

Security Modeling of Cyber-Physical Systems

A Case Study of Smart Grid

Mozhgan Pourabedin Islami

Master's Thesis



UiO : **Department of Informatics**
University of Oslo

Supervisors:

Dr. Phu Hong Nguyen

A. Prof. Dr. Tao Yue

Dr. Shaukat Ali

Oslo, Spring 2017

MASTER'S THESIS

**Security Modeling of Cyber-Physical Systems: A
Case Study of Smart Grid**

by

Mozhgan Pourabedin Islami

Supervisors:

Dr. Phu Hong Nguyen

A. Prof. Dr. Tao Yue

Dr. Shaukat Ali

Oslo, Spring 2017

© Mozhgan Pourabedin Islami

2017

Security Modeling of Cyber-Physical Systems: A Case Study of Smart Grid

<http://www.duo.uio.no/>

Department of Informatics, Universitetet i Oslo

Abstract

With the progress of the digital age, Cyber Physical Systems (CPSs), which are the integration of physical and computational world, are becoming more popular. Security of CPSs is important, as well and it should be considered. Because if the security is not considered, it can lead to some problems such as attackers can cause city blackouts. However, CPSs are often very complex systems and making sure of their security is very challenging. CPSs can be found in many key areas such as transportation, healthcare, and energy. One of the great instances of CPSs is smart grids. Smart grids generate the electricity power and transmit the power to different category of customers. One of the basic parts of smart grid systems is advanced metering infrastructure (AMI). Smart meter is another basic part of smart grid, which measures the power consumption. The goal of this thesis is first to model the basic functionalities of AMI system, then to model security aspects and requirements of AMI system to address security threats and challenges. We will also model some security-related uncertainties of AMI system. Modeling could help capturing the uncertainties in CPSs that might have huge impacts in people's lives and economic state of the society.

At a high-level, the contributions of this thesis are: 1) Modeling core functionalities of AMI system. We need to understand what the main functionalities of AMI system are. We specified and modeled three important cases of AMI system's core functionalities. These are: a) Periodic meter reading; b) Remote meter connect/disconnect; c) On-demand meter reading. 2) Modeling security aspects of AMI system or requirements. There are some security requirements for AMI system. The main requirements are Confidentiality, Integrity, and Availability. We address these security requirements by designing some security mechanisms such as Authentication, Authorization and Encryption/Decryption. 3) Modeling some security-related uncertainties of AMI system. There are some uncertainties in the functionalities of AMI system that can lead to vulnerabilities.

The main methodology used in this thesis to model core functionalities of AMI system, its security aspects and uncertainty is Unified Modeling Language (UML). There are different diagrams in UML. The main diagrams, which we use in this thesis to do modeling are use case diagrams, class diagrams, sequence diagrams and state chart diagrams. We use IBM RSA (Rational Software Architect) tool to model UML diagrams.

Finally, we conclude the thesis and we propose the future work that can be done in the area of AMI system's security and uncertainty in specific and CPSs in general.

Acknowledgment

I would like to thank my supervisors Professor Tao Yue, Dr. Phu Hong Nguyen, and Dr. Shaukat Ali for their advice and guidance throughout this thesis. Without their advice and guidance, I could not finalize this thesis. Special thanks to Dr. Phu Hong Nguyen for his continuous support through the weekly meetings and discussions. Dr. Nguyen helped me all the time during this thesis with his knowledge, motivation and steering the thesis in the right direction. I would also like to thank for his patience during my absence due to family circumstances. I would not have imagined a better advisor and mentor for my master thesis.

Finally, I would like to thank my family, my husband, my parents, and in-laws for their continuous support and encouragement throughout my studies. I am grateful for your moral and emotional support in my life.

To my husband and my child

Glossary

CPS: Cyber Physical System

AMI: Advanced Metering Infrastructure

CIA: Confidentiality, Integrity, and Availability

MDMS: Meter Data Management System

MDE: Model Driven Engineering

MDA: Model Driven Architecture

OMG: Object Management Group

CIM: Computational Independent Model

PIM: Platform Independent Model

PSM: Platform Specific Model

UML: Unified Modeling Language

DSL: Domain Specific Language

UML Profiles: Unified Modeling Language Profiles

OCL: Object Constraint Language

RUCM: Restricted Use Case Modeling

UCSs: Use Case Specifications

RFS: Reference Flow Step

UMF: Uncertainty Modeling Framework

UUP: UML Uncertainty Profile

UMLsec: Unified Modeling Language Security

PLC: Power Line Communication

IBM RSA: IBM Rational Software Architect

CIS: Customer Information System

RBAC: Role Based Access Control

PDP: Policy Decision Point

PEP: Policy Enforcement Point

MMB: Meter Metrology Board

NIC: Network Interface Component

CMAC: Cipher-Based Message Authentication Code

DOS: Denial of Service

IS: Input Stream

OS: Output Stream

RCD: Remote Connect Disconnect

UC: Unit Commitment

ED: Economic Dispatch

CONTENTS

ABSTRACT	V
ACKNOWLEDGMENT	VII
GLOSSARY	IX
1 INTRODUCTION	1
1.1 Overview	1
1.2 Motivation.....	1
1.3 Research questions	3
1.4 Expected outcome	5
1.5 Thesis structure	5
2 BACKGROUND	6
2.1 Model Driven Engineering (MDE)	6
2.1.1 Model Driven Architecture (MDA).....	7
2.1.2 Domain Specific Language (DSL)	7
2.1.3 Model Transformation.....	8
2.2 Modeling Techniques.....	8
2.2.1 Unified Modeling Language (UML)	9
2.2.2 Unified Modeling Language (UML) Profiles.....	13
2.2.3 Object Constraint Language (OCL).....	15
2.3 Restricted Use Case Modeling (RUCM).....	16
2.4 Cyber Physical systems.....	17
2.5 Smart Grids	18
2.6 Security Modeling.....	19
2.7 Security of Cyber-Physical Systems	20
2.8 Security and Security Requirements of Smart Grids.....	20
2.9 Uncertainty.....	21
3 RELATED WORK	23
3.1 Unified Modeling Language Security (UMLsec).....	23
3.2 Model at Run Time Security Handling	24
3.2.1 Topology of Smart Grid	24
3.2.2 Models@run.time	26
4 METHODOLOGY	27
4.1 Addressing Research Question 1 (RQ1)	27
4.2 Addressing Research Question 2 (RQ2)	27
4.3 Addressing Research Question 3 (RQ3)	28
5 CASE STUDY	29
5.1 Structure of Advance Metering Infrastructure	29
5.1.1 AMI Head-end.....	30
5.1.2 Smart Meters	30

5.1.3	Customer Information System (CIS)	30
5.1.4	Core Functionalities of AMI Head-End	31
5.2	Security Design of Smart Grid	32
5.2.1	Authentication	32
5.2.2	Authorization	33
5.2.3	Encryption and Decryption	35
5.3	Use cases of AMI Head-End	37
5.3.1	Table of Actors for AMI Head-End Use Cases	37
5.3.2	AMI Head-End's Initialization Use Case Diagram	37
5.3.3	AMI Head-End Establishes Connection with Smart Meter Use Case	39
5.3.4	Receiving Package from Smart Meter Use Case	41
5.3.5	Sending Package to Smart Meter Use Case	43
5.3.6	Decrypting Package Use Case	44
5.3.7	Response to Smart Meter Use Case	46
5.3.8	Encrypting Package Use case	48
5.3.9	Showing Acknowledgment from Smart Meter	50
5.3.10	Authentication Use case	51
5.3.11	Creating New Session Use case	53
5.3.12	Authorization Use Case	54
5.3.13	Periodic Meter Reading Use Case	57
5.3.14	Remote Meter Connect/Disconnect Use Case	58
5.3.15	On-Demand Meter Reading Use case	61
5.4	Use cases of Smart Meter	62
5.4.1	Table of Actors for Smart Meter Use Cases	62
5.4.2	Smart Meter Establishes Connection with AMI Head-End Use Case	63
5.4.3	Sending Package to AMI Head-End Use Case	65
5.4.4	Receiving Package from AMI Head-End Use Case	67
5.4.5	Encrypting Package Use case	69
5.4.6	Decrypting Package Use Case	70
5.4.7	Response to AMI Head-End Use Case	72
5.4.8	Authenticate Use Case	74
5.4.9	Periodic Meter Reading Use Case Diagram	76
5.4.10	Recording the Meter's Electrical Usage Data Use Case Diagram	78
5.4.11	Remote Meter Connect/Disconnect Use Case Diagram	80
5.4.12	On-Demand Meter Reading Use Case Diagram	83
5.5	Security Related Uncertainties of Smart Grid	86
5.5.1	Examples of Smart Grid Uncertainties	86
5.5.2	Mutation Operators	90
6	MODELING	92
6.1	Class Diagram	92

6.1.1	Class Diagram Description.....	92
6.1.2	AMI Head-End Class Diagram.....	92
6.1.3	Smart Meter Class Diagram	103
6.2	Sequence Diagram	111
6.2.1	Package Encryption Sequence Diagram.....	111
6.2.2	Package Decryption Sequence Diagram.....	113
6.2.3	Establish Connection Sequence Diagram.....	115
6.2.4	Smart Meter Authentication Sequence Diagram	117
6.2.5	Authorization Sequence Diagram.....	119
6.2.6	Periodic Meter Reading Sequence Diagram.....	120
6.2.7	Remote Meter Connect Sequence Diagram.....	122
6.2.8	Remote Meter Disconnect Sequence Diagram	124
6.2.9	On-Demand Meter Reading Sequence Diagram	126
6.2.10	Misuse Model Sequence Diagram for City Blackout Uncertainty	128
6.2.11	Misuse Model Sequence Diagram for Signing Package Uncertainty.....	130
6.3	State Chart Diagrams	132
6.3.1	Smart Meter Registration State Chart Diagram.....	132
6.3.2	Periodic and On-Demand Meter Reading State Chart Diagram	134
6.3.3	Remote Meter Connect/Disconnect State Chart Diagram	136
7	CONCLUSION	138
8	FUTURE WORK	140
9	REFERENCES	142

TABLE OF FIGURES

Figure 1 A Use case diagram of a University	10
Figure 2 A class diagram between student, professor and course	11
Figure 3 A Class diagram with Generalization relationship between classes	12
Figure 4 A state diagram for registration of a class	13
Figure 5 Example of UML profile [20].....	14
Figure 6 Tagged values in UML profile [20].....	15
Figure 7 Object Constraint Language (OCL).....	15
Figure 8 Smart Grid Architecture [33].....	19
Figure 9 Smart grid topology [42].....	25
Figure 10 Application-Level Reference Model for Advanced Metering Infrastructure [43].....	29
Figure 11 AMI Head-End's Initialization Use Case Diagram	38
Figure 12 AMI Head-End Establishes Connection with Smart Meter Use Case Diagram	40
Figure 13 Receiving Package from Smart Meter Use Case Diagram	41
Figure 14 Sending Package to Smart Meter Use Case Diagram	43
Figure 15 Decrypting Package Use Case Diagram	45
Figure 16 Response to Smart Meter Use Case Diagram	47
Figure 17 Encrypting Package Use Case Diagram	49
Figure 18 Showing Acknowledgment from Smart Meter Use Case Diagram	50
Figure 19 Authentication Use Case Diagram.....	52
Figure 20 Creating New Session Use Case Diagram	54
Figure 21 Authorization Use Case Diagram	55
Figure 22 Periodic Meter Reading Use Case	57
Figure 23 Remote Meter Connect/Disconnect Use Case Diagram	59
Figure 24 On-Demand Meter Reading Use Case Diagram	61
Figure 25 Smart Meter Establishes Connection with AMI Head-End Use Case Diagram	64
Figure 26 Sending Package to AMI Head-End Use Case Diagram	66
Figure 27 Receiving Package from AMI Head-End Use Case Diagram	68
Figure 28 Encrypting Package Use Case Diagram	70
Figure 29 Decrypting Package Use Case Diagram	71
Figure 30 Response to AMI Head-End Use Case Diagram	73
Figure 31 Authentication Use Case Diagram.....	75
Figure 32 Periodic Meter Reading Use Case Diagram	77
Figure 33 Recording the Meter's Electrical Usage Data Use case Diagram.....	79
Figure 34 Remote Meter Connect/Disconnect Use case Diagram	81
Figure 35 On-Demand Meter Reading Use case Diagram.....	84
Figure 36 AMI Head-End Class Diagram with Security Functionalities.....	93

