities, a long history of cyber-attacks has been existing alongside it due to the lack of security-oriented design and abundance of vulnerabilities. These innovations are progressively gaining weight and presenting themselves as firm candidates for a great impact in the future days to come and therefore, security needs to grow side by side with it to ensure this evolution.

Smart Meters and Concentrators are Critical Infrastructure [17] [7] in the electrical network system and an attack on them poses a serious threat to health, security and well-being of its citizens and correct functioning of its corresponding States. These vulnerabilities and harsh history make Smart Grid infrastructure a potential victim for ill-intentioned intruders [14] [6].

In order to bound the project, we will only focus the study on the PRIME standard for Smart Grid networks and the implementation of communication and logic mechanisms it brings to the table.

The PRIME Standard is the chosen one for several reasons. As commented before, it is heavily based in the Spanish and European context, hence presenting great closeness and propinquity to the development of this work. As well as geographical reasons, the maturity of its deployment in our environment is a positive point in its favor.

1.2 OBJECTIVES

As commented, the objectives of the project are three well-defined ones. First, it will focus on the study of the PRIME Standard and its inner workings along the most recent versions.

After we have reached a deep understanding of the protocol used for making the Smart Grid network work interconnected, we will go on to analyze the risks it presents for its users and ways in which it could be taken advantage of to cause clients harm.

The final step will be to precisely design the environment in which a Denial of Service attack could take place in an infrastructure running on the PRIME Standard. Here it will be shown the requirements

and most efficient plans to attack the network in the form of detailed attack designs.

1.3 STRUCTURE

The report has started with this introduction to the project, where we have presented a brief entry point to the PRIME Standard environment, motivations for the analysis and study into the cybersecurity aspect of the network protocol and communication.

Once we are comfortable with the scenario of the project, we will carry out an explanation on how the PRIME Standard works in all its aspects, including the elements present in the network, mechanisms provided for registration, communications and consumption report of its nodes, most important fields and security tools used in the protocol.

The third part will consist in the in-depth analysis of the standard from a security standpoint, intending to find every possible way an attacker could exploit any of the elements. With this, we will examine limits, fields and ways of communication in which the different elements have to send the number of messages needed for the correct functioning of the service.

Here, the vulnerabilities will be shown according to their purpose, first treating those usable by passively listening into the network and obtaining info about it. Later, we will get into active territory, expanding on those attacks requiring external network interference to work and its potential uses. Finally, those security holes mentioned before which best serve for a DoS attack will be developed for this purpose in further detail.

The last part will consist on the development of the tools and framework necessary to create the attacks which have greater potential to deny service in a Smart Grid infrastructure. This will include detailing the packet stream and context required for them to take place in pseudo-real life system of these characteristics. We will cover different effects for its various elements ranging from the Switch Nodes to the end clients.

1.4 LEGAL FRAMEWORK

One of the greatest obstacles in the way of Smart Grid's evolution and implementation is the creation of laws and energetic policies adapted to the changes it presents in the electrical network system. This technology presents the creation of new services and business models and require an adequate framework for the control of its development and the market surrounding it [4].

In the Spanish legislation, there is no specific law regulating the use of Smart Grid technologies' integration in the existing electric net-

work but some that control specific aspects of its growth and implementation. *Real decreto 1110/2007* ensures the definition of basic rights and duties for different contexts regarding supply metering, as well as quality of service control. Meanwhile *Orden ITC/3860/2007* means a replacement plan for old metering equipment, being substituted by machines capable of telemanagement and time-frame decision by December 31st 2018 [12] [21].

However, at european level, starting with the *2009/72/UE Directive*, the EU has provided principles, criteria and applicable techniques to privacy-enforcement and data protection in the European Union. Such measures include "data protection from design", Impact Assessment, "Best techniques available" standard and security measures to help "Smart Grid providers improve the infrastructures' cyber resilience" [5] [13].

These specific measures work alongside the General Data Protection Regulation (GDPR) framework, in place since May 2018, to ensure general protection and lack of undocumented use of user data.

1.5 SOCIO-ECONOMIC ENVIRONMENT

Talking about a project concerning electrical infrastructure, it is clear that a strong economic and environmental weight is being dealt with. Analyzing potential vulnerabilities in a Smart Grid can avoid the externally-forced malfunction of the system, hence avoiding serious damage and shortage of service. Furthermore, building a robust protocol ensures all optimization techniques being developed can be put in place securely, avoiding users overconsumption and thus pushing for a more intelligent use of the resources our environment provides.

A more optimal distribution of electric power is thought to provide a service that doesn't put so much pressure onto the working-class consumers and economically fragile communities and contribute to reducing the wealth gap. However, this service has to be kept shielded from wrongdoers to maintain an ethical use of the system just how it has been conceived, one reason more for this study to take place.

The risks conveyed in the misuse of this technology goes further than individual users hijacking and manipulating their consumption reports, against which the developers have built sufficient protection mechanisms. Some of the greatest threats come from the case in which an external malicious hand were to shut off electrical service for an entire community. That is another reason why this thesis' focus is oriented to protect against Denial of Service attacks.

1.6 BUDGET

This thesis presents cyber-attack study and design as its milestone and as such, requires no infrastructure to build or acquire tools that would contribute to the increase in the budget.

The economic value of the equipment used spans the price of the Mac Book Pro 2014 used in the writing and research of the thesis, estimated around 2,100 €. Its weighted price, considering a duration of approximately 9 months for the full development and planning of this work, together with a depreciation for computer equipment of 6 years (according to Spain's Tributary Agency [16]), would yield the following:

$$2100 * 9months / (6 * 12) = 262.5 €.$$

Regarding the professional cost, the estimated value would contain the following: ∼377 hours during the course of 8 months from a Junior Engineer, with a value of around 12 €per hour, and around 24 hours employed by a Senior Engineer (Thesis supervisor), estimated at 30 €per hour.

$$377h * 12€/h + 24h * 30€/h = 5244 €.$$

All of these values are referenced from this European benchmark [19] for the IT and Cyber security field.

In total, this would add up to a professional cost of the engineering labour of 5506.5 €.

2 STATE OF THE ART

The following chapters will present the vulnerabilities and attack procedures found in the study of the PRIME protocol, but first I will set the background and scope of the thesis as well as the standardized behavior under the PRIME Specification [8][15].

2.1 PRIME BACKGROUND

ITU-T accepted version 1.3.6 of the PRIME standard (Physical (PHY), Media Access Control (MAC) and Convergence layers) and this is the most common implementation in the majority of the Smart Grid equipment deployed as of today. Version 1.4 was released when 2014 came to a close, but its implementation is not as widespread as its predecessor.

A PRIME transmitter works as follows: PHY layer receives MAC layer's data, generating the PHY frame, also know as Physical Protocol Data Unit (PPDU). This frame is then convolutionally encoded, scrambled in all occasions and interleaved [10].

NEW ADDITIONS IN VERSION 1.4: V1.4 presents several changes, among which we can highlight a higher data rate (130kbps max in V1.3 –> 1028kbps max in V1.4) due to an extension to FCC and ARIB bands and a *"Robust mode"* through the use of a "Repetition coder" in the PHY layer, plus a longer *Beacon discovery* and an upgraded *Keep-Alive monitoring* in the MAC layer.

Furthermore, this latest version presents some security fixes, encrypting some of the messages regarding communication between Base and Terminal Nodes that, otherwise, travel in cleartext.

The new version of the standard, v1.4, presents too the inclusion of new the new FCC and ARIB band, opening its use for the US and Asia-Pacific markets.

Moreover, the new specification comes with the definition of several changes in the MAC layer. These include the addition of link quality information in the packet header, allowing to obtain such information in any given section of the network and further extending the diagnosis capabilities for connectivity troubleshooting.

As one could expect, this new version also comes with backwards compatibility mechanisms with version 1.3.6.

However, the dissemination of this latest version is not as broad and hence the decision to focus this thesis on the previous 1.3.6 version of PRIME. Just as we commented before, V1.3.6 is the current

and most widespread standard for the momentum-winning NB-PLC technologies' implementation

2.2 SCOPE

2.2.1 *Physical Layer*

The physical layer of an architecture deals with the transmission of bits among the devices present in the network. Attacks on these levels imply an attack on the type of modulation used to confuse or trick the system into understanding an wrong message in a communication.

2.2.2 *Media Access Control Layer*

The MAC layer is responsible for privacy, authentication and integrity of the flowing data in PRIME, and together with the combination of the two *Security Profiles* provided by the standard, it adjusts the encryption of the messages between nodes.

Security Profile 1 is used when encryption is needed, using the Advanced Encryption Standard (AES) 128-bits algorithm and Electronic CodeBook (ECB) block-cipher. Extended explanation on the usage of Security Profiles will happen later in the chapter, in Section 2.3.2.

A cryptanalytic work of the AES implementation goes out of the DoS-oriented focus of this thesis and therefore has no place in this study.

The analysis of the *key exchange* between subnetworks and use of *Master Keys* involves no injection of traffic and requires a mathematical dissection more than an strategy towards a Denial of Service.

As the MAC layer carries the security weight of the standard, its functioning and inner mechanisms will conform the core of my PRIME analysis, as I will further develop in Chapter 3.

2.2.3 *Application Layer*

At application layer, PRIME enables the use of DLMS/COSEM. Together, they define a communication protocol to transmit energy-related models and parameters. Additionally, DLMS comes with several security upgrades that provide the systems with tools to strengthen access control, message encryption and event logging.

However as interesting as this looks, this thesis will not go into detail regarding the vulnerability analysis of the application layer as its implementation is subject to change by any company who decides to build it into their systems.

2.3 PRIME STANDARD: DESCRIPTION

We now know PRIME is a NB-PLC technology whose aim is to upgrade the communication between the DSO and the last mile of AMIs (customers). The service aspires to provide application for the following:

- Improvement of quality control over the provided electrical coverage.
- Energy savings
- Billing optimization and enhancements.
- Remote control and measurements to the DSO.
- Control over distributed energy sources' generation, based mainly on the effect of several renewable energies providing electricity.
- Antifraud mechanisms and techniques.
- Non technical losses detection
- Demand response.
- House automation (Smart Homes)

The variety in which the network of nodes in a Smart Grid infrastructure can be set up over different landscapes and topologies, where little to no similarity can happen between a rural and an urban implementation. Wherever we may find a great density of nodes, element such as data concentrators are needed to handle the vast number of end-machines.

The communication between substations and management units are mostly based on pre-existing technologies (IP), which makes for both a more efficient use of infrastructure already in place and adaptation of existing security solutions.

PRIME consists of a physical (PHY), media access control (MAC) and convergence layers, and have been accepted as standard by the ITU-T in its 1.3.6 version.

2.3.1 *Network elements*

The PRIME specification describes that its Physical Layer (PHY) operates in the CENELEC-A band (41-89 KHz) using OFDM modulation. Carriers use DPSK, which enables up to 130 Kbps rates.

The Media Access Control (MAC) Layer consists of two different type of nodes operating: Service Node (Terminal) and Base Node (also referred to as Concentrator).

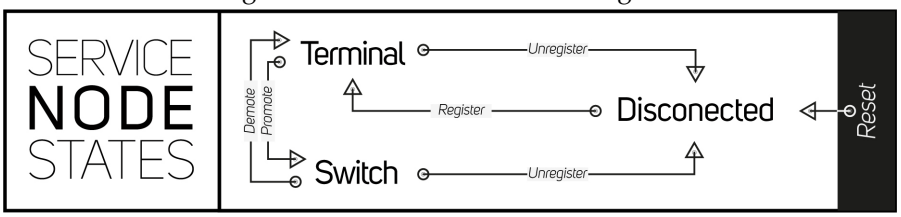
Service Nodes are mostly the Smart Meters themselves, the end device, but can also adopt the function of a Switch when conditions require it. As such, it assumes the two roles.

Switches act as repeaters, reducing the effect of signal attenuation.

On the other hand, *Base Nodes* deal with the management part of the networking and coordinates communications. They are the Concentrators, and there is only a single one of them per network.

Service Node, while operating, can go through three different states: Disconnected, Terminal and Switch. The interaction and evolution from one state to another can be seen in the following Figure 2.1.

Figure 2.1: Service Node state diagram



The PRIME Specification leaves a gap open when deciding the most favorable Node to promote up to Switch, not defining how the decision is best executed. External studies have taken place, concerned with the optimization of the path cost of information sent by the nodes and "enhancement of the algorithm's accomplishments" [2].

The goal clearly is to propose an algorithm in order to find the node providing for the optimal performance within the network, minimizing the time application layers exchange information.

Nodes count with several identification fields in PRIME, that serve diverse purposes of the addressing in each context, as shown in Table 2.1. Interaction between these fields will be seen in Figure 2.7, later in the Chapter.

MAC Layer frames contain several types of message, each encapsulating a different kind of information. As most modern communication protocols, counts with a Header indicating, among other things, the type of packet in question. Each type is described below in Table 2.2 and will be seen in greater detail in the upcoming Section 2.4.

At the same time, Generic Control Packets encapsulates several types of packet types, shown in Table 2.3 below and will be covered further in Section 2.4.1.

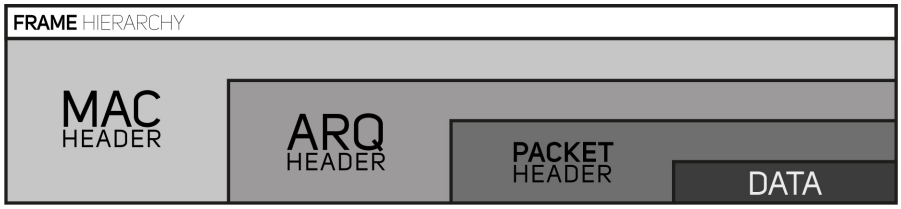
2.3.2 *Security Aspects*

MAC Layer is in charge of all aspects of PRIME dealing with confidentiality, authentication and integrity over the communications. For

NAME	LENGHT	DESCRIPTION / FUNCTION
Extended Unique Identifier	48 bits	MAC Address. As there is a single Base Node per network, its EUI-48 is used as a subnet identifier for the whole BN. This address is also known as the Subnetwork Address (SNA).
Switch Identifier	8 bits	SID identifies a Switch within the network. When a promotion occurs, the Base Node dynamically assigns the SID to the promoted Service Node. Base Nodes have SID = 0 by default.
Local Node Identifier	14 bits	LNID identifies a specific node in the network. Assigned by the Base Node upon registration in its subnetwork.
Local Connection Identifier	9 bits	LCID identifies a connection between a node pair.

Table 2.1: ID Fields in PRIME Nodes

Figure 2.2: Frame Hierarchy diagram



this purpose, the main tool is the use of what it is denoted as Security Profiles, where two different ones exist:

- *Security Profile 0* doesn't rely on encryption nor provides data integrity or private authentication.
- *Security Profile 1* is the most secure of the two, where we can find these three services. For encryption, this Profile uses Advanced Encryption Standard (AES) with 128-bit keys and Electronic Codebook (ECB) as block cypher. Regarding integrity, encryption of the desired data together with its Cyclic Redundancy Check (CRC) is the chosen option.