Figure 37 Smart Meter Class Diagram with Security Functionalities.....	104
Figure 38 Package Encryption Sequence Diagram (AMI Head-End Side)	112
Figure 39 Package Encryption Sequence Diagram (Smart Meter Side)	113
Figure 40 Package Decryption Sequence Diagram (AMI Head-End Side)	114
Figure 41 Package Decryption Sequence Diagram (Smart Meter Side)	114
Figure 42 Smart Meter registration (Establish Connection) Sequence Diagram (AMI Head-End Side)	116
Figure 43 Smart Meter registration (Establish Connection) Sequence Diagram (Smart Meter Side)	117
Figure 44 Smart Meter registration (Smart Meter Authentication) Sequence Diagram (Smart Meter Side)	118
Figure 45 Smart Meter registration (Smart Meter Authentication) Sequence Diagram (AMI Head-End Side).....	119
Figure 46 Authorization Sequence Diagram (AMI Head-End Side)	120
Figure 47 Periodic Meter Reading Sequence Diagram (Smart Meter Side)	121
Figure 48 Periodic Meter Reading Sequence Diagram (AMI Head-End Side)	122
Figure 49 Remote Meter Connect Sequence Diagram (AMI Head-End Side)	123
Figure 50 Remote Meter Connect Sequence Diagram (Smart Meter Side)	124
Figure 51 Remote Meter Disconnect Sequence Diagram (AMI Head-End Side).....	125
Figure 52 Remote Meter Disconnect Sequence Diagram (Smart Meter Side).....	126
Figure 53 On-Demand Meter Reading Sequence Diagram (AMI Head-End Side).....	127
Figure 54 On-Demand Meter Reading Sequence Diagram (Smart Meter Side).....	127
Figure 55 Misuse Model Sequence Diagram for City Blackout Uncertainty (AMI Head-End Side)	129
Figure 56 Misuse Model Sequence Diagram for City Blackout Uncertainty (Smart Meter Side).	130
Figure 57 Misuse Model Sequence Diagram for Signing Package Uncertainty (AMI Head-End Side)	131
Figure 58 Misuse Model Sequence Diagram for Signing Package Uncertainty (Smart Meter Side)	132
Figure 59 Registration State chart diagram (AMI Head-End Side)	133
Figure 60 Registration State chart diagram (Smart Meter Side)	134
Figure 61 Periodic and on-demand Meter Reading State Chart Diagram (AMI Head-End Side) .	135
Figure 62 Periodic and on-demand Meter Reading State Chart Diagram (Smart Meter Side)	136
Figure 63 Remote Meter Connect/Disconnect State Chart Diagram (AMI Head-End Side).....	137
Figure 64 Remote Meter Connect/Disconnect State Chart Diagram (Smart Meter Side).....	137

LIST OF TABLES

Table 1 Table of Actors for AMI head-End Use Cases	37
Table 2 AMI Head-end's Initialization Use Case	38
Table 3 AMI Head-End Establishes Connection with Smart Meter Use Case	40
Table 4 Receiving Package from Smart Meter Use Case	42
Table 5 Sending Package to Smart Meter Use Case	44
Table 6 Decrypt Package Use Case.....	46
Table 7 Response to Smart Meter Use Case	47
Table 8 Encrypting Package Use Case.....	49
Table 9 Showing Acknowledgment from Smart Meter Use Case	50
Table 10 Authenticate Use Case	52
Table 11 Creating New Session Use Case	54
Table 12 Authorization Use Case.....	55
Table 13 Periodic Meter Reading Use case.....	58
Table 14 Remote Meter Connect/Disconnect Use case	60
Table 15 On-Demand Meter Reading Use case	62
Table 16 Table of Actors for Smart Meter Use Cases	63
Table 17 Smart Meter Establishes Connection with AMI Head-End Use Case	64
Table 18 Sending Package to AMI Head-End Use Case	66
Table 19 Receiving Package from AMI Head-End Use Case.....	68
Table 20 Encrypting Package Use Case.....	70
Table 21 Decrypting Package Use Case.....	71
Table 22 Response to AMI Head-End Use Case	73
Table 23 Authenticate Use Case	75
Table 24 Periodic Meter Reading Use Case.....	77
Table 25 Record the Meter's Electrical Usage Data Use case	79
Table 26 Remote Meter Connect/Disconnect Use case	81
Table 27 On-Demand Meter Reading Use case	84
Table 28 Class summary for AMI Head-End Class Diagram with Security Functionalities	93
Table 29 Attribute Summary for Class AMI_HeadEnd in AMI Head-End Class Diagram	94
Table 30 Method Summary for Class AMI_HeadEnd in AMI Head-End Class Diagram	95
Table 31 Attribute Summary for Class SmartMeterController in AMI Head-End Class Diagram ..	96
Table 32 Method Summary for Class SmartMeterController in AMI Head-End Class Diagram ...	96
Table 33 Method Summary for Class SessionManager in AMI Head-End Class Diagram.....	98
Table 34 Attribute Summary for Class Session in AMI Head-End Class Diagram.....	99
Table 35 Method Summary for Class Session in AMI Head-End Class Diagram.....	99
Table 36 Method Summary for Class Thread in AMI Head-End Class Diagram.....	100

Table 37 Attribute Summary for Class DataPackage in AMI Head-End Class Diagram	100
Table 38 Method Summary for Class DataPackage in AMI Head-End Class Diagram	101
Table 39 Attribute Summary for Class ConnectionHandler in AMI Head-End Class Diagram....	101
Table 40 Method Summary for Class ConnectionHandler in AMI Head-End Class Diagram.....	101
Table 41 Enumeration Summary for Class Package Code in AMI Head-End Class Diagram.....	102
Table 42 Method Summary for Class ServerSocket in AMI Head-End Class Diagram.....	103
Table 43 Class Summary for Smart Meter Class Diagram with Security Functionalities	104
Table 44 Attribute Summary for Class MeterMetrologyBoard in Smart Meter Class Diagram....	105
Table 45 Method Summary for Class MeterMetrologyBoard in Smart Meter Class Diagram.....	105
Table 46 Attribute Summary for Class NIC in Smart Meter Class Diagram.....	106
Table 47 Method Summary for Class NIC in Smart Meter Class Diagram.....	106
Table 48 Attribute Summary for Class DataPackage in Smart Meter Class Diagram.....	107
Table 49 Method Summary for Class DataPackage in Smart Meter Class Diagram	107
Table 50 Method Summary for Class Thread in Smart Meter Class Diagram	108
Table 51 Attribute Summary for Class InternalMeterSwitch in Smart Meter Class Diagram.....	108
Table 52 Method Summary for Class InternalMeterSwitch in Smart Meter Class Diagram.....	108
Table 53 Attribute Summary for Class ClientSocket in Smart Meter Class Diagram	108
Table 54 Method Summary for Class ClientSocket in Smart Meter Class Diagram	109
Table 55 Attribute Summary for Class RecordService in Smart Meter Class Diagram	109
Table 56 Method Summary for Class RecordService in Smart Meter Class Diagram	109
Table 57 Method Summary for Class Record in Smart Meter Class Diagram	109
Table 58 Attribute Summary for Class TableData in SmartMeter Class Diagram	110
Table 59 Method Summary for Class TableData in Smart Meter Class Diagram	110
Table 60 Enumeration Summary for Class PackageCode in Smart Meter Class Diagram.....	110

1 Introduction

Section 1.1 gives a general overview of the research topic, which is security modeling of cyber physical systems: A case study of smart grid. Then we present in section 1.2 our motivation for working on security modeling of Advanced Metering Infrastructure (AMI). In section 1.3, we propose some research questions. They are about key components and functionalities of AMI and security requirements of AMI. Section 1.4 is about expected outcome followed by thesis structure in section 1.5.

1.1 Overview

With the progress of software engineering, Cyber Physical Systems (CPS) are becoming more popular. CPS is the integration of computational and physical world. The security of CPSs is also becoming important. CPSs are the next generation of engineered systems. They could have huge impacts on human beings. Many CPSs are also more open, and more prone to security threats. Therefore, it is important to consider the security of CPSs. Some of the CPSs applications are energy, transportation, robotics, healthcare, manufacture, and military. Smart grid is one of the most important application domains of CPSs. Smart grid generates electricity power. It also transmits the power to different consumers. The consumers are houses, hospitals, offices, factories, etc. Smart grids are the modern power grids. Smart grids can enhance the efficiency and reliability aspects of power grids. Traditional power grid cannot communicate, whereas smart grids have advanced communication and computing power. Communication system is one of the key features of smart grid [1]. AMI, which stands for Advanced Metering Infrastructure is one of the key parts of smart grid. It enables the two-way communication between utility and smart meters. Tackling security for CPSs in general and AMI in particular is challenging. One of the challenges is that CPS and AMI are complex systems. Other challenge is that there is little research about security of CPS and AMI, because the concepts are new. One-step forward is to capture security concerns. We will address this in the thesis by modeling core functionalities of CPS or AMI. Then we will design the security aspects of AMI system and specify security-related uncertainties of AMI in addition to modeling core functionalities of AMI system.

1.2 Motivation

In this section, we show why we work on the topic of this master thesis, which is security modeling of AMI. We can start our motivation by explaining the importance of security for AMI. AMI is smart grid's key part enables bi-directional communication between utility and smart meters. Security of AMI is important, because if the security is not considered, it can lead to many problems. The example

can be that a hacker can access to smart grid. There are some security concerns for AMI. There are Integrations within a community and ability to impact consumer's privacy [2]. Smart meters are the other part of smart grids, which communicate with AMI. Smart meters are the digital version of the current power meters. Smart meters are installed at a customer's location. They measure electrical power usage called meter readings. Smart meters are connected to the smart grid. They send meter readings to the smart grid. These readings are used for electrical power state estimation and for billing purposes. There are some security challenges related to smart meters. Tampering with device functionality and communication issues between meter and power supplier are examples. Authentication and identity management in a distributed grid infrastructure also poses a challenge. AMI presents increased dependency on cyber resources, which may be vulnerable to attack [3]. For example, exploited vulnerabilities can result in takeover of devices by attacker. This can lead to crises like city blackouts that can have huge impacts in economy and people's lives [4].