These two Security Profiles are negotiated between the parties involved, (Base and Service Nodes) when exchanging messages. However packets of the following types do not use Security Profile negotiation due to the nature of their respective use: PNPDU, BPDU, REG and SEC messages.

NAME	SENDER	USE
Beacon PDU (`HDR.HT=2`)	Base Node / Switch	Publish own address for Service Nodes to find it and register to. Broadcast functionality occurs through Beacons for synchronization purposes.
Promotion Needed PDU: (`HDR.HT=1`)	Service Node	Upon suffering from connection loss (no Beacon PDU received), used to alert nearby nodes of this situation. When other node (Terminal) receives a PNPDU, requests its Base Node for Switch promotion in order to provide the connectionless node with service again. As a Service Node sends a request for promotion triggered by a previous PNPDU, it attaches a PNA. This ensures the Base Node does not promote several Terminals to provide service for the same disconnected node.
Generic MAC PDU: (`HDR.HT=0`)	All nodes	Generic packets, consist of *Control packets* (`PKT.C=1`): network management, and *Data packets* (`PKT.C=0`): transmission of information.

Table 2.2: Packet Types in PRIME

Security Profile Negotiation takes place during device registration to the Base Node, where a REG_REQ message is sent from the Service Node with the proposed value for the profile. This value is contained in the `REG.SPC` field, and is considered by the receiving Base Node. It can then accept the requested Profile, in which case it will send a REG_RSP keeping that same `REG.SPC` value back to the Node, downgrade the Profile, setting `REG.SPC=0` or directly reject the connection if the proposal is not enough.

2.3.3 *Encryption*

Security Profile 1 is the designed context for encryption to take place, where for each packet encrypted in this mode the Secure CRC (SCRC) is computed. Computation works over the unencrypted packet payload, by dividing *modulo* 2 the generator polynomial $g(x) = x^8 + x^2 + x + 1$ and the polynomial x_8, then taking the remainder. This remainder is multiplied with the unencrypted payload to execute the operation.

NAME	PACKET CODE
Registration Management (REG)	PKT.CTYPE = 1
Connection Management (CON)	PKT.CTYPE = 2
Promotion Management (PRO)	PKT.CTYPE = 3
Beacon Slot Indication (BSI)	PKT.CTYPE = 4
Frame Structure Change (FRA)	PKT.CTYPE = 5
CFP Request (CFP)	PKT.CTYPE = 6
Keep-Alive (ALV)	PKT.CTYPE = 7
Multicast Management (MUL)	PKT.CTYPE = 8
PHY Robustness Management (PRM)	PKT.CTYPE = 9
Security Information (SEC)	PKT.CTYPE = 10

Table 2.3: Generic Packet Types

AES is the chosen mechanism to encrypt 128-bit blocks using a valid working key. The packet sees the subtraction of its Header and only the Clear packet payload, SCRC and "10..PAD" remain to act as "data to encrypt". This data then goes through AES encryption procedure.

2.3.4 *Key Hierarchy*

The PRIME Specification defines a series of keys it uses in its encryption processes, described below:

- *Unique Secret Key (USK):* Used to derive WKo and WK.
  USK = AES_{enc}(DSK, KDIV)
- *Initial Working Key (WKo):* Used by Service Node in disconnection state to decrypt some fields of REG_RSP message.
  WKo = AES_{enc}(USK,0)
- *Working Key (WK):* Used for encryption of all unicast data between Base and Service Nodes. Varies with each Service Node.
  WK = AES_{enc}(USK, SEC.RAN), SEC.RAN being the random sequence received in the field of the same name.
- *Subnetwork Working Key (SWK):* Used for broad/multicasting data or direct communications with the Base Node not involved. Never transmitted over physical channel, computed from other keys.
  SWK = AES_{enc}(SNK, SEC.SNK), SEC.SNK being the random sequence received in the field of the same name.
  WK and SKW updated each *MACRandSeqChgTime* seconds.

- *Master Keys 1 & 2 (MK1, MK2):* Administered by the Base Node and used to derive other keys. Administration of these keys is not defined in the PRIME Standard.
- *Device Secret Key (DSK):* Unique to each Service Node, hard-coded in the device at production time.
  DSK = AES_{enc}(MK1, UI), UI being the EUI-48 of the device.
- *Key Diversifier(KDIV):* Unique to each Service Node but not constant for the entire lifetime of the device. Provision of the key to the Node is not of the scope of the PRIME Standard.
  KDIV = AES_{enc}(MK2, UI), UI being the EUI-48 of the device.

2.4 MAC PDU FORMAT

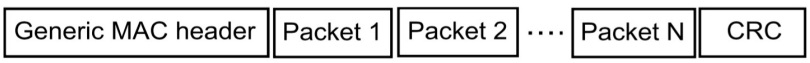
This section will be dedicated to the explanation of the structure of the MAC PDU and some of its most relevant types in this work in order to ease the explanation of the attack implementations to come in Chapter 4.

2.4.1 *Generic MAC PDU format*

Different types of MAC PDUs serve different purposes, but most of the Subnetwork traffic in a PRIME network consists of Generic MAC PDU (GPDU)s. All Control packets and Data traffic are comprised in GPDUs.

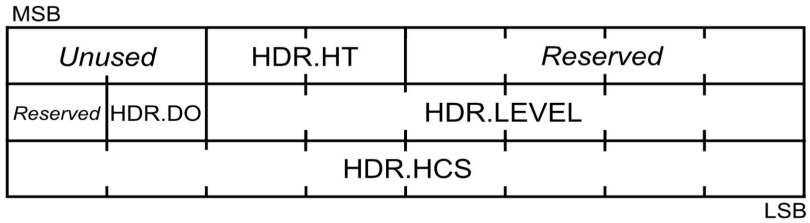
Figure 2.3 shows how the GPDU is "composed of a Generic MAC Header followed by one or more MAC packets and 32 bit CRC appended at the end" [8].

Figure 2.3: Generic MAC PDU Format



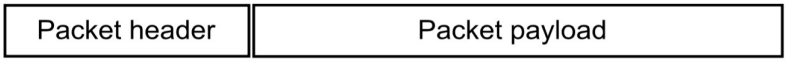
Consists of a size of 3 bytes, its format is represented in Figure 2.4.

Figure 2.4: Generic MAC Header



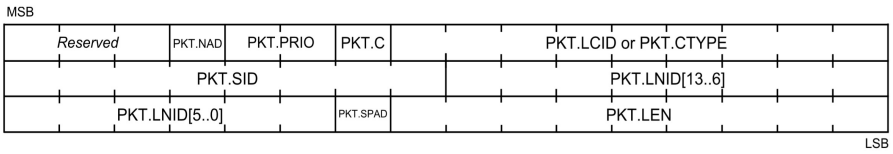
Inside one of the multiple packets a GPDU can carry after its header, the structure that can be found is what it can be seen in Figure 2.5.

Figure 2.5: Packet structure



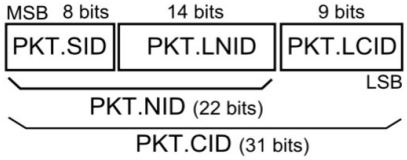
MAC Packet's header has a length of 6 bytes and it is composed of the elements seen in Figure 2.6

Figure 2.6: MAC Packet header



In the packet fields we can find two remarkable values: `PKT.LNID` (14 bits) and `PKT.SID` (8 bits). These two values, together, conform what is known as the NID of a Node, `PKT.NID` in the header. The figure also contains a reference to the field `PKT.LCID` (9 bits), Local Connection Identifier (LCID), which also composes the `PKT.CID` when joined with the NID. The whole scheme is shown in Figure 2.7.

Figure 2.7: PKT.CID structure

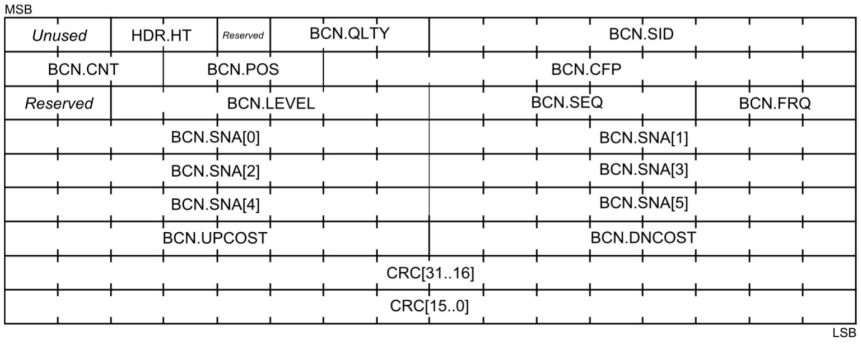


2.4.2 *Beacon MAC PDU format*

Beacons are a mechanism of the standard meant for broadcasting Switch and Base Node's information to potential connecting nodes around, working through the use of the Beacon PDU (BPDU). Its function is "to circulate information on MAC frame structure and therefore channel access to all devices that are part of this Subnetwork" [8].

The transmission of the BPDU is periodic (every (*MACFrameLength - MACBeaconLength*) symbols), and can be used by Service Nodes as a synchronization tool. Beacons always have a length of *MACBeaconLength* symbols long [8]. Its complete structure can be seen in Figure 2.8.

Figure 2.8: Beacon PDU Packet



Another function the Beacon PDU is given is providing Terminals with capability to detect if their corresponding Uplink Switch has stopped being available, be it because external change in the conditions of the medium or directly device failure. Switch unavailability is activated if a Service Node does not receive $N_{\text{miss-beacon}}$ BPDUs in a row, when the link unusable. In doing so, all logical relationships with that Switch are completely shut off: closes all active MAC connections, unregisters from it and even stops sending BPDUs downwards if it was acting as a Switch itself.

2.4.3 *MAC CRC*

For both the GPDU and BPDU, the last field in the packet is a 32-bit CRC, used to detect transmission errors as usual. Its computation involves the complete PDU except for the CRC value itself.

The CRC operation consists of several elements, similarly to the SCRC seen in the previous subsection. The first of which would be the Generator Polynomial: $G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$. Then, the input polynomial's coefficients are derived from the data being checked. The data bits themselves conform the coefficients in reading order, i.e. highest coefficient is assigned to the first read bit.

Finally, the remainder $R(x)$ is computed, and its coefficients will serve as the final CRC value. Such remainder is obtained from the division of $(M(x) * x^{32}) / G(x)$.

2.4.3.1 *Header Check Sequence*

This is a field appearing at the end of GPDUs, whose calculation involve the CRC of *SNA* || *first 2Bytes of Header* (|| meaning concatenation). This CRC is computed in a similar way to the Secure CRC (SCRC) commented in the previous section: "remainder of the division *modulo* 2 of the polynomial $M(x)*x^8$ by the generator polyno-

mial textitg(x) = $x^8 + x^2 + x + 1$" [8]. In this operation, *M(x)* acts as input and it is precisely equal to *SNA || first 2Bytes of Header*. The MSB of this sequence will correspond to the highest coefficient in this input.

3

PROTOCOL VULNERABILITY ANALYSIS

Before any action to be taken towards a Smart Grid network, it is needed to thoughtfully analyze the protocol [8] used by this system and carry out an extensive planning on how could we attack PRIME [15].

First, we will study what are the possibilities to passively work on the inspection of the protocol, listening to the network's traffic and determining what information can be yielded by the nodes belonging to a Smart Grid system.

After that, we will start to take action via an active attack of traffic injection, through which we can effectively intervene on the network. Here we can evaluate what is the nature of the packets that are in our best interest to be introduced to obtain the greatest amount of information possible back from the system in question.

Finally, based on a more extensive knowledge of PRIME we will be in position to understand what aspects of it are more prone to suffering a Denial of Service attack (DoS) over this infrastructure.

3.1 PASSIVE ATTACK: RECONNAISSANCE

As commented before, first our objective is to go over the possibilities of carrying out a passive attack in which we sniff into the network's traffic. In this phase we will not yet consider the option of inserting packets in the system but just passive listening. The objective of this eavesdropping is to determine the biggest amount of information about the network, learning the number of nodes present, each of their types and how much we can learn about them separately by the data they transmit.

3.1.1 *Physical Layer*

It is commented before how the physical layer is in charge of bit-level transmission between machines and their physical interfaces. Unfortunately, the only attack considered in this thesis focusing on this first layer is physical layer jamming and cannot be considered as of passive nature. Therefore, this layer presents no interest from the point of view of this section but will be studied later.

3.1.2 *Media Access Control Layer*

SNIFFING: By the connection of a sniffer device in one of the elements of the network using Security Profile 0, an attacker could be able to gain access to all traffic flowing through it. This means that strategically, someone who wished to gain access to the biggest amount of critical information would aim for a heavily connected piece of equipment like a Base Node.

If we connected a sniffer on a topologically inferior element, only the information regarding the very last few Service Nodes after that would be readable by us. Hence a Base Node stands as a more interesting alternative.

Once connected, if the upper-layer traffic is unencrypted, detailed analysis could be carried out onto the data transmitted to find mostly two types of information:

- NID information about each node and its associated EUI-48 through the study of connection requests to the Base Node under attack.
- Transmission of critical information, such as consumption reports and other private data.

This whole scenario depends on the use of an specific configuration regarding the Security Profiles provided by PRIME. However, regardless of which one is used, they all lack Perfect Forward Secrecy. This means that if a strong attack were to be launched upon the keys for them to be compromised and succeeded, it would reveal past information captured earlier (encrypted at the time). Documented in [18] we can find some past successful attempts on key-compromising attacks at Service Nodes. It will not be covered here as it is out of the scope of this thesis.

Rather than an active intrusion in the network traffic, the sniffing technique opts for a more stealthy alternative, focusing on the capture of flowing information, harming the privacy of the service's users.

3.2 ACTIVE ATTACK: TRAFFIC INJECTION

3.2.1 *Physical Layer*

As it is common with communication protocols of diverse natures, the physical layer is tasked with the transmission of information at bit level, connecting the physical elements in the devices belonging to the network (i.e. interfaces).

This physical layer works with the symbol transmission through the network. Here, the most common type of attack is that of block-

ing or interfering existing communications, known as *jamming*. This jamming consists on transmitting signals onto the channel to disrupt the message transmission by decreasing the Signal-to-Noise ratio.

PRIME uses OFDM-PSK as a modulation technique and, for it, a jamming attack can be carried out using different type of signals to interfere in the communication:

NOISE JAMMING: is the name give to the introduction of noise in the system and conforms the simplest option for signal jamming. Generally, Gaussian noise is employed and it can be used to cover the bandwidth totally or just partially.

INTERFERENCE JAMMING: is now based on the usage of colored signals not in synchronization to the signal we are to target.

CORRELATED JAMMING: is the most effective against the specific type of modulation used by the standard but requires synchronization and profound knowledge of the target signal.