There are some ways to protect cyber-attacks for AMI and address these challenges. These methods are encryption, physical controls, firewalls, etc. One way to secure AMI system is using specification-based intrusion detection. In this method, there is a sensor to check the AMI network's traffic [5].

There are also some main challenges of engineering security for CPSs. One of the challenges is when modern CPS wants to connect to the Internet. By this connection, the worms can be introduced to the system and have impacts on the CPS.

Model based security engineering is a solution to handle security challenges of CPSs. Motivation for using models is that because CPSs are complex systems, modeling gives more high level of abstraction than coding. This would lead to better security engineering of the system. By modeling, security requirements: confidentiality, integrity, and availability can be considered as early as possible.

We address some challenges of AMI in this thesis by using model based security engineering. In chapter 5, there are some use cases with their specifications in form of use case templates. These use cases are about AMI basic functionalities. These functionalities are periodic meter reading, on-demand meter reading and remote meter connect/disconnect. We will also work on security aspects like confidentiality, integrity, and availability of these use cases. Then in chapter 6, we will map the use cases to the class diagrams to show the main functionalities of AMI and some security aspects. There are some security solutions or mechanisms to cover security requirements. These mechanisms, which mentioned in this thesis are encryption, decryption, authentication and authorization. We use these mechanisms in sequence diagrams and state chart diagrams, as well in our thesis.

There are three main security requirements, which are Confidentiality, Integrity, and Availability. These form the CIA term. Confidentiality means that the information is not accessible by unauthorized people. Integrity means that unauthorized people cannot change or delete information. Availability

means that the information is only accessible by the authorized people. During design of smart grids, these three important security requirements should be considered. The sensitive information should not be accessible by unauthorized people or malicious attackers. Unauthorized people should not change the sensitive information.

There are also some issues about uncertainty in the security of CPSs. We should consider uncertainty in CPS because of its impacts on security problems. Uncertainty in functionalities of CPSs can lead to security vulnerabilities in system. This is one of the possible impacts of uncertainty to security problems. These vulnerabilities can have effects on exploiting by attackers or malicious users. Security attacks could also lead to uncertainties in CPSs' functionalities. Therefore, to tackle with these uncertainties, model based security engineering should be focused. It provides a model foundation for reasoning about security-related uncertainties of CPSs, and AMI.

1.3 Research questions

RQ1: What are the key components of the Advanced Metering Infrastructure (AMI) of a smart grid and their security requirements?

The purpose of this question is to identify the key parts of AMI and to understand the business logic of AMI. The other purpose of this question is to know about the security issues and requirements of AMI. To answer to this question, we collect and synthesize functional and security requirements of AMI from different sources. The sources are NISTIR: Guidelines for Smart Grid Cyber Security document [3], advanced metering infrastructure conducted by US Department of Energy Office of Electricity and Energy Reliability [6] and cyber security issues for advanced metering infrastructure (AMI) by F. M. Cleveland [7].

AMI system consists of the following key parts: smart meters, communications infrastructure, local area networks, meter data management system (MDMS) and operational gateways [6]. There are also other components such as AMI head-end and AMI network. Security requirements for AMI system are Confidentiality, Integrity, Availability [3] and Accountability (non-repudiation).

Confidentiality in AMI systems

Privacy is the main issue for confidentiality in AMI systems at the customer site. Customers do not want their personal information like their energy consumption information be public. They want this information be confidential and accessible only by authorized people. This information should not also be accessible over the AMI network. For example, one customer should not see another customer's energy consumption information [7].

Integrity in AMI systems

Integrity in AMI systems means that there should not be any unauthorized control command sent from AMI system to smart meter. There can be security attack like a hacker can send disconnect commands to millions of smart meters. Other parts of AMI system like AMI network and AMI head-end's integrity aspect must be considered. There are some kinds of threats in AMI head-end. An example is "disgruntled employee" threat where the employees make severe damages before the threat is detected [7].

Availability in AMI systems

Availability is also important issue in AMI systems nowadays but in the past, it was not a big issue. Availability means that the data can be available only to authorized people. There are some causes, which make lack of availability [7]. These causes can be cyber tampering, invalid access, internal communication [8].

Accountability (Non-repudiation) in AMI systems

Accountability is also an important issue in AMI systems. In smart meters, AMI network and AMI head-end, the information should not be repudiated. That means the receiver should not deny the reception of information [7].

RQ2: How can the AMI key functionalities be specified and modeled together with the security requirements?

The goal of the question is using models to capture AMI functionalities and security concerns. We use UML for this task, because it is standardized modeling language used in industry. UML profile can be developed for modeling security concerns of AMI.

To answer this question, we first need to know what AMI key functionalities are. The core functionalities of AMI are metering services. The examples are periodic meter reading, on-demand meter reading and remote connect/disconnect of meter. These functionalities are given in chapter 5 in form of use case diagrams and use case templates. There are other types of UML diagrams for describing AMI core functionalities in the thesis such as class diagrams, sequence diagrams, and state chart diagrams, which are given in chapter 6 of the thesis.

We can combine these functionalities with security requirements. For example, in periodic meter reading the integrity of data is important. When smart meter sends meter data to AMI head-end, integrity of data should be considered. It means the data should not be changed during this

transmission. To model security together with functionalities of AMI, we leverage security-modeling techniques such as UMLsec, security patterns that can be fit into the AMI case study.

RQ3: What are some possible security-related uncertainties in AMI and how they can be modeled?

There are some security-related uncertainties in AMI. There are some uncertainties in functionalities of AMI, which might cause vulnerabilities to malicious attacks. The other uncertainty can be about specification, implementation, and evolution of security mechanisms. These can lead to other types of uncertainties in the functionality of AMI. An example can be incorrect access control, which can disable some physical processes. Security-related uncertainty of CPSs and especially AMI is worth to be investigated. This thesis tries to identify and understand a few security-related uncertainties of AMI. We will also attempt to model some of them.

1.4 Expected outcome

In this thesis, some expected outcomes could be the UML models such as use case specification, UML class diagrams, UML sequence diagrams, and UML state chart diagrams that specify the core functionalities of AMI. Second, there will be some UML models developed for specifying security concerns of AMI. Third, we will report on some security-related uncertainties of AMI and our attempt is to model at least one of them.

1.5 Thesis structure

Chapter 2 is about Background. There are some sections in this chapter such as model driven engineering, modeling techniques like UML, UML profile, and OCL, RUCM, cyber physical systems, smart grids, security modeling, uncertainty, etc. Chapter 3 is about related work. It is about UMLsec, model@run.time for smart grid security, etc. Chapter 4 describes the methodology. Chapter 5 is the case study, which is divided into some sub subsections about structure of AMI, security design of smart grid, some use cases of AMI head-end and smart meter, and security related uncertainties of smart grid. Chapter 6 is modeling. There are some subsections in this chapter for class diagrams, sequence diagrams and state chart diagrams. Chapter 7 is conclusion followed by future work in chapter 8.

2 Background

In this chapter, we present key background methodologies and concepts used in the thesis. The examples are Model Driven Engineering (MDE), modeling techniques, restricted use case modeling, cyber physical systems, smart grids, and security modeling. Other examples are security of CPSs, security requirements of smart grids and uncertainty.

In Section 2.1, we present MDE. It consists of model driven architecture, domain specific languages and model transformation. Section 2.2 is about modeling techniques. It is divided to unified modeling language, UML Profiles, and object constraint language subsections. UML is also divided to sections of use cases, class, sequence, and state chart diagrams. Section 2.3 is about Restricted Use Case Modeling (RUCM) used for use case diagrams' specifications. Section 2.4 gives the background about cyber physical systems followed by background of smart grids in section 2.5. Section 2.6 is about security modeling such as UMLsec. Section 2.7 is about security of cyber physical systems followed by security and security requirements of smart grids in section 2.8. Finally, uncertainty is in section 2.9.

2.1 Model Driven Engineering (MDE)

During last decades, the use of software has been increased. We use software everywhere, for example, in education systems, banking, transportation, engineering, medical equipment, etc. We book a hotel online, we shop online, etc. The software is not only running in traditional computers. We use software in smart phones and other devices as well. Some modern cars can have millions lines of codes [9].

The software engineering society faces with some challenges. Some challenges are how to maintain these million lines of codes and how to make sure they are out of errors and they are correct. Therefore, the abstraction seems a reasonable answer.

Here, the Model Driven Engineering (MDE) methodology comes to place. MDE means using models instead of programs. Example of modeling language is UML, which is a graphical modeling language. The main principle of MDE is that "Everything is model" [10].

MDE is a software development method. We use models and abstractions instead of writing codes to deal with complexity. It reduces complexity and increases automation in program development. MDE has some advantages to use. The reason we use model driven engineering can be these factors:

- It would increase the quality. It is supposed to be less error-prone, because we use modeling instead of programming and writing codes.
- MDE could be cost effective. Designing models can take shorter time than writing lines of codes. We can design models at a lower cost.

2.1.1 Model Driven Architecture (MDA)

MDA stands for model driven architecture. It is proposed by OMG (Object Management Group). MDA is a software development method. MDA is a specialization of MDE and it focuses on UML based modeling languages. MDA can be seen as OMG's vision on MDE [11].

MDA has three abstraction levels. These are computational independent model (CIM), platform independent model (PIM), platform specific model (PSM). These layers can be transformed to each other by using model transformations. For example, CIM can be transformed to PIM and PIM can be transformed to PSM [11].

There are differences between MDE and MDA. MDA is more restrictive than MDE, taking more attention to UMLs. However, there are also differences between MDA models and UML models. MDA models have formal meaning or semantics in contrast with UML models. The three goals of MDA are portability, interoperability, and reusability [11].

2.1.2 Domain Specific Language (DSL)

DSL is a short form of Domain Specific Language. It is a high-level language designed for a special domain. DSL is in contrast with general purpose modeling languages such as UML.

DSLs are small languages. They are easier to program. They are designed to solve the problems only in special domains. DSLs are simple, expressive languages and they should be understandable. They are better for describing things within the domain. The code generates from DSL is reliable and the system can be updated. The reason we use domain specific language is that it is for specific domains. We have two types of domain specific languages. One type is Internal DSL, which is also called embedded DSL. The other type is external DSL, which is an independent language [12].

Domain specific languages have some advantages and some disadvantages. Advantages can be that they increase productivity, maintainability, reliability, and portability. Besides that, they can be reused for different purposes. Disadvantage is that the cost is huge for designing and implementing. Besides that, it costs more to teach DSL to programmers and developers. Therefore, it requires more time, effort and cost to teach the DSL [13].

Example of DSL is SQL, which is database or query language. Other examples are HTML, Latex, and XML. DSLs can improve productivity and can promote better communication with customers. DSLs are used to create models and they usually use graphics.

2.1.3 Model Transformation

Model transformation plays a great role in Model Driven Engineering. Model is the simplified representation of the system. Models can be graphical things like UML diagrams. Model transformation means to transform one model to another model. Transforming a source model to a target model is an example of model transformation

There is model-to-model or model to text transformations. There is also model to platform transformation. This is usually called model to code transformation. Model-to-model transformation generates a model. Model to text and model to code generate text and code, respectively. Model transformations can be unidirectional or bidirectional. Unidirectional transformations transform source model to target model. Bidirectional transformations can transform source model to target model and vice-versa [14]. Example of model transformation is transforming platform PIM to PSM in Model Driven architecture [14]. There are different relationships between source model and target model in model transformation. In some model transformations, source and target models are same. In others, source and target models are different [15].

2.2 Modeling Techniques

In this part, we will discuss modeling techniques such as:

- UML
- UML Profiles
- OCL

Moreover, in more detail we will discuss about some diagrams of UML model such as:

- Use case diagrams
- Class diagrams
- Sequence diagrams
- State chart diagrams

Before we go in more detail, first we define the modeling and in specific UML modeling. A model represents abstraction of functionality, structure, or behavior of a system. We use graphical representations for models to show interconnection with a system we represent. We have graphical

models such as UML. It is described in more details in the next sections. We use models to understand the system under development.

2.2.1 Unified Modeling Language (UML)

Trends have shown in recent years, modeling becomes popular in area of software engineering. Models are usually graphical things. In this case, the definition of UML takes place as mentioned below.

UML stands for Unified Modeling Language, which is a graphical representation of the system. It gains popularity in the recent years. UML is OMG standard. It is one of the most popular modeling languages. There are different types of UML diagrams. Examples are class diagrams, state chart diagrams, use cases, activity diagrams, and sequence diagrams. We will discuss four important UML diagrams: use case, class, sequence, and state chart diagrams. One of the advantages of UML is that developers can learn it easily [16].

We use UML when we want to model the system and not only do programming tasks. UML is available to everyone and software industries have freedom to use it. UML has range of usage. It can be used in business modeling or software modeling [16].

UML profile is another subject of UML. It is the extended form of UML diagram. It has concepts of stereotypes, constraints, and tagged values.

Use Case Diagram

Use case diagram is one type of UML diagrams. It is used for showing the interaction between user and the system.

In use case diagrams, there are some notations. One is the system or system boundary. For example, bank's ATM can be a system for doing some operations such as withdrawing cash, payment, etc. Other example is university where students register for classes, do assignments, check exam results. The other notation is use cases. Use cases are shown in elliptic shape and they show the actions or operations, which are done in the system. There are other types of use cases such as extend and include use cases. The last notation is actors, which are people or things showed up as roles to interact with the system. Here, there is an example of use case diagram with university as a sub-system and student and professor as actors. There are different use cases inside the university sub system. These are actions or operations performed by a student and a professor.

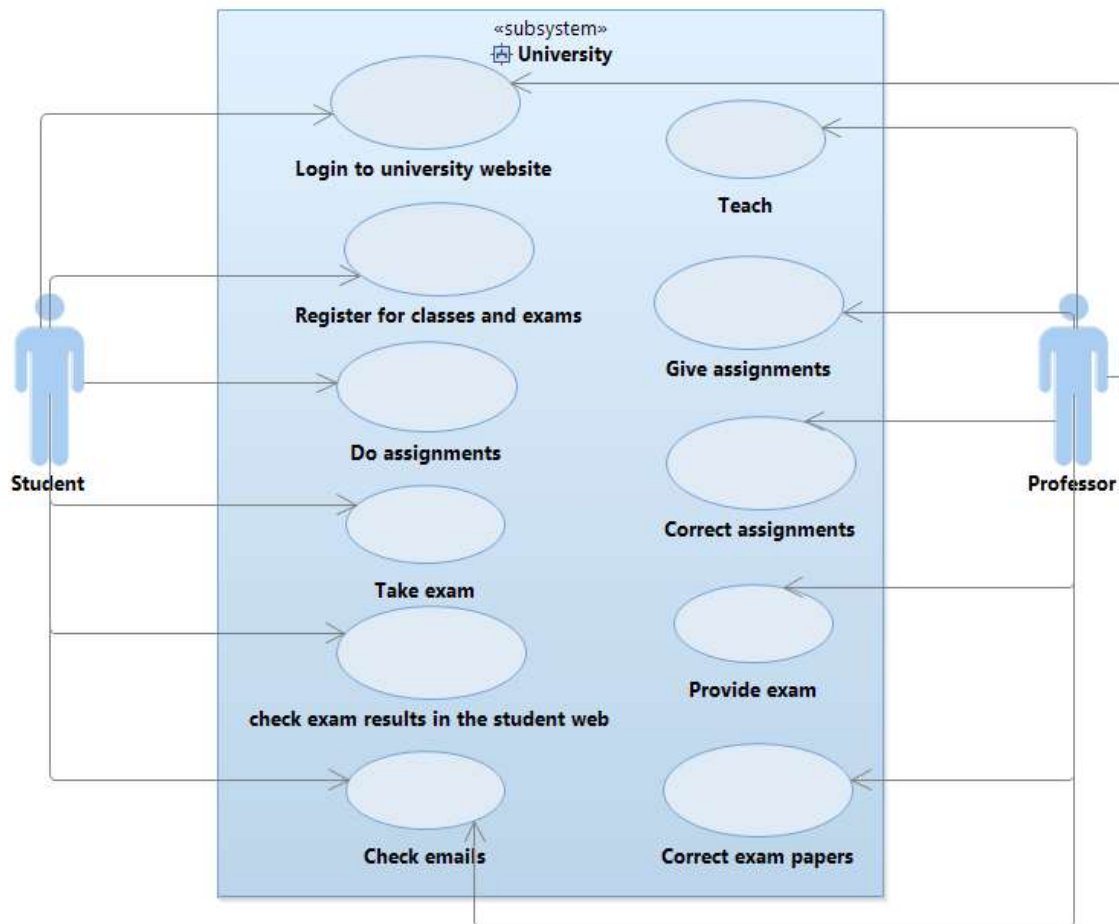


Figure 1 A Use case diagram of a University

We specify use case diagram by use case template. It is a table with fields such as use case name, use case description, and other information.

Class Diagram

Class diagram is the other type of UML diagram. Class diagram describes the entities or objects of classes with the relationship between them. There are some classes in the class diagram. These classes have different types of communications with each other. The examples of these communications are association, aggregation, composition, generalization, etc. Association is the most used type of relationship between classes. In this relationship, two classes associate with each other. Aggregation is a stronger type of association. In this relationship, one class is a part of the other class, but with no strong dependency. Composition is a stronger type of aggregation with strong dependency. In composition, existence of one class depends on the existence of the other class. Generalization is other type of relationship. In generalization, one or several classes inherit from other class. Classes are shown in rectangles. Each class consists of three parts. The upper part of the class is the name of the

class. The middle part is some attributes with their data types. The example for attribute is name: string. The last part is some operations, which each class can perform [16, 17].

Each class can interact with other class through associations. In each association, there are multiplicities for example, from 0...1 at one end to 1...* at the other end. The multiplicity between student and professor in the example below is 1...* at both ends. It means each student can have one to many professors and each professor can have one to many students. Below is the example of a class diagram in a university:

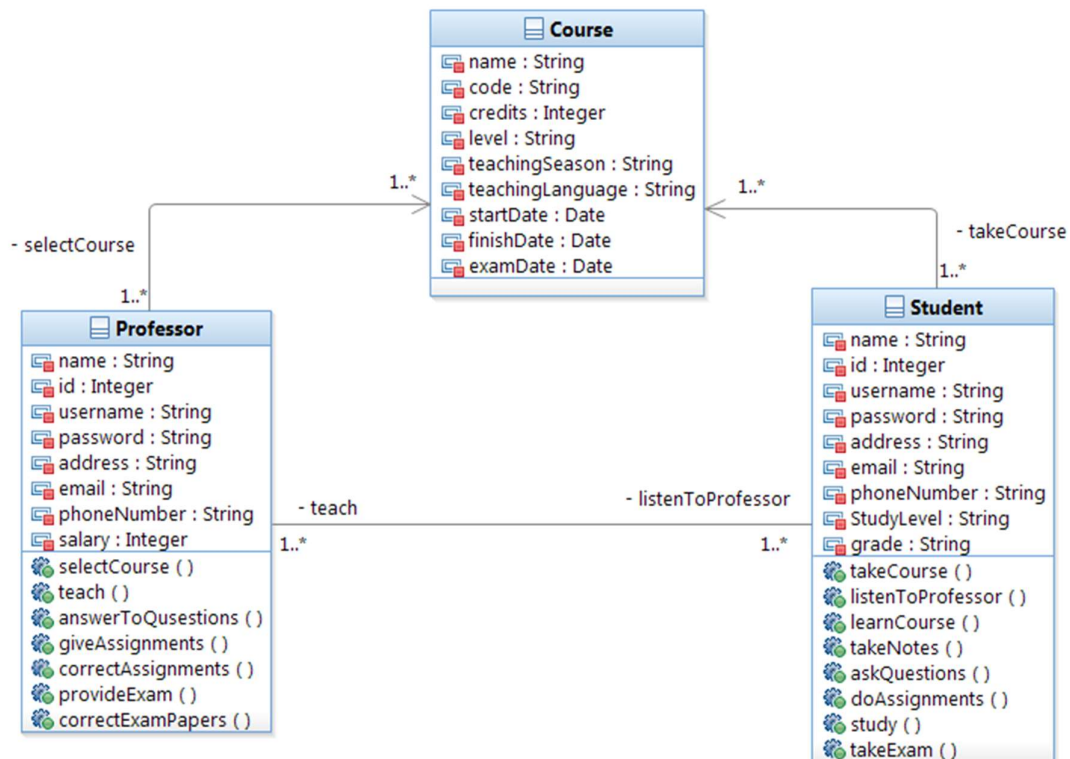


Figure 2 A class diagram between student, professor and course

There are also generalizations in class diagrams. Generalization means a class or some classes can inherit from its parent class. For example, class student and professor in the example below inherit from class person. Therefore, class person is the parent of both class student and professor. This relationship between class student and person or class professor and person is generalization.