Apart from *correlation jamming* – which, on the other hand, presents higher demands to be done effectively – the OFDM modulation used in PRIME is robust against noise, which obviously reduces the success expectancy of a *noise* or *interference jamming*.

An attack on Service Nodes would not be the most effective as the dynamic topology of the network in PRIME allows for its recovery when a number of switches and Service Nodes are taken down, whereas an attack on a Base Node presents a very different scenario. If one of these nodes is affected by a successful jamming attack, the network would be put at serious risk.

Regardless of the method employed, a jamming attack would require physical access to the channel and even though a protection of the Base Node usually takes place in PRIME real-life implementations, protecting Service Nodes is a tougher challenge. They are usually the previous piece to the client's power station and protecting every link with the Service node requires a major economic investment.

This is the reason why jamming attacks are more interesting from a wireless point-of-view, as the air being used as the transmission channel makes the "physical access" to the communication infrastructure trivial. However, we will leave jamming attacks to the side when executing an attack and focus more on packet injection to the MAC Layer after an intensive passive sniffing of the ongoing traffic and enough information has been extracted.

3.2.2 *Media Access Control Layer*

As we have seen, PRIME MAC frames are divided into three types: Beacon PDU, Promotion Needed PDU and Generic MAC frames; depending on what does the MAC Header determine.

PROMOTION NEEDED FLOODING: Promotion needed PDU is a packet meant for when a Service Node acting as a Terminal wants to move up and become a Switch, so then it uses this PNPDU to alert nearby nodes of this intention. When others Terminals in its surroundings receive this frame, they alert the Base Node to request a Switch promotion: *Promotion Request* message which includes the Promotion Needed Address (PNA) to identify the requesting node. This mechanism also prevents the Base Node from processing two promotions coming originally from the same Terminal Node.

However, the standard does not include any tool or criteria to decide which node should be promoted, which could lead to problems for the system.

One could modify a Promotion Request message changing its PNA, which would lead to confusion in the network as the Base Node would promote a different node than the one who requested such promotion. Furthermore, if the network were subjected to injection of a forged Promotion Request when another request is generated by a real node, this could result in a competition between several Service Nodes. Such situation could lead to a shortage of service. In this strategy, an attacker could spoof PNPDUs from forged EUI-48 addresses to generate negative results:

PNPDUs are meant for Nodes with poor communication with the Base node and hence no encryption can take place between them, leaving these messages unencrypted in all Security Profiles. This makes for an interesting vulnerability as it spans all networks.

The number of PNPDUs flowing in the system is recommended to be time-limited to avoid total flood, but a dense traffic of this type would severely affect legitimate requests of real nodes experiencing connectivity problems. The potential effects and dangers of this attack will be further explored in the next part, Section 3.3, as they might cause a severe shortage of service.

LSID OVERFLOW: Again, this approach exploits the relay system in PRIME for nodes lacking quality connection with the Base Node. Now, an attacker could set up multiple rogue Service Nodes and craft promotion requests to Switch in the topology.

The maximum number of Switches in the network is defined by its identification with the Base, for which the Local Switch Identifier (LSID) is used. It is an 8-bit value, thus conforming up to 256 different possibilities for Switch identification.

When this limit number of Switches is reached in the network, it is up to the Base Node to decide wether to block new requests

or start assigning new positions getting rid of the oldest entries in the registry. For the Promotion Requests to go though as valid, the PNAs must differ from those used previously to trick the Base Node into taking them in as new Switches, what leads to great noise in the network for the attack to take place.

REGISTRATION OVERFLOW: When a new node wants to register onto a network, they establish a communication with its corresponding Base Node. Here, they use their EUI-48 address as identification to obtain what is called a *Node Identifier* (NID), unique in the network. This NID is a 22-bit identification value made up of two parts: Switch ID (SID, 8 bits) and Local Node Identifier (LNID, 14 bits), as previously seen in Figure 2.7.

It is worth mentioning that this LNID is referred to a Switch so it is unique for each node connected to it. With all this in mind, the maximum number of identifiers goes only as high as 16384 IDs for a single switch (nodes potentially registered).

This registration poses a risk for the network, as an attacker could forge this Registration Requests by selecting an SID to use in the connection (this being an alternative option, rather than letting the switch select it). This might be the preferred alternative as in real networks, the number of switches is generally limited, allowing us to narrow down the attack.

If the attacker spoofs these requests using EUI-48 IDs, in pseudo-random fashion, it could end up exhausting the LNID space for valid identifiers, focusing on the most critical switches in the network (i.e. targeting their corresponding SIDs).

Under this situation, should a real working node get disconnected from the network or suffer from a loss of connection, it would find itself unable to become part of its former topology again. There would be no available LNID for it to use with that Switch until the timeout has run out. This attack presents serious problems for a working network and will be analyzed in further detail in the next section, 3.3.

NODE REGISTRATION SPOOFING: When a node wants to connect to a network, a registration request is sent with its EUI-48. However if that node happens to be already registered and its Identifier, the standard presents no guideline on what to do.

Clearly, in this case, only two outcomes can be expected: rejection or reassignment. In the first case, not until that registered node is unregistered and its EUI is out of the Base Node's list will the requesting Service Node be able to enter. Alternatively, the Base Node could decide to accept that request and assign a new NID (there cannot be reused NIDs until timeout).

This, of course, presents several ways of exploitation. (1) Rejection: As an attacker, one could spoof the victim node's EUI-48, a Service Node, and register with it in the network so when the legitimate node attempted to do the same, it would not be allowed access.

(2) Acceptance with new NID: As seen before, spoofing a EUI-48 address and request registration with it. This Identification would belong to an existing and working Service Node, and when the attacker creates this (second) request for the same EUI, the corresponding Base Node would assign it a new NID to which it will send all further messages. Of course, the attacker would carry out the request through a different SID so the victim is no longer sent the response and future messages This attack would leave a legitimate Service Node with no connection to the network if its identification is spoofed, causing Denial of Service in the long run for the system.

BASE NODE SPOOFING: According to the specification, there can only be ONE single Base Node per network, so any interfering Base Node Beacons would severely disturb the behavior of a functioning network.

If an attacker were to gain physical access to the infrastructure at play, could connect a rogue Base Node so nearby Service Nodes would receive its Beacons and potentially request connection to it. Once they are connected to the attacker's BN, Service Nodes would stop receiving real data and lose connection to the outside network.

According to the standard, it is the Service Node who is in charge of communicating its supported Security Profiles when establishing the connection to the Base Node. Then, the BN either accepts that Profile or decides to downgrade it to an unprotected context. If the chosen is Security Profile 0 it means the Base Node has been able to establish a connection in low-protection conditions even if the Service Node supported a protected Profile 1, effectively hijacking its function. The use of this method will be discussed in the next section, 3.3.1

DISCONNECTION SPOOFING: Getting one node element off the network works in a way similar to the mechanism previously explained in the previous attack. Now we will work with demotion of switches, unregistration and finally disconnection.

These actions can have a variety of origins. A Service Node can initiate the process through a request message (*REG_UNR_S* / *PRO_DEM_S* / *CON_CLS_S*; unregistration, demotion and disconnection respectively), and is answered through an acknowl-

edge. Alternatively, a Base Node can take over the initiation of the process (*REG_UNR_B* / *PRO_DEM_B* / *CON_CLS_B*).

Sniffing here can be a useful tool to find suitable targets (LNID/ SID/ LCID) and later spoof a message from the Base Node requesting Unreg/Demot/Discon towards that target. That Service Node will then, as explained before, acknowledge the request. However this ACK would be received by a Base Node who hasn't really sent any request in the first place, but anyhow interprets that SN's response serves as a proper request, for which it will then proceed to send a message to be acknowledged again by the Service Node. As expected by a now, unwanted, message through the SN's eyes, it will be discarded.

This process results in an action being finally acknowledged and executed by both parties involved while none really requesting it. The misunderstanding between devices would unavoidably lead to an unregistration, demotion or disconnection of a Service Node, causing potential DoS. Further implications will be discussed in the next section.

REPLAY ATTACK: This mechanism conforms one of the classic attacks on network infrastructures and consists on the manipulation of traffic to be resent and repeat the action that data executed. A previous and necessary step for it to happen is the examination (eavesdropping) of a channel until valuable and usable traffic comes through and has the potential to be used for the attack.

In the PRIME Standard, a replay attack can take place when using Security Profile 0 and facing a lack of anti-replay mechanisms in the upper layers. Captured traffic then will have to go through a change in its Local Connection Identifier (LCID) to the desired one for it to be used (assuming an already working connection).

Through this mechanism, we can replay traffic originally meant for a Service Node in another. We would need to change the LNID, SID (and LCID) for it to be accepted by the victim node.

The use of Security Profile 1 in this situation would toughen the attacker's job by making cryptoanalysis necessary to dilucidate wether a packet is interesting or not. However, headers are not encrypted nor integrity-protected by Security Profile 1. This means header fields (for instance, LCID) can be modified at will, but CRC needs to be recomputed for every change made to the packet. This security profile poses severe problems for anyone attempting an attack of this sort, as there's another restriction to be faced: WK are device and session specific, meaning that only traffic from the same device in the same time span (while

registered and using that WK) can be replayed. As seen, Security Profile 1 noticeably reduces the attack's scope for one with the aforementioned characteristics.

SECURITY CRC: Integrity protection is provided by this SCRC in the Security Profile 1. It uses the unencrypted packet payload for it to be computed and gets encrypted for transmission, whereas standard CRC comes unencrypted.

For a receiver to consume a packet, first has to check the CRC in the header, decrypt and lastly check the SCRC. An attacker can make use of this by attaching an encrypted payload for which its key is unknown to a valid header and CRC. As a result, every receiving node has to check CRC and decrypt a packet before it can detect its content is forged, facing intense use of resources in the process, while an attacker having very low computational power needed.

The attack scope and effect depends on the victim, as a Service Node suffering from resource exhaustion would leave just a client disconnected and off the network, but a Base Node under attack would mean a loss of service for the whole network while kept busy enough.

Modern PRIME chipsets now come with AES crypto modules that ease the workload on the main node processor by precomputation of the CRCs, resulting on a lower impact, leaving the attack few room for damage.

3.3 ACTIVE ATTACK: DENIAL OF SERVICE

3.3.1 *MAC Layer*

We have seen the Media Access Control Layer dealing with the logical heavyweight action of the protocol, including addressing, Switch and Node registration management, channel access, security and the logical topology itself. Aforementioned attacks have been studied for this layers where it shows potential vulnerabilities and now it is time to plan how these could be exploited to achieve the desired Denial of Service in the system.

PROMOTION NEEDED FLOODING: Abuse of the PNPDU mechanism can be done by spoofing of the requests. When a Base Node receives multiple Promotion Requests from nodes apparently lacking connectivity, it makes use of its tools to prevent it. This means, the BN falls into the trap of enabling all those node to become Switches when, in reality, their situation was completely normal.

Through this attack, a malicious operator could downgrade the communication quality with bigger delays, operation errors... The communication would not be completely blocked but noticeably hindered.

Any of the Security Profiles provides encryption protection to PNPDU messages, so every network running PRIME is subjected to this potential harm.

For each PNPDU generated by a requesting node, several are forwarded to the closest Base Nodes by other nodes neighboring the requester. This leads to an *amplification* of the traffic, which, if used in a spoofing attack could lead to a traffic overload to that Base Node.

To solve a connectivity problem, the Base Node would grant a node the Switch state. However, if this Base Node receives Promotion Requests from apparently many, if not all, nodes in the network it could result in an enormous growth in the topological complexity of the net.

Promotion Needed requests always include the address of the node to be promoted, should the Base Node agree to do so. This is referred to as PNA in order for the Base Node to manage and potentially evaluate requests. This attack means that the Base Node will be confronted with the majority of its associated nodes apparently lacking quality connectivity and likely accepting most requests to promote them into Switches.

No way is available to avoid traffic amplification as described previously, only a limit in the number of PNPDUs received in a time span. This offers limited protection; an inadequate interval would worsen the functioning of the network affecting legitimate nodes too. On the other hand, some protective measures can be taken against Switch Promotion requests' dramatical increase in number. Base Node firmware modifications can be executed to select the PNAs capable of being promoted and ignoring those request coming from other addresses already registered.

Putting these facts to practice would result in a noisy network, PNPDU packets being more and more frequent until the network faces congestion, and a growing number of Terminal Nodes turning into Switches, overcomplicating the logical topology of the communications. As a result this will be one of the attack mechanisms to consider in the next part for the implementation of attack designs, Chapter 4.

NODE REGISTRATION OVERFLOW: This attack would occur in the scenario of a wide group of malicious nodes requested connec-

tion to a Base Node and ended up exhausting the registration space available for legitimate ones.

Denial of Service takes place when real working nodes attempt to register onto one of these Base Nodes and find themselves unable to do so for the LNID table in those BN is already full and is forced to leave them out. The most obvious and interesting objective for the attack would be to interfere with the most critical elements in the topology. In order to maintain ease of interaction with nodes connecting with the first time or newcomers to the network, PRIME specification keeps the registration messages unencrypted in all Security Profiles. However, if the intention is, as the case with our malicious nodes is, to complete such registration, it will be needed to comply with the Profile being used in that segment of the network. Clearly, a much easier task if SecProf 0 is being used.

It is worth mentioning the noisiness of this attack, as the generated traffic in order to exhaust the registration table entries has to be relatively massive. This adds to the fact that PRIME networks operate over a dynamic logic and nodes are rarely added to an already working network. Therefore, systems running a Network Intrusion Detection System (NIDS) could have a higher probability of successful defense against this situation.

If this attack were to take place, and a legitimate node attempted to connect to the affected switch would not be able to do so for, at least, an specified timeout established in the standard.

LNID TIMEOUT: The specification dictates that when a Base Node needs to use a LNID, it shall not choose a value for the LNID recently released by an unregistration process. The exact time needed to go by before the use of that value would be the following: *(macMaxCtlReTx +1) * macCtlReTxTimer* seconds [8].

macMaxCtlReTx corresponds to the "maximum number of times a MAC entity will try to retransmit an unacknowledged MAC packet".
macCtlReTxTimer corresponds to "the number of seconds for which a MAC entity waits for acknowledgement of receipt of the MAC packet from its peer entity" [15].

The goal of this mechanism is to make sure retransmission packets have safely left the subnetwork and will not be mistakenly redirected to the new node, using that unregistered LNID.

In a similar manner, a Base Node shall not reuse a recently freed LNID. Freeing happens through the Keep-Alive process and reuse needs to wait until T_{keep_alive} seconds have passed.

For this, the standard uses the largest value between the last acknowledged T_{keep_alive} value and the last unacknowledged T_{keep_alive}, for the Service Node receiving that LNID value.

With all of this in mind, we can now see how under this attack, the network would not be able to assign a new LNID if a node were to disconnect or lose connectivity. Free LNID values would not be available under the affected SIDs for, at least, this timeout.

However, the behavior of the Base Node regarding this exhaustion of LNIDs for a switch is not specified, so the actions it took would depend on the specific implementation.

NODE REGISTRATION SPOOFING: PRIME specifications fails to determine how to handle registration requests from already registered IDs, and it can either reject the registration or accept it, assigning a new NID.

It was commented before the situations to which each behavior leads to regarding security concerns and ways in which both can be exploited with malicious purposes. Although registration rejection can be undesirable (registering all LNIDs and leaving legitimate nodes out at registration time), it has not nearly the potential danger conveyed in the 'node impersonation' that the second behavior conforms, as it would mean the rogue node will act and receive communication in the name of the victim node.

In order to chose a victim and obtain its EUI-48, a simple sniffing would be enough to learn this kind of data as it travels unencrypted to a listener sitting in the middle of a registration request communication, as we do not count on physical access to the nodes to obtain it.

We have seen in the case above that when a node stands in the place of a legitimate node previously there, it has to keep up with the security specifications established with the Base Node until the registration process is complete.

LSID OVERFLOW: The Local Switch Identifier Overflow consisted on creating functioning Switches in the network. Through this behavior, an attacker would attempt to take over existing switches and reduce the introduction of new ones by creating fake ones that overflow the LSIDs in the network. Clearly this creates connectivity problems for those Terminals connected to the affected Switches.

Take control of old or existing switches would probably mean its replacement with spurious or fake switches, leaving those Service Nodes depending on those connectivity-less and isolated from the system by the attack. To register into the network

for this mechanism to take place, we again need to comply with the security specs of the Base Node.

BASE NODE SPOOFING: Theoretically, only one Base Node can operate in the network, so the appearance of another will distort the normal behavior of the system due to nodes now requesting connection to it too and the creation of unconsidered situations by the standard.

Through this attack, one could perform a DoS against the Service Nodes belonging to a network by letting them know the existence of a legitimate-looking Base Node so they would connect to it and the posterior loss of connection to real stimuli.

In this context, the victims cannot be chosen, as through this mechanism an attacker just ensures Beacons are sent. Even though we haven't been able to find a way of 'forcing' selected Nodes to connect, in Chapter 4 we show the best approach we could to achieve this effect (Section 4.3).

The only variation that could be attempted is the placement of the Base Node. If one were to place it closer to the desired target Nodes, it could coarsely affect the accuracy of the attack.

When the attack is carried out, downgrade of the connection between SN and BN can occur, to a Security Profile 0 even if the Service Node supports Profile 1. In this situation, where the rogue element operates as a totally trusted Base Node, it can request crucial information from the rest of the network. Examples of this could be *consumption reports* or even *power shutdown*, meaning a DoS attack has been successfully executed.

DISCONNECTION SPOOFING: This attack tackles the demotion, disconnection or unregistration mechanism of the switches in PRIME, which works in a similar way and can be initiated by either the Base Node or a Service Node. In a nutshell, an attacker would try to demote, unregister or disconnect a number nodes in the network, which would leave them without service.

The effect of the attack would vary according to the action being taken and so would its impact in the network being, in general, demotion the most destructive.

- If demotion took place, all the nodes connected through a same affected Switch would suffer from this DoS.
- In the case of an unregistration of a Service Node, the effect would only apply to that specific one, denying it service and connection.
- Undergoing disconnection could suffer unregistration of a Switch and lack of service in a connection.

Unregistered nodes have mechanisms to connect back into their former network, making this attack cause a temporary effect per se. Therefore, from an strategic standpoint, it is more interesting to make this serve more as a tool for a greater picture, a mechanism to temporarily create a situation from which another mechanism or attack can benefit from. This idea will be further developed in the next chapter's Section 4.4.

4 ATTACK DESIGN

The goal of this work is, not only to study potential vulnerabilities found in the PRIME Standard, but to further analyze those interesting from a Denial of Service standpoint. Moreover, we will attempt to design a theoretical framework in which these vulnerabilities could be exploited for a real-world attack with moderate chances of success.

4.1 ENVIRONMENT AND INFRASTRUCTURE

CYBERPHYSICAL INFRASTRUCTURE: First we will define the structure and setup which would be used when one of these attacks were to take place.

It would consist of the mechanism needed to inject traffic into the last mile of the Smart Grid system, in between the Smart Nodes and the Data Concentrator, or even right into the Smart Meter itself. In here, our injection can affect Base and Service Nodes contained in the Smart Meter Panel and hence its potential to affect the network is widened.

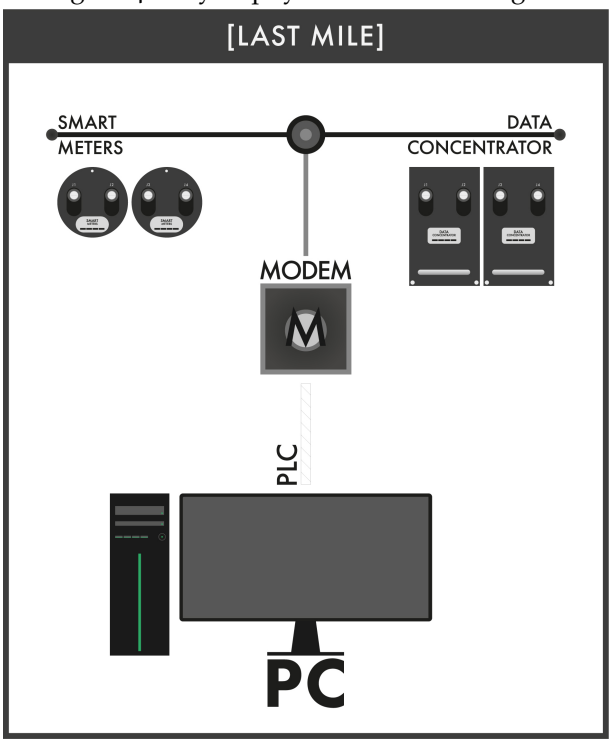
The mechanism in charge of this injection would consist of a human-operated host (PC) connected via PLC to a modem (μ), in charge of the connection between our interfering system and the victim network. A complete view of how the structure would fit in the Last Mile of the Smart Metering system can be seen in Figure 4.1.

On the other hand, Universidad Carlos III de Madrid counts with the following setup, which this thesis has used as a reference (Figure 4.2).

The system consists on real and virtual equipment UC3M research group owns, used to emulate a fully-working electrical metering network infrastructure running the PRIME Standard.

Four physical Smart Meters (ZIV manufacture) and a Data Concentrator conform the main Panel connected with the outside. The connection runs all the way from the PRIME Panel to an instance of the commercial Meter Data Management System (MDMS) IRIS from STM Telecontrol. This system is tasked with the addition of virtual devices to the physical infrastructure of the environment, and allows for the management of up to 11 Data Concentrators and 1593 Smart Meters. A PRIME Sniffer (PRIME MANAGER from ZIV) is in charge of protocol and traffic management, running connected to the electrical network side by side to an Wireshark application connected to the Ethernet network.

Figure 4.1: Cyberphysical Structure diagram



For network performance simulations on a PRIME system, a co-simulation framework has been used combining Matlab and OMNeT++ ((Varga, 2008), a modular simulation environment based in C++), just as other projects of the same nature have employed to model the effects of PHY and upper layers (MAC and Logical Link Control (LLC)) [15] [2]. Application layer uses a payload modeling, implementing DLMS/COSEM to complement the PRIME Specification.

The goal of this is to create simulated topologies emulating a general European low-voltage network.

4.2 ATTACK 1: NODE REGISTRATION OVERFLOW

This attack depends highly on how the standard is implemented, hence it will be detailed here according to the PRIME Specification [8].

Upon registration with the corresponding Base Node, a Terminal is assigned a Node Identifier (NID). This NID is conformed by the the SID and LNID, referring back to its corresponding Switch. The number of NIDs available for a single Switch is relatively small (see Section 3.2.2), so this is what will present the greatest vulnerability.

Therefore, selecting a Switch and its Subnetwork as objective (prioritizing the most critical ones) through its SID, we can spoof multiple registration requests until the LNID space is exhausted. Once this

Figure 4.2: UC3M PRIME Simulation Setup

took place, a legitimate node attempting to request connection to the Switch will face rejection due to the lack of available LNIDs for it until timeout.

If the attack were to take place, it would present a spike of the number of LNIDs interacting in the network due to the increase in allegedly "functioning" Nodes and the network's logical topology. The number of different LNIDs in place at the time of the attack depends on the strength of the overflow. For instance, a study's simulations uses a detection threshold of 10,000 new LNIDs in a span of 600s [15]. Clearly, the volume of SIDs needs to be significantly higher than what the victim system can process for this behavior to turn into a hazardous attack.

4.2.1 *Implementation*

A Service Node in a *disconnected* state shall transmit a Registration (REG) Control packet to the Base Node for it to be included in the Subnetwork. Thus, this is the procedure to be used by the forged requests in the attack's implementation.

4.2.1.1 *Packet Flow: Registration Process*

When we execute the registration procedure with our fake Service Node, we first send a REG Control packet, Registration Request (REG_REQ). However, at this point, no LNID or SID are still allocated to the Terminal so the following values would have to be included:

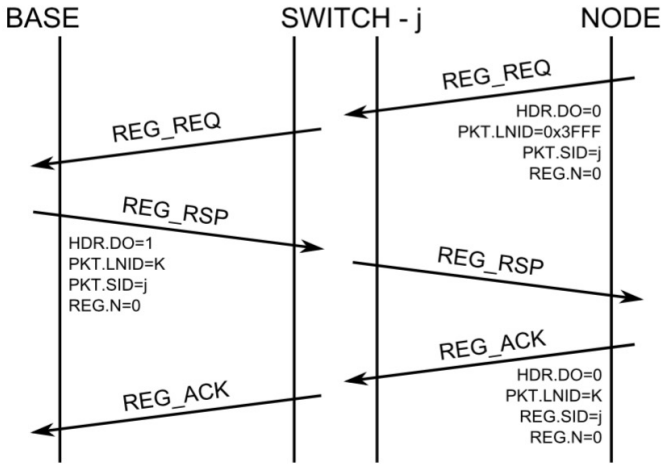
```
PKT.LNID = 0x3FFF (no value assigned yet)
PKT.SID = j (SID of the objective Switch (victim))
```

Once the request is complete, the Base Node shall allocate a unique LNID for our Terminal, always within the domain of our chosen victim Switch and its Subnetwork. This LNID value, together with the SID selected previously conforms the NID of the registering node. The allocation takes place in the Response (REG_RSP):

```
PKT.LNID = k (LNID value assigned)
```

Finally, the registration is concluded by the acknowledgement of the REG_RSP through the REG_ACK, as it is a three-way process. The complete packet flow can be seen in Figure 4.3.

Figure 4.3: Registration procedure: success



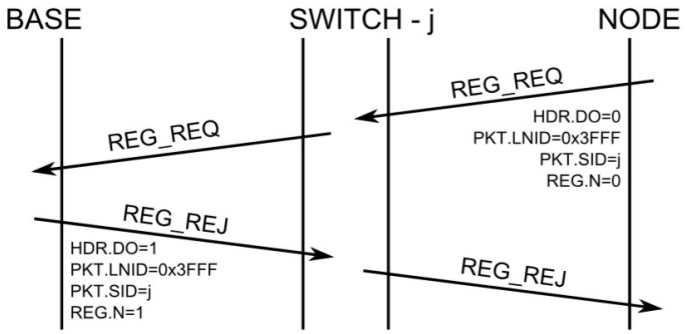
Once this process has been executed a large number of times, large enough to cover the complete LNID space for our desired victim Switch Node, no more IDs are available for more nodes to connect. At this point, when a legitimate node – either because it was disconnected, momentarily lacked connection, or lost connectivity by any other reason – would now face rejection from the Base Node responsible to manage said Switch, as we can see in Figure 4.4.

4.2.1.2 *Packet Details: Registration Request*

The first Registration Request packet is transmitted as a GPDU, identified with the `PKT.C = 1` bit to be considered as Control packet. Each type of Control message is then identified with the `PKT.CTYPE` field and the payload contains the data and information of the Control packet, different for every type.

In this case, we would need to employ Registration management (REG) Control packets. For it, we require `PKT.CTYPE = 1`.

Figure 4.4: Registration procedure: rejection



This Control packet is used to negotiate the Registration and Unregistration process of the Nodes, and its meaning varies according to the direction of the packet and its source.

4.2.1.3 *MAC Header*

Generic MAC Header fields are detailed in Figure 2.4, while its values for the Registration Request can be found in Table 4.1 below.

FIELD NAME	VALUE	DESCRIPTION / FUNCTION
Unused	0 (2b)	Unused bits are always 0.
HDR.HT	0 (2b)	Header type. HT=0 for GPDU.
Reserved	0 (5b)	Always 0 for this version, reserved for future use.
HDR.DO	0 (1b)	Down/Uplink. 0: PDU is Uplink.
HDR.LEVEL	? (2b)	PDU's Switching Hierarchy Level. If DO=0: represents level of the transmitter.
HDR.HCS	? (8b)	Header Check Sequence. Security calculation involving CRC and SNA values.

Table 4.1: REG Generic MAC Header fields

The field `HDR.LEVEL` is, in principle, unknown. One option would be to try increasing value until we obtain a response from the Base Node responsible for the Subnetwork. Generally, infrastructures do not implement more than two or three levels of hierarchy, so the options are bearable.

Failing to succeed by trying increasing values would leave us with the option to sniff packets from other previously connected nodes. Registration packets would be ideal, but do not happen very often in a normal network. However, this header is repeated for all Control and Data packets (GPDU) thus giving the attacker many valid options.

The computation of the last value, `HDR.HCS`, was seen in Section 2.4.3.1 previously. It is obtained by calculating a SCRC over the SNA and the 2 first bytes of the Header.

4.2.1.4 *MAC Packet contents*

We saw how the Packet consisted in header and payload, and header fields were detailed in Figure 2.6. For our purpose of registration, the needed field values are those shown in Table 4.2.

FIELD NAME	VALUE	DESCRIPTION / FUNCTION
Reserved	0 (3b)	Always 0 for this version. Reserved for future use.
PKT.NAD	1 (1b)	No Aggregation at Destination. 1: packet not aggregated with other packets at destination.
PKT.PRIO	3 (2b)	Indicates packet priority, from 0 to 3.
PKT.C	1 (1b)	Control. C=1: it is a Control packet.
PKT.LCID / PKT.CTYPE	1 (9b)	If C=1: CTYPE (Control packet type) is used. CTYPE = 1: Registration.
PKT.SID	0 (8b)	SID upon which we request registration. If HDR.DO=0, SID represents that of the packet source.
PKT.LNID	0x3FFF (14b)	LNID to be assigned to our requesting Terminal. If HDR.DO=0, LNID represents that of the packet source.
PKT.SPAD	- (1b)	1: padding is inserted while encrypting payload. *Only used with Security Profile 1*.
PKT.LEN	*len* (9b)	Length of packet payload (bytes).