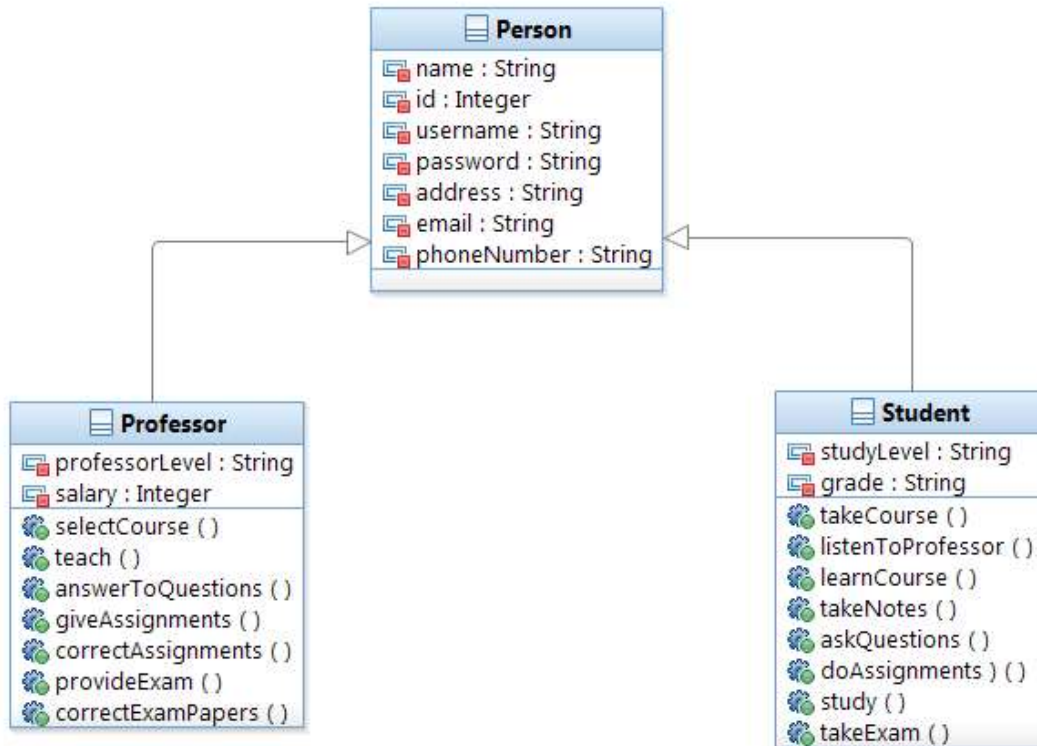


Figure 3 A Class diagram with Generalization relationship between classes

Sequence Diagram


A sequence diagram is a type of UML diagram. It is one of the interaction views in UML. We use sequence diagram for showing the dynamic behavior of the system [18]. It describes the sequence and history of actions that happen in a period over a system. The sequence diagram consists of some parts such as objects, messages, lifelines. There are some structures called combined fragments for showing loops, conditions, and parallel fragments. Objects are elements represented as roles to interact with other objects through messages. They are shown in a rectangle [19]. There is a lifeline for each object. It is a vertical line connected to object to represent entire interactions of the object. In the sequence diagram, each object interacts with another object through messages. The messages are shown as a horizontal line with arrows. There are different types of messages and based on the type, the arrows can be different. Combined fragments are other part of sequence diagram. They show different structures like loops, alternative paths, parallel actions, conditions, etc. They are shown as nested rectangles [18].

The messages are shown in order. For example, the first message is located upper than other messages. We draw sequence diagrams based on use case and class diagrams. For each use case diagram, we define one sequence diagram. Objects of sequence diagrams are usually the actors in use case diagram

or some of the classes in class diagram. The messages are usually the method calls. The names of messages are the names of methods of classes in class diagram.

State Chart Diagram

State chart diagram is the other type of UML diagram [16]. In the state chart diagram, there are states such as initial state and finish state. Initial state is drawn in a black circle: ●. Finish state is drawn in white and black circle as shown like this: ○

There are other states between start and end state. They are shown by rounded rectangles: 

There are transitions between states, which are similar to associations in class diagrams. These transitions transit one state to another state and connect states to each other.

Below is the example of a state diagram for the university between professor and student:

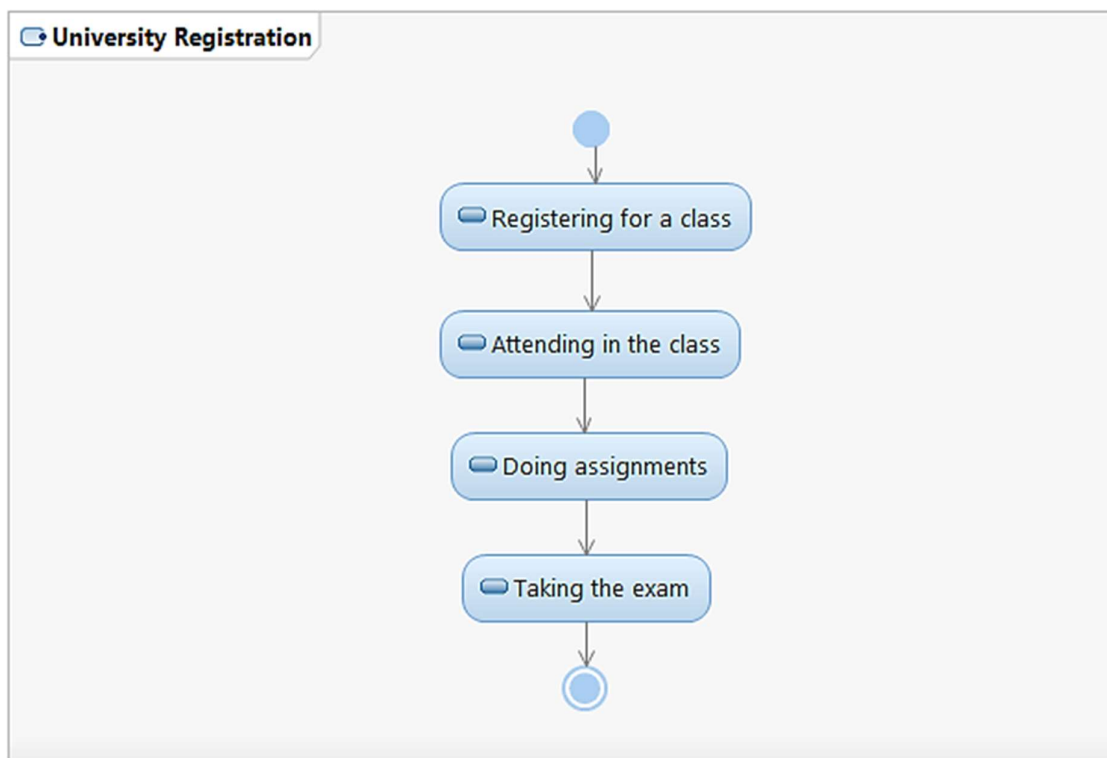


Figure 4 A state diagram for registration of a class

2.2.2 Unified Modeling Language (UML) Profiles

UML profiles are UML extensions that are used for specific domains and have these parts: Constraints, Stereotypes, and Tagged Values [20].

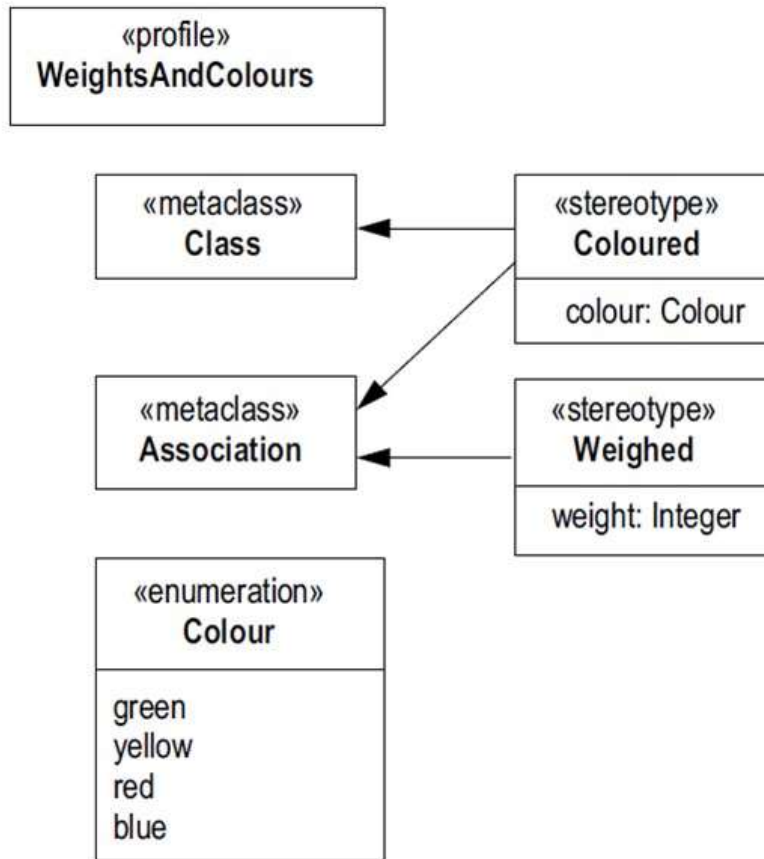


Figure 5 Example of UML profile [20]

Constraints:

Constraints are modeling rules, which get helps from OCL.

Stereotypes:

Stereotypes are profile classes. They define how metaclass should be extended in UML profile. They cannot use stereotypes. They must be used with metaclasses.

Tagged values:

Tagged values are the other part of UML profile. Tagged values have a name and type. They are associated to a specific stereotype. They are attributes of the stereotype classes. For example, in figure 5, stereotype «Coloured» has a tagged value with the name colour and its type is Colour. In the figure 6, the colour value is red.

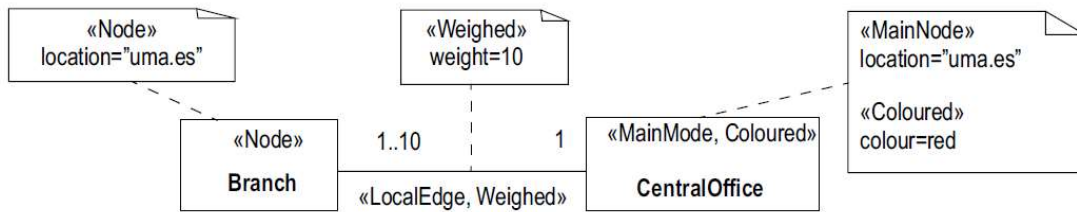


Figure 6 Tagged values in UML profile [20]

2.2.3 Object Constraint Language (OCL)

OCL is a language, which is the complementary form of UML to cover limitations and details of UML. OCL is part of UML. IBM [21] calls it as a business modeling language.

In UML, we cannot reply to some questions. For example, can students who withdraw the course, take the course for the next semester in the university [9]?

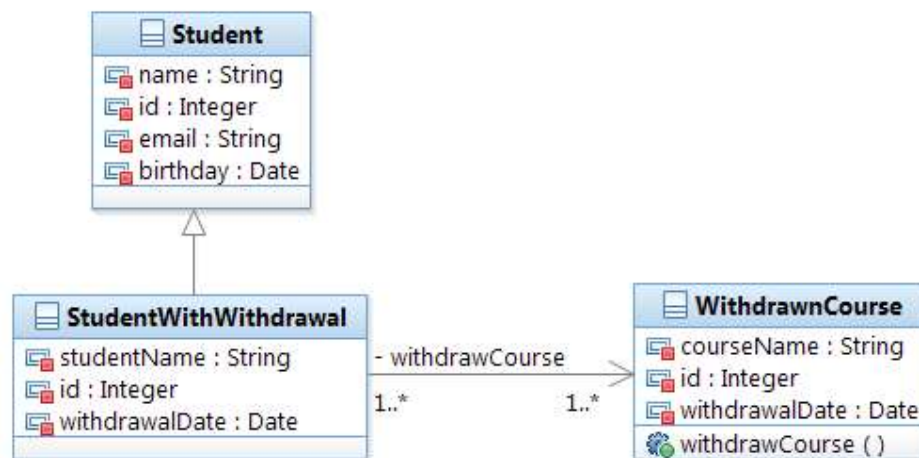


Figure 7 Object Constraint Language (OCL)

OCL has different types. Some of its famous types are bag, set, sequence, and order set. These are inherited from type collection. They are subset of collection.

We use OCL when we cannot reply to some questions with UML. There are some OCL collection operations such as including, excluding, etc. Including operation adds some elements to the collection as including bag {1, 2} adds 1 and 2 to the bag. Excluding operation removes an item from the collection.