Table 4.2: REG Packet header fields

It is worth noting that the `Priority` field is given the maximum value, as in the attacker's position we shall present no regard for other legitimate packets's priority and force the malicious traffic first.

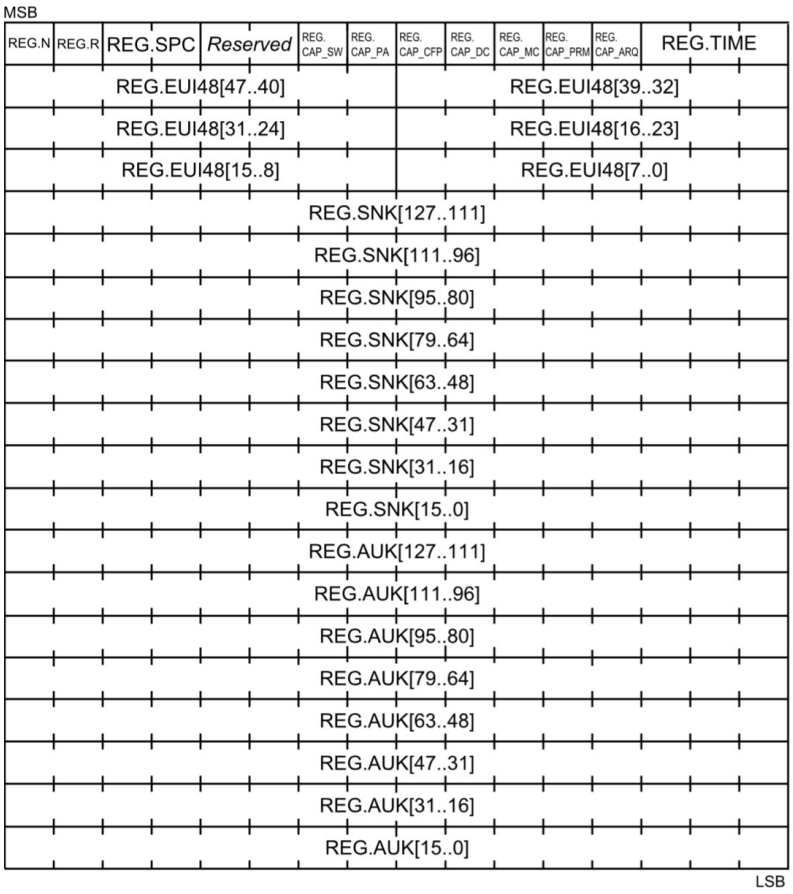
As for this one and the remaining times we craft a packet with a Service Node as a source, we will assign `PKT.SID` a value of `0` as the Specification details. However, and as an additional measure, it is recommended to sniff previous traffic to assure the target implementation follows these guidelines for Terminal's SIDs.

Meanwhile, `SPAD` field is left blank as it is only employed during the use of Security Profile 1, which is in our best interest not to use. It will always be preferable to use Security Profile 0 (`REG.SCP = 0`) to avoid unnecessary encryption problems in the attack.

Implementation details would come into play here to decide if the requesting Node can decide over the Security Profiling. It could be imposed by Base Node restrictions, hence posing the extra problem of key decryption, but as this exchange is initiated by this request's `REG.SCP`, it will be assumed that is not the case here.

The full structure of the packet can be seen in Figure 4.5.

Figure 4.5: Registration Control packet structure



Next, in Table 4.3, we can the more detailed description of the packet contents (payload) of a Registration Request, the required values for all its fields in order to serve the needed purpose for the attack.

As commented before, an attacker would strive for Security Profile 0 use. Hence this packet contains `REG.SPC = 0` to trick the responder Concentrator we only can manage encryption-less communications.

The field `REG.CAP_PA` indicates packet aggregation capabilities, and in the uplink request direction, our Terminal indicates its own capabilities. However, the downlink response is the Base Node's means of evaluating if all the devices int he cascaded chain (from itself to this

FIELD NAME	VALUE	DESCRIPTION / FUNCTION
REG.N	0 (1b)	Negative: 0 for Positive Register.
REG.R	0 (1b)	Roaming: Node not yet registered, requests a clear registration process (previous connection info from this Node shall be lost).
REG.SPC	0 (2b)	Security Profile Capability for Data PDUs.
Reserved	0 (2b)	Reserved for future versions of the protocol.
REG.CAP_SW	1 (1b)	Device is able to act as a Switch.
REG.CAP_PA	1 (1b)	Device has packet aggregation capability.
REG.CAP_CFP	1 (1b)	Device is able to perform the negotiation of the CFP.
REG.CAP_DC	1 (1b)	Device is able to perform direct connections.
REG.CAP_MC	1 (1b)	Device is able to use multicast for its own communications.
REG.CAP_PRM	1 (1b)	Device is able to perform PHY Robustness Management.
REG.CAP_ARQ	1 (1b)	The device is able to establish ARQ connections.
REG.TIME	- (3b)	Max. waiting time for ALV_B msgs until we assume disconnection. *Reserved for REG_RSP messages.*
REG.EUI-48	_:_:_:_ (48b)	EUI-48 of the Node requesting registration.
REG.SNK	- (128b)	Encrypted Subnetwork key. *Not present, not used for REG_REQ.*
REG.AUK	- (128b)	Encrypted authentication key. *Not present, not used for REG_REQ.*

Table 4.3: REG Control packet fields

Terminal) have aggregation capabilities. If they do, `REG.CAP_PA = 1`, otherwise `REG.CAP_PA = 0`.

Note that the `REG.CAP_PA, CAP_CFP, CAP_DC, CAP_MC, CAP_PRM` and `CAP_ARQ` fields all take value `1`, for we pretend to grant the biggest capabilities possible to the Base Node in order to reduce the rejection probability to a minimum.

"The REG Control packet, in all its usage variants, is transmitted unencrypted" [8]. However, some specific fields (`REG.SNK` and `REG.AUK`) do carry encryption using secure material depending on the context.

This all conforms an authentication mechanism for the registration mechanisms through REG_ACK, from `REG.AUK`'s encryption in REG_RSP to its decryption at the receiving end and possible encrypted retransmissions.

Fields REG.SNK and REG.AUK have an important role only in REG_RSP and REG_ACK with Security Profile 1 (REG.SCP = 1). For the rest of the exchange variants in the REG message context, these fields shall not be present in order to reduce the payload's length.

As an additional note, in case encryption was forced, in REG_RSP, the REG.SNK and REG.AUK are always encrypted with WKo (see end of Section 2.3). This means that if this field were to be cracked once, the encryption material it provides could be reused to infinity by an attacker. This material is shared among several nodes, connections and requests so cracking one could mean serious key management vulnerabilities.

The key itself is derived from random material in the SEC Control packet [8], broadcasted to everyone in the network without encryption. From an attacker's standpoint, having this information and any other pairs obtained by sniffing would mean relative ease for these keys to be cracked and used repeatedly in this scenario.

4.2.1.5 *Packet Details: Registration Acknowledgement*

After the Registration Request (REG_REQ) has been successfully crafted and transmitted over to the Base Node, it would respond with a REG_RSP packet. At that point it would be time for the attacker to reply and end the Registration process with a REG_ACK crafted packet. This packet means the acknowledgement of the registration process by the Service Node.

Most fields would stay the same in the Packet Header as those in the Request (Table 4.2), only differing in the following:

```
                PKT.LNID = K (<0x3FFF);
use the value in REG_RSP packet, sent by the Base Node

                PKT.SID = j (<0x3FFF);
use the value in REG_RSP packet, sent by the Base Node
```

LNID and SID values are generated by the Base Node at the time of the Response (REG_RSP packet). Before, in the Request, these values were both 0 as there was no Switch to refer to.

From the Response, the Terminal must take these values and incorporate them onto its Acknowledge packet, both different from 0.

The packet payload content is shown in Table 4.4, except for all the Capabilities' values (REG.CAP_SW, PA, CFP...) whose values stay the same as those seen in the Registration Request before, Table 4.3.

FIELD NAME	VALUE	DESCRIPTION / FUNCTION
REG.N	0 (1b)	Negative: 0 for Positive Register.
REG.R	0 (1b)	Roaming: previous connection info from this Node shall be lost.
REG.SPC	0 (1b)	Security Profile Capability for Data PDUs.
Reserved	0 (1b)	Reserved for future versions of the protocol.
REG.EUI-48	_:_:_:_ (48b)	EUI-48 of the Node requesting registration.
REG.SNK	- (128b)	Encrypted Subnetwork key. *Only present in Security Profile 1*.
REG.AUK	- (128b)	Encrypted Authentication key. *Only present in Security Profile 1*.

Table 4.4: REG ACK Control packet fields

4.2.2 *Extra: Forced Node Disconnection*

Finally, we might encounter the event that even though the remaining LNID space has been completed, but the victim nodes are still connected to the network thanks to them connecting before the attack launched. The objective of the attack is to leave all nodes out of the network, service-less and then face lack of LNIDs at reconnection attempt.Therefore an attacker in this situation would have to force disconnection of those node already in the network.

This behavior will be seen in greater detail in Attack #3, Section 4.4, but forced Node Disconnection can be executed through the forge of Unregistration Request packets. This makes the target Terminal think the Base Node is ordering its disconnection from the subnetwork and ends up requesting it itself.

According to the specification [8], "when assigning an LNID, the Base Node shall not reuse an LNID released by an unregister process until after (*macMaxCtlReTx +1*) * *macCtlReTxTimer* seconds, to ensure that all retransmit packets have left the Subnetwork".

This would mean that once the node has been forced to disconnect, an attacker would need to wait that *reuse time* to keep occupying those LNID values before the nodes attempt reconnection.

4.3 ATTACK 2: NODE REGISTRATION SPOOFING

Here we will explain in greater detail the steps needed to execute a *Node impersonation* attack. When it takes place, the rogue node will hijack the communications and functions of a legitimate working Service Node in the target network.

As commented before during the Denial of Service analysis Section 3.3.1, this attack reaches its biggest damage potential when used against a system that assigns a *new NID* to an already-registered EUI-48 requesting registration. This would mean a spoofer node registering with a victim's address in the same Base Node could effectively leave the victim out of the network by channeling all its communication through a different NID.

The execution of a spoofing of this type depends highly on the timeout presented by the Base Node. How long it takes the Base Node to unregister a Terminal within its reach conditions the effectiveness of the attack. Optimally, we would want the higher value possible so no unwanted unregistrations take place during the time of the attack. Through a long timeout value, even a node that gets disconnected to the network and reconnected again would then mimic the behavior of an attacker's spoofer node.

4.3.1 *Implementation*

In order to obtain a victim's EUI-48 address, the preceding step to attempting a registration onto its Base Node, one can make use of a technique mentioned previously: Sniffing. This address could be obtained also by direct access to the Node, where the number is shown in a screen. However here it will be assumed no physical access to the node and hence being sniffing the best option available.

Through this technique we can gain access to that desired Node Identification and perform the spoofed Node's registration. This is possible because the information travels unencrypted through the network, as it could be expected of a registration process of this nature.

Afterwards, registration of the spoofed node needs to take place, following the usual procedure. Packet contents and structure's details of the Registration Control packet (REG) can be found in Table 4.1, Table 4.2 and Table 4.3, together with the diagrams back in Section 2.4.1.

4.3.1.1 *Registration Process: Traffic Traces*

The exchange's trace from the perspective of the Base Node would look like the following for a Registration process:

```
[RX]
GPDU: sna: _:_:_:_:_:zz  UP  level:0  frametime:__  SCP
REG_REQ: eui48: _:_:_:_:_:xx  sid:0  spc:0  caps:0x45
(...)
[TX]
GPDU: sna: _:_:_:_:_:zz  DO  level:0
REG_RSP:
eui48: _:_:_:_:_:xx  sid:0  lnid:A  spc:0  caps:0x47  time:_
```

```
(...)
```

4.3.1.2 *Registration Spoofing: Steps and Traces*

However, we first need to check the system's Base Node manufacturer implemented behavior (2) from those mentioned at the beginning of this Attack section and further described in Section 3.2.2 of the protocol study. This behavior means the repeated registration of the *same EUI-48*, the same device, is assigned a *different LNID* every time. Once checked, we would need to exploit this situation by registering our rogue Service Node onto the network as a previously registered node (the victim), impersonating its functions and communications.

The checking and exploitation process would be the following:

i Registration (I) occurs when the request is sent to the Base Node (Z) with the node's LNID, to which I will refer here as `LNID (A)`.

ii Registration request (II) is later sent and this same Base Node accepts it, assigning a new different LNID, `LNID (B)`.
This proves the Base Node is using the desired behavior for the attack.

iii As an attacker, we could spoof this second request, Registration (II), forcing the Base Node to change the assignation from `LNID (A)` to `LNID (B)`, which the attacker controls and effectively hijacks the legitimate node's functions.

```
[RX]
GPDU: sna: _:_:_:_:_:zz  UP  level:0  frametime:__  SCP
REG_REQ: eui48: _:_:_:_:_:xx  sid:0  spc:0  caps:0x45
(...)
[TX]
GPDU: sna: _:_:_:_:_:zz  DO  level:0
REG_RSP:
eui48: _:_:_:_:_:xx  sid:0  lnid:A  spc:0  caps:0x47  time:_
(...)

...

[RX]
GPDU: sna: _:_:_:_:_:zz  UP  level:0  frametime:__  SCP
REG_REQ: eui48: _:_:_:_:_:xx  sid:0  spc:0  caps:0x45
(...)
[TX]
GPDU: sna: _:_:_:_:_:zz  DO  level:0
REG_RSP:
eui48: _:_:_:_:_:xx  sid:0  lnid:B  spc:0  caps:0x47  time:_
(...)
```

Proof of the attack being executed would be the multiple registration attempts of a Terminal Node, ending up with different LNID value afterwards.

Now the victim's communications are hijacked and the packet flow to it will stop and be redirected towards our Spoofer Node. Changes in the LNID, which can be seen marked in blue, highlight how the spoofer has taken over the real node. Meanwhile, in the eyes of the Base Node, the legitimate Service Node has just re-registered and has just been given a new LNID.

Depending on the distance from the attacker to the victim node, the behavior observed by the victim node can differ. Being at a great distance and connected through different switches will leave the victim no chance of detection and simply stop receiving packets meant for it. However, if they are close to each other, the victim could experiment a flow of anomalous packets and receive an unwanted Registration Request response. This leaves the door open on the side of the victim Terminal to update its LNID and adapt to the changes presented by the Base Node, thus stopping the attack as designed. This behavior is implementation-dependent and not directly present in the standard.

4.4 ATTACK 3: BASE NODE SPOOFING

Only one Base Node per network is allowed. If an attacker managed to inject a second one, it would severely disturb the network. Moreover if all nodes were to disconnect from the legitimate concentrator and connect to the attacker's rogue Base Node, they would be left at their mercy while thinking everything is working perfectly for them. This attack presents this exact possibility: Terminals being forced to disconnect from the real Base Node and being given the chance to connect to a forged Base Node that would, in reality, just keep them connected but service-less.

This risk (Section 3.3.1) presents great potential damage to a working network, leaving its Terminals with a connection to a spoofed Base Node they believe will forward them legitimate data, but effectively disconnected from the real network.

However, this attack presented the necessity for the Nodes to disconnect from the real functioning Base Node in order to get them reconnected to the spoofed one, a serious drawback for the attack to take place on its own.

This leaves the effectiveness depending on how an attacker could make this reconnection happen, potentially through forcing another unwanted behavior in the nodes.

In this case, the *Demotion/Unregistration/Disconnection Spoofing* (Section 3.3.1) appears as a viable option. This attack consisted on injecting traffic onto the node's communications to make Service Nodes

think their Base Node is ordering them to disconnect from the network.

Assuming this were to happen, when a Node is finally disconnected due to the traffic injection, it will become subject to becoming a victim of the aforementioned Base Node Spoofing attack.

4.4.1 *Implementation*

The first step would be to place the malicious Base Node we plan on using for the spoofing. This, of course, would depend on the actual network chosen as victim, whose topology could be already known or learnt from using sniffing techniques commented previously. Modulation techniques would also play a significant role, but it is outside the scope of this thesis.

Following that, carrying out the *Demotion/Unregistration/Disconnection Spoofing* would be next, by inserting, via our proposed infrastructure (Figure 4.1), the corresponding traffic for Service Node disconnection: CON_CLS_B.

This injection will make it think the instruction came from the Base Node and will answer accordingly. An answer the Base Node itself would interpret as a disconnection request from the Terminal (as explained in Section 3.3.1). The attack is to be repeated for the target Service Nodes in the network until all targeted nodes are affected and thus, disconnected from the original Base Node.

As our Spoofer Node is already in place, disconnected nodes would then request connection to our Base Node. This request would be made thanks to the Base Node transmitting the adequate Beacons to alert those Terminals of its existence.

As they receive the Beacon and have no other Base Node to connect to, the recently disconnected Terminals request registration onto our malicious Concentrator. Once the registration is completed and both sides have agreed on communication parameters, that Service Node would effectively be out of service and away from the functioning network's reach.

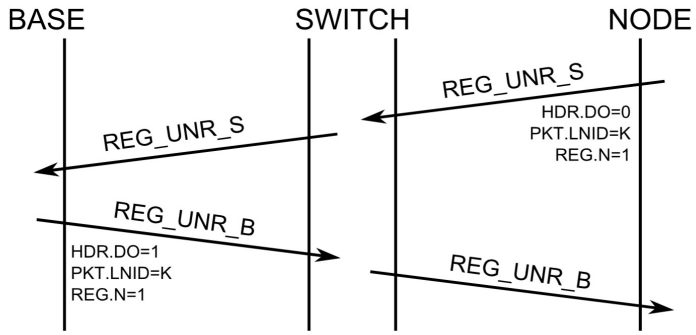
With the Terminals connected to the malicious Base Node, it now would need to keep them connected and feeling their communication is stable through Keep Alive messages even though no effective data will reach them.

4.4.1.1 *Disconnection Process*

At any point, Base or Service Node might chose to end an existing connection, what is known as *Disconnection process*. If any of the elements receives a disconnection request, the Node shall acknowledge it and begin the process actions. Another option is to approach this from an unregistration perspective, but we will opt for the former here, disconnection.

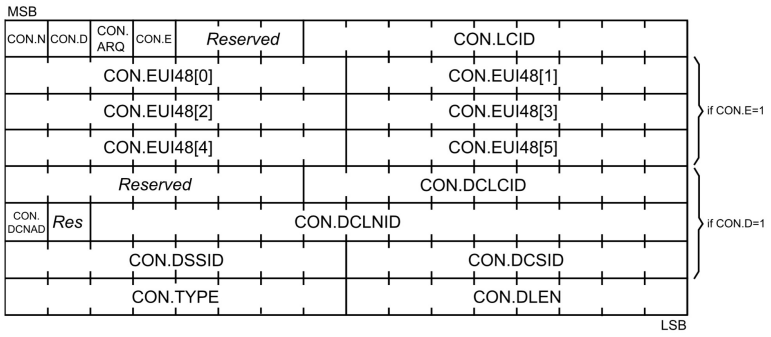
Once successfully unregistered, a Service Node shall move back to a *Disconnected* functional state, while the Base Node shall relocate the resources that were being employed for the Terminal. If this process were to be begun by a Service Node, as it will be shown in this description, the packet exchange would look as it does in Figure 4.6.

Figure 4.6: Unregistration process, initiated by a Service Node



The Disconnection process will occur through the use of the CON Control packet (`PKT.CTYPE = 2`). It is generally used for negotiating connections and, as well as some others seen before, its meaning varies depending on the direction it goes and its values. The complete structure of the CON packet can be seen in Figure 4.7.

Figure 4.7: Connection Control Packet



This diagram shows the complete version of this CON_CLS packet, including all optional fields. However, some of them would not be present in the traffic this attack needs to generate. Only the `CON.N` field is relevant in closing an existing connection.

The packet needed here would be 12 octets shorter in total. When `CON.D` is 0, `CON.DCNAD`, `CON.DSSID`, `CON.DCLNID`, `CON.DCLID`, `CON.DCSID` and the reserved field between `DCNAD` and `DSSID` will not be present in the message (6 octects less). In addition, when `CON.E` is 0, the field

`CON.EUI-48` will not be present, making the message another 6 octets smaller.

As a side note, the alternative way to run this step commented previously, unregistration, would need of Registration Control packets (REG) instead of Connection Control ones. In a similar fashion, they would take on the form of REG_UNR packets to initiate the process.

For this scenario, we have decided to craft a Connection Close (CON_CLS) packet coming from a *fake* Base Node, denoted as CON_CLS_B. This means the target for this first spoofed packet is the Service Node we want to force disconnection upon.

Once a Service Node receives a CON_CLS, presumably coming from its Base Node above, it will reply back. This reply message, CON_CLS_S, would unleash a chain reaction between Base and Service Node that ends up on the Terminal's forced disconnection from the Subnetwork. This scenario was explained in detail in Section 3.3.1's Disconnection Spoofing.

The detailed value description for each field in the packet is found in Table 4.5 and Table 4.6. Note that Table 4.5 shows only the most significant values for the Control Packet header and those changing value, the rest maintain the same values used for the REG packet in previous attacks (Table 4.2). Generic MAC Header values (Table 4.1) will not be shown in a new table; they maintain the same values used before.

FIELD NAME	VALUE	DESCRIPTION / FUNCTION
HDR.DO	1 (1b)	1: MAC PDU is Downlink.
PKT.C	1 (1b)	Control. C=1 if it is a Control packet.
PKT.LCID / PKT.CTYPE	2 (9b)	If C=1, CTYPE represents the Control packet type. CTYPE = 2. Connection.
PKT.SID	0 (8b)	If HDR.DO=1, SID represents that of the packet destination.
PKT.LNID	? (14b)	If HDR.DO=1, LNID represents that of the packet destination.
PKT.LEN	*len* (9b)	Length of packet payload (bytes).

Table 4.5: CON Packet header fields

Regarding the header, and given this is one of the few packets needed to go downlink, some values present significant change. `SID` is now `0` for the packet's destination is a Terminal, not a Switch. Similarly `LNID` is now referring to the victim Node's Identifier, whose value is given by the Concentrator at the time of registration. For this reason, the most viable way of obtaining such value from the point of

view of an attacker is sniffing any of the packets between these two devices, where this value will be present.

FIELD NAME	VALUE	DESCRIPTION / FUNCTION
CON.N	1 (1b)	Negative connection.
CON.D	0 (1b)	Information about direct connection is not carried by this packet.
CON.ARQ	0 (1b)	ARQ mechanism is not enabled for this connection.
CON.E	0 (1b)	EUI presence. 0: Not having EUI-48. Connection management towards Base Node.
Reserved	0 (3b)	Reserved for future versions of the protocol.
CON.LCID	? (9b)	LCID of the connection being managed. 0-255: connection initiated by Base Node. 256-511: connection initiated by Service Node.
CON.EUI-48	- (48b)	*Not present.*
Reserved	0 (7b)	Reserved for future versions of the protocol.
CON.DCLCID	- (9b)	*Not present.*
CON.DCNAD	- (1b)	*Not present.*
Reserved	- (1b)	*Not present.*
CON.DCLNID	- (14b)	*Not present.*
CON.DSSID	- (8b)	*Not present.*
CON.DCSID	- (8b)	*Not present.*
CON.TYPE	? (8b)	Connection Type. Identify which Convergence layer to be used.
CON.DLEN	*len* (8b)	Length of CON.DATA field (bytes).
CON.DATA	*(variable)*	Connection specific parameters, depend on Convergence Layer.

Table 4.6: CON Control packet fields

Just like it happens with the `LNID` field in the header, the `CON.LCID` value in the packet's payload would need to be sniffed from some previous traffic between these two devices. It represents a local value for their connection, fixed during the registration process, and nothing intrinsic to the devices one could know beforehand.

One should note how the `CON.TYPE` field cannot be fixed here. Its value depends on the Convergence Layer being used in each specific implementation. In addition, the `DATA` field is also Convergence Layer specific and thus cannot be defined here as those parameters specify each individual connection. For more details on each of the options, refer to Annex E of the PRIME Specification for this version [8].

When the Service Node receives this packet *presumably* from the Base Node, the Terminal would interpret the Base Node considers their connection no longer active. The victim Service Node will acknowledge the situation with a CON_CLS_S packet, that will serve as a trigger in the Concentrator to effectively disconnect both parties involved.

4.4.1.2 *Beaconing Process*

Each Service Node keeps a table of Switch and Base Nodes internally, updated upon reception of Beacon PDUs. The disconnected Node then decides, based on implementation policies, a Concentrator upon to which request registration.

If no beacons were received in an specified time span (*macMinSwitchSearchTime*, see [8] for more information), the Terminal would broadcast a PNPDU itself. A PNPDU that the legitimate Base Node could receive and attempt to reconnect such node. As an attacker, one shall avoid that event by broadcasting a Beacon PDU before the timeout, attracting the Terminal and inviting it to register.

BPDUs are explained in greater detail in Section 2.4.2, but its main use is broadcasting information on Subnetwork's frame structure and is used by disconnected Service Nodes to acknowledge the existence of Switch/Base Nodes in its vicinity.

Switch and Base Nodes have the transmission role for these BPDUs. However, as in our forged Subnetworks there are no Switches, the Base Node itself would be in charge of its complete transmission to attract victim Terminals that just suffered the *Disconnection attack*.

The Beacon PDU our rogue Base Node would need to transmit would contain the values in Table 4.7.

There are some remarkable fields whose value needs some additional explanation. `QLTY` takes a value of 7 (maximum) as we would be operating as the network's own Base Node.

Similarly, `LEVEL` takes the maximum value – all 1s – as the Base Node is on the top of the Switch hierarchy.

We would not want to create space for other Switches to transmit their beacons in the MAC frame, so `BCN.CNT = 1` in order to define a single Beacon slot in the frame for the Base Node.

Frequency of transmission is maximized (`FRQ = 0`) as we would desire maximum transmission priority and frequency within each frame, reason why Base Nodes shall always transmit exactly one Beacon per frame. The Base Node at that point has no other goal but to ease the way for Terminals to connect.

Finally, the `SNA` and `SID` values can be crafted by the attacker, as no service is going to be served and no external parameters need to be followed. If the transmitting device was a Switch, `SNA` would be that of the Base Node on top and `SID` that of the Switch itself, but in this case the transmitter is the actual Base Node so the values are

equal. Depending on specific topologic conditions, an attacker might want to learn about surrounding Base Node and Switch identification values so those are not repeated when crafting values for the rogue Beacon, causing undesired problems.

4.4.1.3 *Registration Process*

Now, the recently disconnected Terminal Nodes (in *Disconnected* functional state) shall proceed to carry out the registration process, just as we saw in Attack #1 (Section 4.2). They need to transmit a REG Control packet to our Base Node to get included in the Subnetwork and start receiving traffic again, even though that is exactly the opposite of what they would see happen.

When the Service Node requested connection to our Spoofer BN, the negotiation would be that in Figure 4.3 and the MAC PDU's content is detailed in Table 4.1, Table 4.2 and Table 4.3.

After the Registration Requests we would receive as Base Node, one shall then reply with the subsequent accept message (we do not consider rejecting any). For this purpose, the first step it should take is to allocate a NID value for each node trying to be part of the Subnetwork it manages.

The Response packet REG_RSP would be similar to that of the Request, but the needed changes are detailed below in Table 4.8.

Again, the `LNID` value would be assigned by us, the Base Node, and will identify that node within the Subnetwork. It will have a value < 0x3FFF.

The packet's destination is a Terminal, not a Switch, which is why it is given a `SID` value of `0` as we have done throughout this work.

Now regarding the `REG.TIME` value, we have a special case for this field takes on a value of zero except for the Registration Response. In the RSP, we have chosen to maximize its value in order to increase the Terminals' tolerance to unregistration without a Keep-Alive packet. The higher this number, the less ALV_B packets the Terminal needs before assuming unregistration.

If the attack were to take place over a real Subnetwork, traffic traces during the Base Node Spoofing attack would include something like those shown below. These are taken from a simulated Smart Meter Panel in [15].

```
[RX]
GPDU: sna: _:_:_:_:_:aa  UP  level:0  frametime:__  SCP
REG_REQ: eui48: _:_:_:_:_:bb  sid:0  spc:0  caps:0x45
(...)

...
...
```

```
[RX]
GPDU: sna: _:_:_:_:_:cc  UP  level:0  frametime:__  SCP
REG_RSP: eui48: _:_:_:_:_:dd  sid:0  spc:0  caps:0x45
(...)
```

The traces show how, despite being in a single Subnetwork, where only one Base Node is supposed to be operating, something does not work as expected. First, a Terminal with EUI: . . . :*bb* requests Registration upon the . . . :*aa* Concentrator and later a different Service Node (. . . :*dd*) connects to a second Base Node with address . . . :cc, which would be the malicious device the attacker controls.

4.4.1.4 *Subnetwork Maintenance: Keep-Alive Process*

Once successfully connected to the attacker's Base Node, it needs to maintain its basic functions as such to keep the victim Terminals under its control. For this, the Keep-Alive process is a vital procedure the rogue Concentrator needs to take care of.

The Keep-Alive process is used to detect if a Service Node leaves a Subnetwork it was registered onto. This can be decided due to fatal errors it is unable to recover from, changes in its configuration, etc

From registration time (reception of REG_RSP), the Service Node starts a timer $T_{keep\text{-}alive}$, employing the `REG.TIME` field for it. From here, every time this node receives a ALV_B packet, it will restart such timer with the value from the `ALV.TIME` field in the packet.

If the timer were to ever expire, the Service Node would assume the Base Node has unregistered it from the network. The Service Node is also in charge of the transmission of the ALV_S packet back to the Base Node before $T_{keep\text{-}alive}$ with the same purpose. Otherwise, it would be removed from the Switch tables.