There are other operations in OCL such as size, which returns the size of a collection. For example, set {1, 2} results in size: 2. Other operations in OCL are: select, reject and collect.

Select operation is used when there is a selection of items from a collection. For example,

Set {1,2,3,4} -> select($i \bmod 2 = 0$) returns Set {2,4}: Set(Integer) in the result.

Reject operation means to do rejection in a collection. It is the opposite form of select. For example, in this collection: Set {1,2,3,4} -> reject ($i \bmod 2 = 0$) it returns this: Set {1,3}:Set(Integer) in the result.

Collect operation creates a new collection from the existing collection. For example, in:

Set {1,2,4,6,8,10} -> Collect ($i * 2$) it returns Set {2,4,8,12,16,20}:Set(Integer) [9].

2.3 Restricted Use Case Modeling (RUCM)

RUCM stands for restricted use case modeling. It is a type of use case template for specifying use case diagrams. This approach is used in the case study of the thesis for specifying the most important functionalities of smart grid especially AMI part.

The reason for using RUCM rather than other use case specifications (UCSs) is it decreases ambiguity. It also facilitates the automated analysis. There are some restriction rules in the RUCM, which restrict the way users document UCSs. These rules lead to less ambiguity. RUCM is in the form of textual table, which is similar to other UCSs and use case templates in case of fields. However, some fields are different. The common fields are use case name, brief overall description, precondition, post condition, basic flow and alternative flows [22]. The RUCM template has 11 first column fields. These are use case name, brief description of use case, precondition, primary actor, secondary actors, dependency, generalization, basic flow, specific alternative flows, global alternative flows, and bounded alternative flows. The first seven fields are described in the second column. The last four fields, which are basic flow and alternative flows are divided to some different parts such as: RFS (a reference flow step number), steps and post condition.

A basic flow describes the main successful path called “happy path”. It is without conditions and branches. There could be only one basic flow for each use case diagram. Alternative flows are flows with conditions and branches including both success and failure branches. Alternative flows are divided to three parts. These are specific alternative flows, global alternative flows and bounded alternative flows. Specific alternative flow is a specific step in the reference flow. Global alternative flow determines any step in the reference flow. Bounded alternative flow is for having more than one-step in the reference flow.

About restriction rules, there are 26 restriction rules in RUCM. They are grouped into two parts. Rules 1 to 16 are the rules for restricting the use of natural language. They are categorized in the restriction

table with a little description of what each rule means. For example, these rules mention to how to use subject, which time tense to use, use of simple sentences, etc. Remaining rules from rule 17 to 26 are about the restricted use of control structures, except rule 26. This rule is about flows and their post condition. Applying these rules lead to decrease of ambiguity of UCSs. Additionally, it facilitates automated processing [22].

2.4 Cyber Physical systems

CPSs are integration of physical systems with computational devices. Every physical system that is in the network or has Internet connection is CPS. They are embedded systems, which monitors the physical environment [23].

The examples of CPSs are drones, scooters, or modern cars such as Tesla. There are some modeling languages used for physical environment or hardware. These languages are different from the one for the software part. Nowadays, there are devices, which are embedded in computers and they use sensors. Smart phones are typical type of CPSs, which gain popularity among people.

The security of CPSs is important. For example, if modern cars driving program is hacked, then the driver cannot drive it. Therefore, security should be considered. In this thesis, we will consider the security issues of CPSs. Confidentiality, integrity, authenticity, and availability are the most important security issues [8].

The design of CPSs is integration of physical and computational parts. First, they are simulated, then translated to real hardware (physical part) and implemented (computational part). For designing integrated system of CPSs, they have some layers. The layers are application, hardware, which is a physical system, environment, and platform. Each of them is interacting with each other. The development of CPSs is model based. It means the models are used for developing CPSs [24].

The applications for CPSs can be energy, health, medical resources, traffic control, robotics, communication systems, modern transportation systems, sensor networks, water resources, manufacturing, home appliances, electric power like smart grids, etc. In this thesis, we will work with electric power example, which is called smart grid [25] [26].

One application of CPS in health and medical care is assistive help for elderly people. There are devices that help elderly patient in cleaning home, vacuuming home, making lunch or dinner, taking medicine, etc.

CPSs become more popular. The reason is that they are in the area of new research and they have efficiency and effectiveness. The other reason is smart grid, which is type of CPSs decreases the

amount of using fossil fuels energy. Fossil fuels produce CO₂ that is harmful for the environment. Besides harm, it can lead to global warming, which is a scientific research area nowadays. Therefore, using CPSs improves the efficiency and effectiveness. One of the other reasons for popularity of CPSs is related to health. Using cyber physical medical treatment can reduce the chronic disease of aging people in United States [27].

The security, robustness, and safety can be a challenge for CPSs [28]. The security of CPSs is important. In the thesis, we address security related uncertainty of CPSs, especially smart grids. Smart grid generates electricity power to consumers. There is adversary model for security of CPSs [29].

2.5 Smart Grids

Smart grids are one type of CPSs. They are power electricity systems or networks. They generate electricity power and transmit this power to customers such as factories. Smart grids are one of the largest interconnected networks around the world. A failure in one part of smart grid can cause failures to whole network of smart grid. The examples of smart grids can be wind power such as wind turbines, which produce electricity power through turbines. Wind power is one of the renewable energy forms that reduce the production of carbon- dioxide.

Researches have shown that demand for smart grid consumption increases. The reason is that it increases the efficiency of the supply. Consumers tend to use it in an effective manner. The other reason to use smart grids is they use less fossil fuels energies. Fossil fuels energies produce more CO₂ and pollution and makes global warming. Smart grids are renewable energies. They do not produce CO₂, which is harmful for the environment. We see this electricity consumption also in transportation. Some people tend to use electric cars, which use electricity power instead of gasoline. Some buses are hybrid buses, which use electricity power [30].

The benefits of smart grids can be that they can reduce the peak load demand or optimize it. It leads to less generation of electricity power. Other benefit is that smart grids can increase energy efficiency, because they can make customers more involved in the electricity usage [31].

However, besides benefits of using smart grids, there are some disadvantages such as security. Attackers can access to the smart grid network and hack some information of it or they make some damages to the system [31].

The security of smart grids should be considered. If the security is not cared, it can cause damages. Attackers can hack some information. It causes costs and efforts to recover it and it has economic impact.

Reliability can be one of the security challenges of smart grids. The other challenge can be quality of smart grids [32].

The main features of smart grids could be that the smart grid can provide smart meters for the customers. Smart meters can measure the amount of use and price of use. The smart meter provides the security and therefore, the attacker might not access to it.

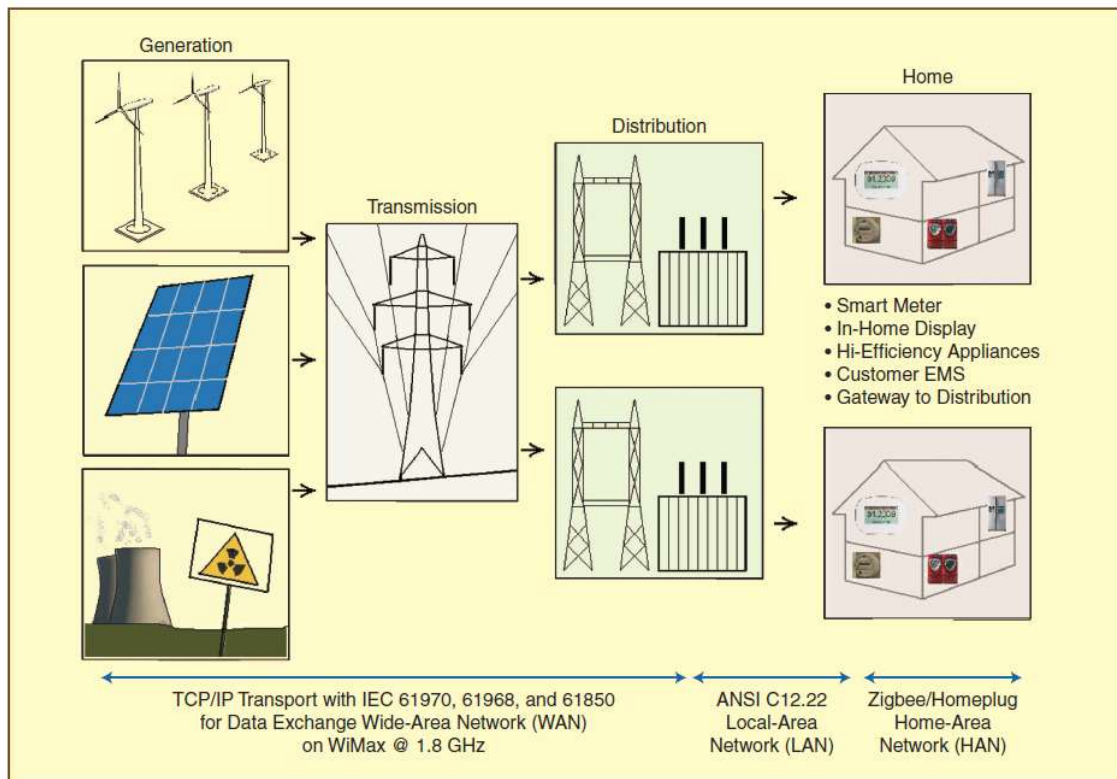


Figure 8 Smart Grid Architecture [33]

2.6 Security Modeling

Security usually concerned with confidentiality, integrity, availability, and accountability. These are security requirements. UML models and UML profiles can be developed and used for specifying and modeling security requirements together with functionalities of a system [34], [35]. Since there can be malicious software that can be harmful to the system we should secure the system. We would leverage some security modeling techniques such as UMLsec to model security for advanced metering infrastructure.

2.7 Security of Cyber-Physical Systems

Security of CPSs is important and it should not come as an after-thought [36]. If it is not considered early while engineering CPSs, it cannot be engineered properly. This can lead to vulnerabilities exploitable by malicious programs and attacks from outside.

We can consider important security requirements such as Confidentiality, Integrity, and Availability. Confidentiality means that the data is confidential and cannot be accessed by unauthorized actor. Integrity means that the data is not changed or modified by unauthorized actor. Availability means that information is available and accessible only by authorized actor.

CPSs can be secured by some methodologies such as encryption, access control, and authentication. Security of CPSs should consider attacks and hacks from outside as well.

The example of CPS is smart grid. Its security issues will be considered in the next section.

2.8 Security and Security Requirements of Smart Grids

The security of smart grids is important. Smart grids are devices, which generate electrical power and transmit the power to consumers like homes, offices, and factories. Smart grid consists of different parts such as Energy transmission infrastructure, energy distribution infrastructure, Data communication network, smart meters, home gateways, network gateways, monitoring modules, smart appliances, Decision making modules, energy generators, energy stores, data stores, and electricity market [37].

Data communication network is the important part of smart grid. In data communication network, different components interact with each other. Interaction of components with each other in smart grid network will introduce security risks. Besides that, smart grid's network transmits data to other places. This can introduce security risks [38]. There are three different smart grid security objectives or requirements. These are confidentiality, integrity, and availability. We have provided their definitions in chapter 1, section 1.3 under the research questions, research question 1 (RQ1).

Availability is the most critical and important security requirement for power system reliability. Electricity should always be available in power grid. Integrity of data is the second important security objective. Confidentiality is the least important security objective [38]. Other security requirements can be data accuracy, and trust [3].

Smart meter is one of the most important parts of smart grid. It generates data related to energy consumption. The data should be confidential. Additionally, customer billing information and forecast information of energy consumption should be confidential. The reason is that they handle sensitive information [37]. Data privacy is another important subject in smart grid. Customers' data like identification, energy consumption, and address should be private and not public.

There are some security threats in smart grid. Examples are physical tampering of meter data and change in smart grid control commands [37]. There are some methods to overcome these threats. These methods are encryption, digital signatures, etc. However, encryption is not always a choice for securing smart grid especially AMI system. Because, there is a possibility that a hacker cracks the security of AMI and sends some copies of remote disconnect to other customers [7]. There are four important security requirements for AMI systems. These are Confidentiality, Integrity, Availability, and Accountability (Non-repudiation).

2.9 Uncertainty

Our daily life is becoming more dependent on cyber physical systems. Since they are complex systems and have unpredictable physical environment, they are designed under uncertainty. Uncertainty means that there is the lack of knowledge about timing and nature of inputs [39]. As reported by [39], uncertainty could happen at different parts of CPSs. It could happen at application level, infrastructure level and integration level. Additionally, there are some types of uncertainty. These types are occurrence, time, content, environment, and geographical location. Depending on the type of uncertainty, different types of measurement can be used. Ambiguity, probability, and vagueness are types of measurement methods for measuring the uncertainty. Human actions and technology can cause uncertainties. There are some elementary uncertainties families. The examples are data delivery uncertainties family, execution environment, storage, governance uncertainties families [39].

There is a CPS uncertainty modeling framework (UMF) proposed in [40]. In this framework, there are some libraries such as risk assessment library and measure library. Additionally, UMF guidelines, some profiles, UML, and other elements are provided in this framework. Additionally, there is another concept called UML uncertainty profile (UUP). It is figured by some meta classes and stereotypes [40].

Smart grids, which are type of cyber physical systems, also face with uncertainty. There are some security-related uncertainties in the AMI. There are some uncertainties in functionalities of AMI, which might cause vulnerabilities to malicious attacks. The other uncertainty can be related to specification, implementation, and evolution of security mechanisms. These can lead to other types of

uncertainties in the functionality of AMI. An example can be incorrect access control, which can disable some physical processes.

3 Related work

In related work, there are some sections about security modeling. Section 3.1 is about UMLsec. Section 3.2 is titled as model at run time security handling, which is divided to two parts. The first part is about topology of smart grid. It is about description of smart grid design, structure, and security modeling of topology. The second part is about models@run.time. It is a reactive security technique for smart grids.

3.1 Unified Modeling Language Security (UMLsec)

By growing society and technology, the need for networked information systems increases. As a result, the attacks against these systems can have impacts on economical or physical aspects of people's lives and organizations. Therefore, to respond these attacks, there is a need to use security models to reduce security risks and increase customer confidence. UMLsec is a type of model-based security model used in securing cyber physical systems [41]. UMLsec is an extension of UML in form of UML profile where security requirements and properties are inserted as stereotypes with tagged values and constraints. Stereotypes' names are written inside brackets: « ». Tagged values are pairs of name-value where the name is referred to the tag. The notation is like this: {tag=value}. Constraints are defined by using mathematical notions.

Threat modeling with mathematical notations can be used in UMLsec. The example is a function called: Threats A(S). In this function, A is adversary type and S is a stereotype. The function returns {delete, read, insert, access}. Read means adversary can read a message. Insert means adversary inserts a message. Delete means adversary deletes a message. Access means adversary accesses to a message [41]. Important security properties such as secrecy and integrity can also be defined mathematically.

It is better to consider about security in design stage rather than development. The reason is that it reduces the cost. The UMLsec is one of the most popular UML based security modeling approaches. The reason for its popularity is UML. Many developers and programmers are trained in UML. Therefore, it is easy to learn and use UMLsec for the security issues. In addition to that, UML is relatively precisely defined [41].

There are other security frameworks for security modeling of cyber physical systems. The examples are SECUREMDD, which is not UML based approach, SECUREUML, etc. [35].

3.2 Model at Run Time Security Handling

3.2.1 Topology of Smart Grid

Smart grid topology is defined for analyzing, simulating, and designing of smart grid infrastructures. There are different parts of smart grid topology such as smart meters, repeaters, which are other smart meters. These repeaters are used to connect the smart meters to concentrators since there might be distance or noise in the way of smart meters to concentrators. Other topology characteristics are concentrators. They control the smart meters. At the top of the topology is central system, which stores the consumption data [42].

In the topology, there might be water meters and gas or heat meters. They are connected to smart meters and smart meters are connected to concentrators. Water meters are used to measure water consumption of customers. Gas or heat meters are used for measuring gas consumption of the customers. They are not directly connected to concentrators.

The topology is in the form of tree, which has subtrees. Smart meters are connected either directly to concentrators or indirectly via repeaters due to noise and distance. Concentrators are the root node of subtrees. The central system is the root of the whole topology tree. Smart meters and the repeaters are the leaves of the topology tree.

There are different kinds of measurement for topology. The examples are the number of smart meters in the topology and the average number of them. Other measurements are the path length from a smart meter to concentrator. It is called the number of hops. The physical distance from smart meters to concentrators is the other example for measurement.

The topology is not the same all the time. It can change, since there are repeaters or other smart meters in the topology. Topology can evolve over time [42]. Picture below shows topology of smart grid in the form of tree with its subtrees and central system as a root node:

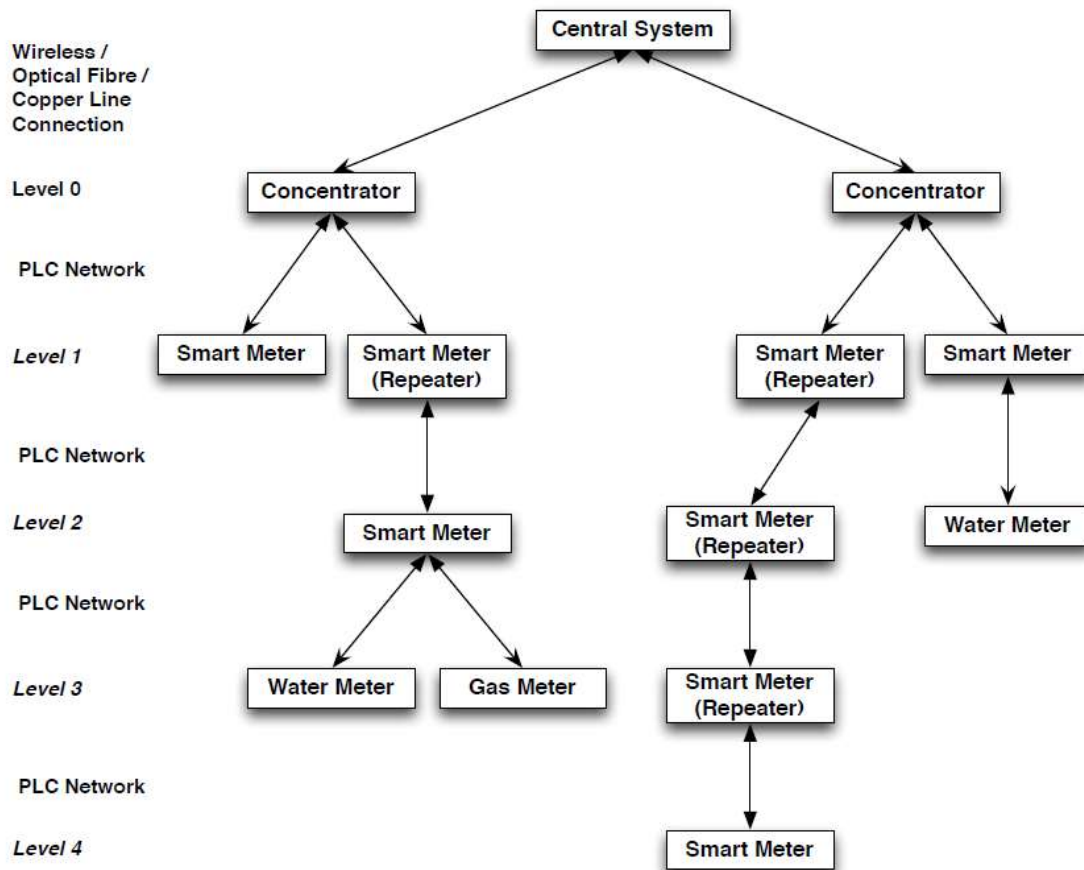


Figure 9 Smart grid topology [42]

In case of security and security modeling for smart grid topology, since the smart grid topology is a communication topology, therefore the security is considered for the communication aspect. The other factor related to security is that the communication topology can be created based on the power line communication (PLC) network. Therefore, where there is a network, the security should also be considered. One drawback of PLC network is emergence of electrical noise and disturbance that might occur within this network. To solve this problem, advanced error detection techniques are used [42].

Another security aspect related to topology of smart grid is communication between water meters and smart grids. This communication is encrypted. Encryption is a mechanism to make the communication more secure. Additionally, the subtrees in the smart grid topology are connected via wireless technology. This means there should be a security modeling in order to model the relationships between different parts of the topology subtree and tree [42].

3.2.2 Models@run.time

The security of smart grids is an important issue. Some actions like disconnecting smart meters in smart grids can affect security of smart grids. By disconnecting, cyber-attacks can increase. In this way, there are proactive and reactive approaches for securing the smart grids. The proactive method is not sufficient method. It cannot handle and delete all the attacks coming from outside. Therefore, the reactive approach is used for handling the security issues. It is used besides the proactive approach. The main reasons for using reactive approach are first, by monitoring and adapting smart grid to failures and attacks, these attacks are dealt with. Second, the global impact of local attacks and failures are minimized by using reactive security techniques [4].

Models@run.time based reasoning engine is reactive approach with corrective security techniques for smart grids. Based on this approach, the protection mechanisms can be used in near real time [4]. To describe how model@run.time is used for security modeling, we mention to reasoning engine. Reasoning engine can simulate and explore potential actions to react with an event [4]. For example, when an intrusion is detected in smart grid, reasoning engine can deactivate communication module to prevent cascading failures to happen. The other tasks that reasoning engine can do are: intrusion detection. Examples are specification-based intrusion detection systems or signature-based intrusion detection systems. Other examples are electrical load, communication network traffic, DOS/DDOS attacks. The reasoning engine and frequency of disturbances and state changes can detect these attacks. To address the security issues of smart grid, the model@run.time and the reasoning engine are running on the top of smart grid topology. When there is a problem in the topology, model@run.time and the reasoning engine can be used for modeling the security issues [4].

One example of security issues the reasoning engine addresses is “malicious shutdown commands”. The reasoning engine handles monitoring and detecting the areas in smart grid, which are remotely shutdown by the malicious attacks. Then, reasoning engine adds these malicious attacks to blacklist to prevent affecting other areas. The greedy algorithm is used for detecting the entities, which are shutdown [4].