The packet header and payload field values for the ALV_B the Base Node needs to transmit are as seen on Table 4.9 and Table 4.10 respectively. As before, only the most significant packet header values are shown, the rest stays as shown in Table 4.2, as well as Generic MAC PDU Header's values (Table 4.1). This excludes `HDR.DO` as the packet is intended to go from Base to Service Node, downstream direction (ALV_B):

```
HDR.DO = 1 (Keep-Alive message from the Base Node, ALV_B)
```

In the header, `PKT.LNID` values would normally be obtained by sniffing the previous exchanges in the Subnetwork. However, now it is the Base Node controlled by the attacker the Node responsible for the LNID management (and assignment, at registration time), so it would be a known value at the time this packet needs to be sent.

The `ALV.TXCNT` and `ALV.RXCNT` fields are meant to keep track of the number of Keep-Alive packets a Node transmits and receives, respectively. This applies for both the Base Node and the Service Node. The field has a cyclic nature, as represented by the *Modulo 8* in its description. These counters are reset to 0 in the Registration process.

"The algorithm used by the Base Node to determine when to send ALV_B messages to registered Service Nodes and how to determine the value ALV.TIME and REG.TIME is not detailed" in the PRIME Specification for v1.3.6. [8]

It is worth noting how, while most of the previous Control packets before had `PKT.SID = 0` (due to it being transmitted by a Terminal), now it is the Base Node who sends it. This means the value for this field needs to be that which the Concentrator acquired during the Beaconing process (Table 4.7).

4.5 ATTACK 4: LSID OVERFLOW

As explained in Section 3.3.1, this attack consists on the over- promotion of nodes to Switches, thus blocking the promotion of new legitimate ones due to the LSID space being full as well as taking over existing working Switches. At that point, all real nodes connected to the malicious Switches will experience serious service problems.

Blocking the entrance of real Switches into a network implies the completion of the LSID space, where there are not new values available. This is possible because there are only 256 possibilities (8 bit value) for this field, a relatively weak point from a Denial of Service perspective.

As some of the previous attacks seen in this chapter, there is high implementation dependence for the accurate description of this overflow.

For the attack to occur, similarly to the *LNID Registration Overflow*, there needs to be a noticeable growth in the number of different SIDs for a determined time span. As the previously mentioned case, the volume of SIDs needs to be significantly higher than what the victim system can process for this behavior to turn into a hazardous attack. In the same study referenced in that Section 4.2 [15], shows how the attack could set the detection threshold when the number of LSIDs exceed 200 for an interval of 3,600 seconds.

4.5.1 *Implementation*

For both "variations" on the attack (exhaustion of LSIDs and Switch hijacking), the same first steps are needed so the two will be explained alongside each other.

4.5.1.1 *Registration Process*

This process maintains the same registration procedures detailed in previous attacks (Section 4.2). In here, a disconnected Node requests registration (REG Control packet) upon a Base Node, asking to become part of its Subnetwork. After a positive response, the negotiation is acknowledged and now that node has a LNID assigned.

The complete successful packet flow communication is shown in Figure 4.3. The details of the packets needed to be sent from the side of our rogue Node are again detailed in Table 4.1, Table 4.2 and Table 4.3.

4.5.1.2 *Switch Promotion Process*

The next step towards our goal would be to transform those Service Nodes recently registered onto the network into Switches.

According to the PRIME Specification [8], for this use two mechanisms arise as adequate candidates for the task: Promotion Needed PDUs and PRO Control packets. Promotion Needed PDUs are a type of MAC packet working as a notification from a disconnected Node to any of its neighboring Terminals requesting the promotion of any of them who could turn into Switch. It is used when the disconnected Node has not received any Beacons for connection and lacks connectivity with any existing Switches.

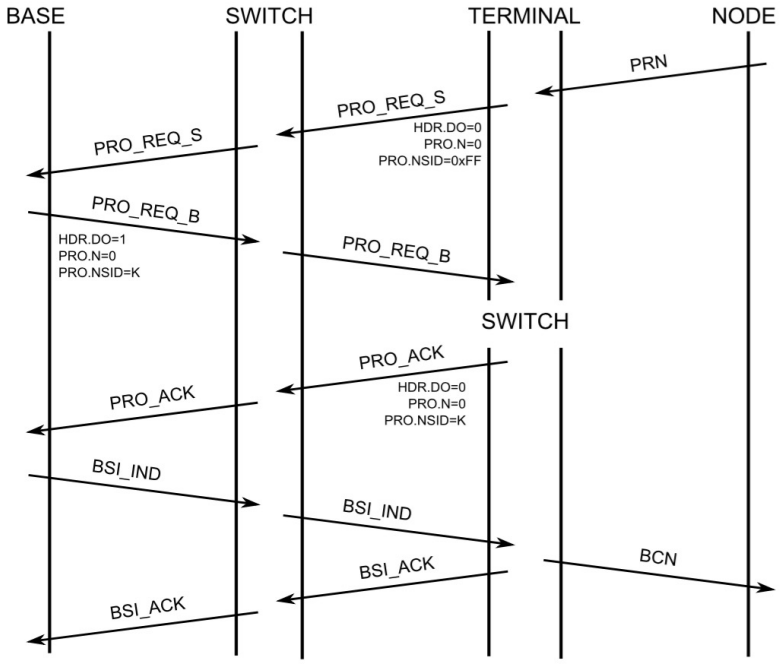
However, the PRO Control packet (`PKT.CTYPE = 3`) is the chosen one for this functionality: "promoting a Service Node from Terminal function to Switch function" [8]. Remember that Control packets are encapsulated inside of Generic MAC PDUs. As seen before, this packet can take on different meanings depending on its direction and values inside.

Below, in Figure 4.8, we can see what the exchange would look like for a case like ours: successful Promotion Request initiated by the Service Node.

For the Generic MAC Header values, the packet shall employ the same values the Registration Request used before, see Table 4.1. Just like it has been done in previous occasions for the PRO Control packet header values (Table 4.11), here we will only show those most significant and different from those used in Table 4.2. As a reminder due to it being referenced in the tables below, the value for `HDR.DO` is still `0`, as the packet goes in uplink direction.

The complete structure of the PRO Control packet can be seen in Figure 4.9. In this case, we need to send a PRO_REQ_S packet, as it encapsulates the Promotion Request initiated by the Service Node. This PRO_REQ_S includes all fields in a PRO packet, whereas other messages only contain `PRO.N, PRO.RC, PRO.TIME` and `PRO.NSID`.

Figure 4.8: Connection Control Packet



Later, in Table 4.12, we can find a detailed representation of all the field values needed for this purpose in the PRO_REQ_S packet's payload.

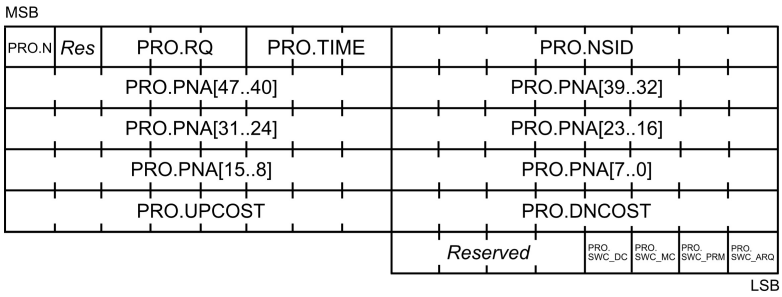
Note that, according to the PRIME Specification, the PNA could be another node's EUI-48, not necessarily the sender's. In this case, each node is meant to manage its own promotion so it should be its own. In order to get the Promotion Requests to be accepted by the Base Node, it is needed to use different PNAs in the various packets to be sent so the Base Node is tricked into accepting such requests to turn the Nodes into new Switches. The value for packet header's `SID = 0` as the transmitting device is still functioning as Service Node.

Also in the header, the `LNID` value is that which was given to the Terminal during the Registration process. Therefore, this value would not be unknown for the operator of the Service Node itself as it would have appeared in previous exchanges with the Base Node.

It is worth remarking fields `UPCOST` and `DNCOST`, whose value would be calculated "in the same way a Switch Node calculates the value it places into its own Beacon PDU" [8]. This field cannot be established here in a general way, and would assume different value depending on the Network specifications and the modulation scheme used in every hop of the topology (in each direction).

Furthermore, the `PRO.SWC_DC, SWC_MC, SWC_PRM` and `SWC_ARQ` fields all take value `1`, for we pretend to grant the biggest capabilities possi-

Figure 4.9: Connection Control Packet



ble to the Base Node in order to reduce the rejection probability to a minimum.

Once the Request has been executed, the Base Node in charge of the Subnetwork would reply (assuming fits its requirements and the request is accepted) with a PRO_REQ_B message. This Promotion process would need to be closed with the acknowledgement from the hands of the device who started the process, our Terminal. This packet means the acknowledgement of the successful Promotion process by the Service Node.

The packet header stays the same as that of the Promotion Request (Table 4.11), so it will not be repeated again.

medskip The packet payload content is shown in Table 4.13, but only the most important fields and those presenting changes compared to the Request.

`PKT.NSID = K (<0x3FFF)`, where `K` would be the value in PRO_REQ _B (Response packet), sent by the Base Node. `NSID` value is generated by the Base Node at the time of the Response (PRO_REQ_B packet) and this one shall be copied from that packet's. Before, in the Request, this value was `0` as there was no Switch to refer to as source, it was still a Terminal.

Repeating this Promotion Request enough number of times would result in the exhaustion of the LSID space that would lead to new Switch Promotion Requests being rejected by the Base Node.

4.5.1.3 *Switch Hijacking Process*

Depending on the implementation used in the Base Node we could find that once the LSID limit is reached, the Base Node could decide to demote the oldest switches to make room for the new (rogue) ones.

Replacing old switches, with nodes associated to them, with new bogus devices would mean serious connectivity problems for the Terminals working at the orders of those demoted. However, it would be hard to determine the switches being replaced first as there is a

high implementation dependence in this aspect. As a result, rather than a targeted attack to a specific Switch, it would be more effective to opt for a brute-force alternative: continue promoting devices until no real Switch is

Once the legitimate Switches have been all removed, though the beaconing process, the working nodes would be attracted to the rogue switches now in the network via the Beaconing mechanism (review Section 4.4 for more information on packet forging details for it).

These nodes would request registration, to which it would be necessary to respond (refer to Table 4.2 and Table 4.3 for details on this, taking into account the needed variations for an adequate REG_RSP packet, as seen in Table 4.8).

Following the registration process, maintenance of the Subnetwork needs to happen in order to trick the legitimate Terminals into the illusion they are connected to a real Switch and just waiting for data they will never receive. For this, it is necessary to maintain the Keep-Alive process under control, as detailed in Attack 2, Section 4.4 (packet details: Table 4.9 and Table 4.10).

4.5.1.4 *Extra: Base Node Beacon Management*

At some point, the Base Node could decide to change parameters in the Beacon or exchange information about its transmission. For this, the Beacon Slot Information (BSI) Control packet (`PKT.CTYPE = 4`) would be employed.

These packets are generated by Base and Switch Nodes but need to be acknowledged even in case of rejection by the receiving devices, including ours. As a rogue Switch, it is not likely that an attacker would send any of these, but could be its receptor.

For detailed instructions on how the Acknowledgement packet needs to be crafted, refer to Section 4.4.5.6 of [8].

4.6 DISCARDED IMPLEMENTATIONS.

Some attacks attempting a successful exploit of some of the vulnerabilities commented previously in this work have been already tested. However, the approaches below did not meet the expected results regarding performance, harm, or noticeable effects on a normally functioning network. Here, they will be briefly explained here even though they will be discarded due to their results not meeting the effectiveness desired.

4.6.1 *Promotion Needed Flooding*

According to this experiment carried out in [15], the performance of the attack in a simulated environment, showed how in a network the

number of Switches would significantly increase as one would expect, regarding the attack's behavior commented previously. There is an increase in the congestion in the network due to the more frequent PNPDU messages flowing and promoting nodes to Switches. This, of course, overcomplicates the network topology.

Even though the smart meters able to report their activity is reduced noticeably, the final statistical results for "the time needed to obtain the readings" are not significantly larger when undergoing the attack. This would mean that, despite the attack, the network would be able to operate at a reduced rate reporting reads with no high degradation in the time needed.

4.7 ATTACK EXTENSIONS

Some of the mechanisms used in the previous sections of this chapter can allow an attacker to perform extra inferences with the target systems. Even though they do not conform an active asset for a Denial of Service attack beyond its use in some of the attacks before, they can then generate considerable collateral effects and serve as valuable tools.

4.7.1 *Sniffing*

Sniffing was used in the *Node Registration Spoofing* attack and needed/recommended in most of the rest for diverse purposes. It can be used as well to eavesdrop crucial information flowing through the network, proven useful in attacking offensive scenarios.

If a sniffer was put in place during a consumption report request or transmission, apart from its use in the attack, it can allow the attacker to obtain such report interpreting the data contained in the packet. From the hexadecimal trace, data and consumption data can be obtained.

Here it should be noted that this collateral leak of information is dependent on the use of Security Profile 0 configuration by the system, whereas it is not necessary for its "main" use on this report as the desired data then always travels in clear text.

FIELD NAME	VALUE	DESCRIPTION / FUNCTION
Unused	- (2b)	Alignment with MAC_H field in PPDU header.
HDR.HT	2 (2b)	Header Type. 2: Beacon PDU.
Reserved	0 (1b)	Reserved for future versions of the protocol.
BCN.QLTY	7 (3b)	Quality of round-trip connectivity from this Switch to Base. Base Node: 7 (max).
BCN.SID	? (8b)	Switch identifier of transmitting Switch.
BCN.CNT	1 (3b)	#Beacon slots in the frame.
BCN.SLT	0 (3b)	Slot in which this BPDU is transmitted. 0: reserved for Base Node.
BCN.CFP	0 (10b)	Offset of CFP from start of frame BCN.CFP=0 indicates absence of CFP in a frame.
Reserved	0 (1b)	Reserved for future versions of the protocol.
BCN.LEVEL	0x3F (6b)	Hierarchy of transmitting Switch in Subnetwork. 0x3F: max.
BCN.SEQ	0 (5b)	Sequence# of this BPDU in super frame. Incremented for every Base Node's beacon, propagated down by Switches.
BCN.FRQ	0 (3b)	Transmission frequency of this BPDU. 0: 1 Beacon per frame.
BCN.SNA	? (48b)	SNA in which the Switch transmitting this BPDU is located.
BCN.UPCOST	0 (8b)	Total Uplink cost from the transmitting Switch to the Base Node. Base Node: UPCOST = 0.
BCN.DNCOST	0 (8b)	Total Downlink cost from the Base Node to the transmitting Switch. Base Node: DNCOST = 0.
CRC	? (32b)	Calculated as explained in Section 2.4.3.1, over the complete BPDU except for the CRC field itself.

Table 4.7: Beacon PDU fields

FIELD NAME	VALUE	DESCRIPTION / FUNCTION
HDR.DO	1 (1b)	1: MAC PDU is Downlink.
PKT.SID	0 (8b)	If HDR.DO=1, SID represents that of the packet destination.
PKT.LNID	? (b)	If HDR.DO=1, LNID represents that of the packet destination.
REG.SPC	0 (2b)	Security Profile Capability for Data PDUs.
REG.TIME	7 (3b)	Time to wait for ALV messages before assuming disconnection from the Base Node. TIME = 7 -> 4096 seconds ~68.3 mins.

Table 4.8: REG RSP Control packet fields

FIELD NAME	VALUE	DESCRIPTION / FUNCTION
PKT.C	1 (1b)	Control. C=1: it is a Control packet.
PKT.LCID / PKT.CTYPE	2 (9b)	If C=1, CTYPE represents the Control packet type. CTYPE = 2 for Connection.
PKT.SID	? (8b)	If HDR.DO=0, SID represents that of the packet source.
PKT.LNID	? (14b)	LNID assigned to our Terminal. If HDR.DO=0, LNID represents that of the packet source.
PKT.LEN	*len* (9b)	Length of packet payload (bytes).

Table 4.9: ALV Packet header fields

FIELD NAME	VALUE	DESCRIPTION / FUNCTION
ALV.RXCNT	0 (3b)	Counter for # of received ALVs (mod 8).
ALV.TXCNT	1 (3b)	Counter for # of transmitted ALVs (mod 8).
Reserved	0 (7b)	Always 0 for this version of the protocol.
ALV.TIME	7 (3b)	Time to wait for ALV messages before assuming disconnection from the Base Node. TIME = 7 -> 4096 seconds ~68.3 mins.
ALV.SSID	0xFF (8b)	0xFF for Terminals. SID for Switches.

Table 4.10: ALV Control Packet fields

FIELD NAME	VALUE	DESCRIPTION / FUNCTION
PKT.C	1 (1b)	Control. C=1: it is a Control packet.
PKT.LCID / PKT.CTYPE	3 (9b)	If C=1, CTYPE represents the Control packet type. CTYPE = 3 for Promotion.
PKT.SID	0 (8b)	If HDR.DO=0, SID represents that of the packet source.
PKT.LNID	? (14b)	LNID assigned to our Terminal. If HDR.DO=0, LNID represents that of the packet source.
PKT.LEN	*len* (9b)	Length of packet payload (bytes).

Table 4.11: PRO Packet header fields

FIELD NAME	VALUE	DESCRIPTION / FUNCTION
PRO.N	0 (1b)	N=0 for Positive promotion (to Switch).
Reserved	0 (1b)	Reserved for future versions of the protocol. Always 0 for this one.
PRO.RQ	7 (3b)	Receive quality of PNPDU from Terminal requesting promotion. 7: max.
PRO.TIME	? (3b)	The ALV.TIME value being used by this Terminal.
PRO.NSID	0xFF (8b)	New Switch Identifier of the Terminal whose promotion is being managed, to be assigned by the Base Node.
PRO.PNA	_:_:_:_ (48b)	Promotion Need Address. EUI-48 of the Terminal requesting promotion. Only included in the PRO_REQ_S message.
PRO.UPCOST	? (8b)	Total Uplink cost from the Terminal Node to the Base Node. Only included in the PRO_REQ_S message.
PRO.DNCOST	? (8b)	Total Downlink cost from the Base Node to the Terminal Node. Only included in the PRO_REQ_S message.
Reserved	0 (4b)	Reserved for future versions of the protocol. Always 0 for this one.
PRO.SWC_DC	1 (1b)	Direct Connection Switching Capability. 1: device able to behave as Direct Switch.
PRO.SWC_MC	1 (1b)	Multicast Switching Capability. 1: device able to behave as Direct Switch.
PRO.SWC_PRM	1 (1b)	PHY Robustness Management Switching Capability. 1: device able to perform PRM for Terminals when behaving as a Switch.
PRO.SWC_ARQ	1 (1b)	ARQ Buffering Switching Capability. 1: device able to perform buffering for ARQ connections while switching.

Table 4.12: PRO Control Packet fields

FIELD NAME	VALUE	DESCRIPTION / FUNCTION
PRO.N	0 (1b)	N=0 for Positive promotion (to Switch).
PRO.NSID	? (8b)	New Switch Identifier of the Terminal whose promotion is being managed, to be assigned by the Base Node.
PRO.PNA	_:_:_:_ (48b)	Promotion Need Address. EUI-48 of the Terminal requesting promotion. Only included in the PRO_REQ_S message.

Table 4.13: PRO Control Packet fields

5 CONCLUSION

Smart Grid implementations are growing in number every day and that fact strengthens the need for secure design and development of such technology. I trust this thesis can help with the improvement of various security aspects of the PRIME Specification seen here in the form of attack designs.

This work has been built on the decision of focusing mainly on Denial of Service attacks for this infrastructure as it has presented itself as one of its main weakness. Previous instances of Internet of Things applications have suffered from this same hazard and poses itself as a remarkable threat for Smart Grid real-life secure implementations. These attacks are powerful in that they do not always need a unique critical crack in the protocol's design but can exploit several small mistakes or simply open unguarded "doors" to take down a service as crucial as this one. This makes for a greater chance of finding possibilities for attack designs fulfilling our requirements, using either one or multiple vulnerabilities found in the standard's analysis.

One should also note the Critical Infrastructure status Smart Grid applications present regarding its desired role in society and the functioning of future cities. This status has clearly limited the free access to real devices and tools to study PRIME, as it will be commented later on. Smarter equipment makes for greater opportunities but at the same time for greater risks if the service was controlled by the wrong hands.

5.1 OBJECTIVES

The PRIME Standard has been successfully analyzed with security in mind, examining its details and striving to find points that still pose a risk in any new scenario where the protocol is put into place in a network. Reporting has taken place dividing potential threats according to the style of attack, first covering passive interventions all the way to active traffic injections with diverse malicious and harmful intentions in mind until our attack style of choice.

The most vulnerable mechanisms from a DoS point-of-view contained in the standard have been highlighted, including Security Profile 0 usage for communications, node registration mechanisms and Base Node weaknesses.

While this analysis is limited by the style of attack in place, positive results have come out of the study, although leaving plenty of space for improvement as well as attack design.

Again, designing threats for a full implementation of a Smart Grid network running PRIME is subjected by the decision to focus on Denial of Service-type of attacks. However, even with this conditions, an adequate structure and planning has been shown in order to create up to four design and implementation plans for attacks with relative expectations of success.

Tackling the most vulnerable sections of the standard has led to two attacks focusing on the Node Registration mechanisms of PRIME, the first one harming the process by overflowing Node Identifiers and the second one detailing the mechanism to impersonate correctly-working nodes in order to hijack their position. Afterwards, another two offensive mechanisms came up by overflowing the use of the LSID field by overcomplicating the topological structure of the network, first, and then spoofing Base Node to create harm in a network combining two strategies shown in the study section of the thesis.

5.2 FUTURE WORK

Despite the completed objectives of this study, several points have been left to be done along the lines of this project to further expand over its work.

Initially, the idea as how far this study could go did not consider theoretical developments and analysis, and aspired to create a real attacking environment over a functioning Smart Grid system.

As exposed in various occasions during this thesis, the possibility to experiment with a real-life environment of a PRIME Smart Grid network is, at least, limited, if not only available to manufacturers. This, together with the realization that much study of the standard on the authors' part was pending before any attack could be successfully designed, shifted the focus of this project towards the study that is now.

Therefore, if this work were to be continued, it would complete these aspects of the total idea left unexplored. This is, once the detailed study of the standard is finished and the attack implementations are design adequately, use the tools available to create a simulation of a PRIME network in which to test out these mechanics.

Detailed previously is the setup used in UC3M for this kind of experiments, which uses a semi-simulated environment but still counts with PRIME Nodes and concentrators to work with. These elements are the ones this study could not count on

If such elements had been available physically or in a software fully-simulated environment, the study would have gone forward to try out the designed attacks. This work would have attempted to create a "lack of service" situation for the simulated working PRIME network, testing its resistance to DoS attacks with the harmful implementations detailed in the last part of the current study.

The cyber-physical infrastructure for the attack would be the same as explained here, where a malicious host connects to the last mile of the Smart Grid (meters). There, it would effectively inject the needed traffic in order to create the desired effect.

The use of a packet crafting tool (*Scapy*) could have been employed to create PRIME-like frames to insert into the network via, for instance, a *pcap* interface. Later, using a network sniffing application such as Wireshark, connected between the attacking host and the victim nodes, we could obtain the readings and packet traces corresponding to the attack to effectively certificate it has taken place and the network has been taken down.

If a complete implementation of a simulated environment were to take place in the context of this work, a realistic simulation of a PRIME-based PLC architecture would need the inclusion of physical phenomena, channel attenuation, multi-path effects and packets of variable length. This last feature would need a set of Matlab calculation tools, as it is the case with similar simulations taking this into account. The use of OMNeT++ network simulation is valuable when modeling the telematic effects of the communication process. Finally, the implementation of DLMS/COSEM as application layer is necessary for complementing the specification of PRIME.

BIBLIOGRAPHY

[1] PRIME Alliance AISBL. *PRIME Alliance Deployements*. 2017. URL: http://www.prime-alliance.org/?page_id=492 (visited on 0008–2018).

[2] Eduardo Alonso, Javier Matanza, Carlos Rodriguez-Morcillo, and Sadot Alexandres. «A Switch Promotion Algorithm fro Improving PRIME PLC Network Latency.» In: *IEEE International Symposium on Power Line Communications and its Applications* (2014), pp. 1–6.

[3] Concha Mora de Amarillas, Javier Matanza Domingo, and Gregorio López López. *Implementación y evaluación del período libre de contienda en PRIME*. 2017. URL: https://www.smartgridsinfo.es/comunicaciones/comunicacion-implementacion-evaluacion-periodo-libre-contienda-prime (visited on 0008–2018).

[4] Tecnologías de la Información y Telecomunicaciones Observatorio Industrial del Sector de la Electrónica. *Smart Grids y la Evolució de la Red Eléctrica.* 2011. URL: http://www.mincotur.gob.es/industria/observatorios/SectorElectronica/Actividades/2010/Federaci\%C3\%B3n\%20de\%20Entidades\%20de\%20Innovaci\%C3\%B3n\%20y\%20Tecnolog\%C3\%ADa/SMART_GRIDS_Y_EVOLUCION_DE_LA_RED_ELECTRICA.pdf (visited on 0006–2018).

[5] European Commission's Smart Grids Task Force. *Smart grids and meters*. 2018. URL: https://ec.europa.eu/energy/en/topics/markets-and-consumers/smart-grids-and-meters (visited on 0007–2018).

[6] Adam Gauci, Sandeep Pathania, Mathieu Salles, and Jose Manuel Ruiz. *Análisis y Evolución de Ciberseguridad en el IoT de Infraestructuras Críticas Eléctricas*. 2018. URL: https://www.smartgridsinfo.es/comunicaciones/comunicacion-analisis-evolucion-ciberseguridad-iot-infraestructuras-criticas-electricas (visited on 0006–2018).

[7] Manuel Sánchez Gómez-Merelo. *Infraestructuras Críticas y Ciberseguridad*. 2011. URL: https://manuelsanchez.com/2011/07/06/infraestructuras-criticas-y-ciberseguridad/ (visited on 0006–2018).

[8] PRIME Alliance Technical Working Group. «Draft Specification for PoweRline Intelligent Metering Evolution.» In: (2013), pp. 56–146.

[9] Donald E. Knuth. «Computer Programming as an Art.» In: *Communications of the ACM* 17.12 (1974), pp. 667–673.

[10] Mehdi Korki, Cishen Zhang, and Hai L. Vu. «Performance Evaluation of PRIME in Smart Grid.» In: *IEEE SmartGridComm 2013 Symposium - Communication Networks for Smart Grids and Smart Metering* (2013), pp. 1–6.

[11] GRID Innovation Online. *InovGrid Project - EDP Distribuicão (Portugal)*. 2017. URL: http://www.gridinnovation-on-line.eu/articles/library/inovgrid-project---edp-distribuicao-portugal.kl (visited on 0008–2018).

[12] Inmaculada Revuelta Pérez. *Regulando las Smart Grids: Las "Mejores Técnicas Disponibles" en privacidad y seguridad*. 2018. URL: https://www.smartgridsinfo.es/comunicaciones/regulando-smart-grids-mejores-tecnicas-disponibles-privacidad-seguridad (visited on 0007–2018).

[13] Grupo Tecma Red S.L. *Dossier de Resultados: IV Congreso Smart Grids*. 2017. URL: https://www.congreso-smartgrids.es/wp-content/uploads/Congreso-SG-4-2017-Informe-Resultados.pdf (visited on 0008–2018).

[14] Miguel Seijo Simó, Gregorio López López, and José Ignacio Moreno Novella. *El reto de la ciberseguridad en Infraestructuras de Medición Avanzada*. 2017. URL: https://www.smartgridsinfo.es/comunicaciones/comunicacion-reto-ciberseguridad-infraestructuras-medicion-avanzada (visited on 0006–2018).

[15] Miguel Seijo Simó, Gregorio López López, and José Ignacio Moreno Novella. «Cybersecurity Vulnerability Analysis of the PLC PRIME Standard.» In: *Security and Communications Network* (2017), p. 18.

[16] Agencia Tributaria. *Tabla de coeficientes de amortización lineal*. 2015. URL: https://www.agenciatributaria.es/AEAT.internet/Inicio/_Segmentos_/Empresas_y_profesionales/Empresas/Impuesto_sobre_Sociedades/Periodos_impositivos_a_partir_de_1_1_2015/Base_imponible/Amortizacion/Tabla_de_coeficientes_de_amortizacion_lineal_.shtml (visited on 0009–2018).

[17] Universidad Internacional de Valencia. *¿Qué se considera una infraestructura crítica?* 2018. URL: https://www.universidadviu.es/se-considera-una-infraestructura-critica/ (visited on 0005–2018).

[18] Javier Vazquez Vidal and Alberto Garcia Ilera. *Lights Off! The Darkness of the Smart Meters*. BlackHat Europe. 2014.

[19] Willis Towers Watson. *CompSource Market Compensation Data*. 2018. URL: https://compsource.towerswatson.com (visited on 0009–2018).

[20] edp distrib. *InovGrid*. 2017. URL: https://www.edpdistribuicao.pt/pt/rede/InovGrid/Pages/InovGrid.aspx (visited on 0008–2018).

[21] smartgridsinfo.es. *El 62% de los contadores existentes en España son inteligentes*. 2017. URL: https://www.smartgridsinfo.es/2017/03/16/62-los-contadores-existentes-inteligentes (visited on 0007–2018).

DECLARATION

Put your declaration here.

Madrid, September 2018

Jorge Erustes de la Vega

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede and Ivo Pletikosić. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/