4 Methodology

The methodology is the methods, tools, or techniques used for a specific task. In this thesis, we use UML as a main methodology to model CPSs, security and security related uncertainties of CPSs. There are some diagrams in UML. The tool for drawing UML diagrams is IBM Rational Software Architect (IBM RSA). We address research questions outlined in chapter 1, section 1.3 under the research questions with applying UML methodology. There are three subsections in this chapter. In section 4.1, we address research question 1 (RQ1). In section 4.2, we address research question 2 (RQ2). Finally, in section 4.3, we address research question 3 (RQ3).

4.1 Addressing Research Question 1 (RQ1)

We use UML in this thesis to specify, design and model one type of cyber physical systems, which is called smart grids. In the smart grids, the UML methodology and other methodologies are applied for the key part of smart grids, which is AMI. We propose Different types of UML diagrams in the thesis for addressing some of the challenges of AMI system, which are mentioned in the Introduction chapter. For example, one of the challenges in AMI system is cyber-attacks. In these attacks, attackers can send malicious packages to millions of smart meters and cause big problems such as city blackouts. The proposed UML diagrams in this thesis are Use case diagrams with their specifications called Use case templates, Class diagrams, Sequence diagrams and State chart diagrams.

4.2 Addressing Research Question 2 (RQ2)

By using UML as a methodology, we can design basic functionalities of AMI and then security and security-related uncertainties of AMI to handle the security requirements of AMI system. Basic functionalities of AMI are periodic meter reading, remote connect/disconnect of meter, and on-demand meter reading.

We handle the security requirements of AMI system by using some security mechanisms and solutions. These mechanisms or solutions will be modeled by using UML in this thesis to address the security challenges and security requirements of AMI system. OCL is also used in the thesis for specifying the constraints and the security requirements of AMI. We collect, synthesis the functionalities and security requirements of AMI from the papers about smart grid and AMI security. To model security together with functionalities of AMI, we leverage security-modelling techniques such as UMLsec, security patterns that can be fit into the AMI case study. In this thesis, we use authentication, authorization, encryption and decryption security patterns in the design to address security requirements of AMI system.

4.3 Addressing Research Question 3 (RQ3)

We survey the literature about smart grid to collect possible security issues, and then filter and classify them from the uncertainty point of view. Based on that, we will attempt to model at least one of security-related uncertainties of AMI.

5 Case Study

This chapter is divided to many sections and subsections. Section 5.1 gives the overview about the structure of AMI. This section includes four subsections. In these subsections, we provide some general descriptions about AMI head-end, smart meters, Customer Information System (CIS), and core functionalities of AMI head-end, respectively. We discuss about the security design of smart grid in section 5.2. This section is divided to three subsections, which are titled as authentication, authorization, and finally encryption and decryption. In section 5.3, we show some use cases of AMI head-end followed by use cases of smart meter in section 5.4. There are different use cases in these sections. Some of them are related to core functionalities of AMI. Some others are related to security of AMI system. We model these use cases as use case diagrams and we use RUCM approach to specify the use cases. Finally, section 5.5 is about security related uncertainties of smart grid. It includes two subsections, which are some examples of smart grid uncertainties and mutation operators, respectively.

5.1 Structure of Advance Metering Infrastructure

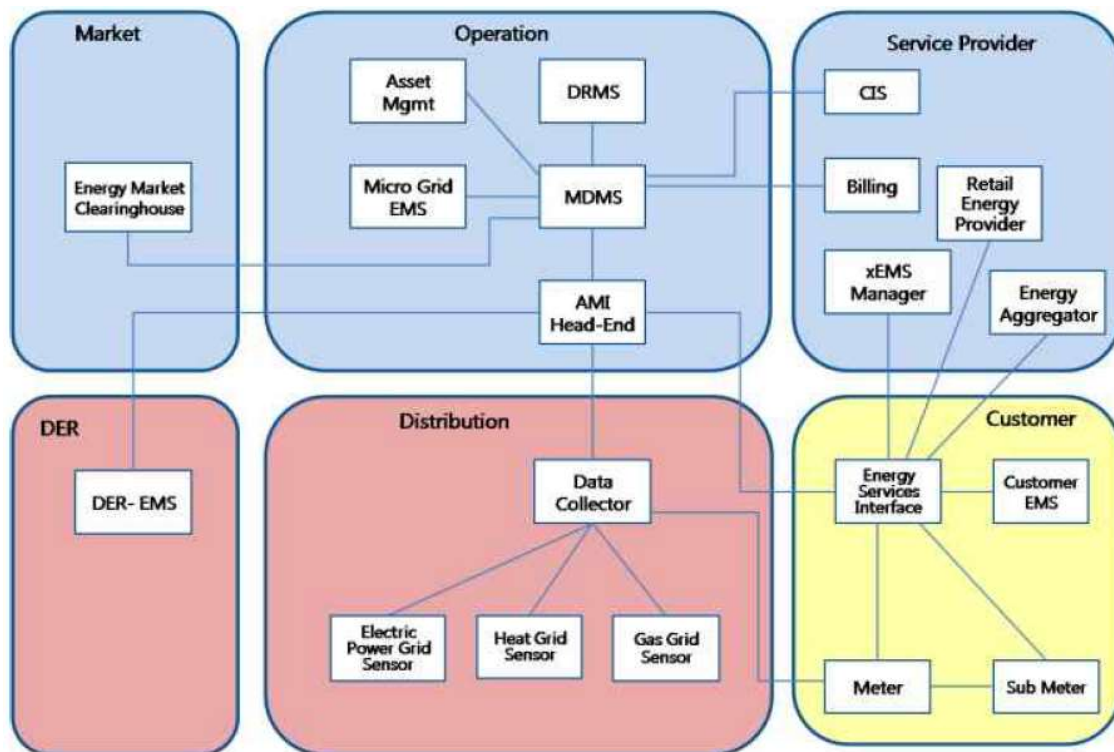


Figure 10 Application-Level Reference Model for Advanced Metering Infrastructure [43]

5.1.1 AMI Head-end

AMI head-end is the back office system that controls and manages the overall advanced metering infrastructure system [44]. AMI head-end is a sub-system of smart grid, which establishes two-way communication with smart meter [45].

AMI head-end is in the server side of smart grid. However, smart meter is in the client side of smart grid. Therefore, AMI head-end and smart meter are located in different sides. That is the reason we separated AMI head-end and smart meter in UML design of this thesis.

The main functionalities of AMI head-end are metering services such as periodic meter reading, on-demand meter reading and remote meter connect/disconnect. We will design these functionalities by using UML in the following sections. We will specify them by use case diagrams, use case specifications, class diagrams, sequence diagrams, and state chart diagrams. We will also add security and security-related uncertainty aspects to the design.

5.1.2 Smart Meters

Smart meters in smart grids are used for measuring the electricity consumption of consumer. They have automatic and bi-directional communication between consumer and the utility. Smart meters can read the electricity consumption both locally and remotely.

The benefits of smart meters are that they inform the consumers how much electricity power to consume and how often and when to use. Other benefit is that smart meters allow the consumers use electricity power in a more efficient and effective ways. The consumers get information about the use of their electricity power through smart meters [32]. By smart meters, users can have more power to control and manage the system. They can get information through smart meters. Then it is possible for them to manage the information.

5.1.3 Customer Information System (CIS)

CIS is defined as Customer Information System. It is used in remote meter connect/disconnect and on-demand meter reading specifications in this thesis. The role of CIS in remote meter connect/disconnect is to send remote meter connect/disconnect request message to AMI head-end. At the end, AMI head-end sends closed/opened internal meter switch verification message to CIS. In on-demand meter reading, CIS sends on-demand meter read request message to AMI head-end. At the final step, AMI head-end sends meter read data to CIS.

CIS is also a part of smart grid, which is located in the server side of smart grid. It is also in different sub-system in comparison with AMI head-end and smart meter. In this thesis, we do not cover CIS in details because the focus of the thesis is more about the communications between AMI head-end and smart meter. Therefore, we occasionally refer to CIS in remote meter connect/disconnect, and on-demand meter reading. It is as a trigger in these cases. We do not mention it as an actor in the use case diagrams and use case specifications.

5.1.4 Core Functionalities of AMI Head-End

Periodic Meter Reading

Periodic meter reading is one of the core functionalities of AMI head-end [44], which we will address in this master thesis by modeling it. In periodic meter reading, smart meter records electricity power consumption periodically in determined intervals. In our case, it is 15 minutes intervals. It means the recording of the electricity consumption is performed every 15 minutes. Then smart meter will collect meter read data, which is electricity power consumption every 4 hours in our case. Smart meter will send the meter read data to AMI head-end. At the end, this data will be sent to customer for billing and payment purposes.

On-Demand Meter Reading

On-demand meter reading is another core functionality of AMI head-end. It is similar to periodic meter reading. The difference is in periodic meter reading, the meter read data or electricity power consumption will be sent to AMI head-end and customer periodically. However, in on-demand meter reading, the meter read data will be obtained based on the demand and request in scheduled date and time. CIS sends on-demand meter read request message to AMI head-end. AMI head-end sends this message to smart meter. Smart meter will retrieve the meter read data and will send it to AMI head-end. AMI head-end will send this information to CIS at the end [44].

Remote Meter Connect/Disconnect

The other functionality of AMI head-end, which we will address in our thesis is remote meter connect/disconnect. In this functionality, smart meter will be connected and disconnected for different reasons. For example, smart meter can be disconnected for the non-payment reason. If the customer tends to not pay the payment for his/her consumption, the smart meter will be disconnected. In remote meter connect/disconnect, CIS sends remote meter connect/disconnect message to AMI head-end. AMI head-end sends this message to smart meter. Smart meter then will act based on the message it receives. If the message is remote meter connect, smart meter will close the meter switch in order to be

connected remotely. Otherwise, if the message is remote meter disconnect, smart meter will open the meter switch to be disconnected remotely. Then in both of the cases, smart meter will send the verification or acknowledgment message to AMI head-end to show that it is connected or disconnected. AMI head-end will send this verification message to CIS at the end [44].

5.2 Security Design of Smart Grid

In addition to modeling basic functionalities of smart grid in the thesis, we model security aspects of smart grid and we cover security requirements of AMI system. There are some security requirements for AMI system. These requirements are Confidentiality, Integrity, Availability and Accountability (Non-repudiation). Confidentiality means the information should be accessible and disclosed only to authorized entities. The lack of confidentiality means there is an unauthorized disclosure of information. Integrity points that the information should be protected against modification or destruction. The lack of integrity means the unauthorized modification or destruction of data. Availability indicates that the information is available and there is a reliable access to information. The lack of availability means that the access to information is disrupted [3]. Accountability implies that the receiver of the information should not deny the receipt of the information. If the receiver denies the receipt of the information, it indicates the lack of accountability. In order to cover these security requirements, we propose some security mechanisms in the thesis for designing smart grid system. These mechanisms are described in the following sub-sections. They are Authentication, Authorization, Encryption, and Decryption security mechanisms. We use UML diagrams as a methodology for modeling the security aspects.

5.2.1 Authentication

Authentication is a security mechanism or security solution for handling security requirements. As mentioned before, there are three important security requirements, which are Confidentiality, Integrity, and Availability. We use authentication security mechanism in our thesis. Authentication means the parties who are involved in the communications should be the same parties as they were expected to be, not the malicious attacker or any unknown third party [46]. In the smart grid, there is a two-way communication between AMI head-end and smart meter. Smart meter should be authenticated by AMI head-end. In this thesis, the authentication mechanism to authenticate smart meter by AMI head-end is based on smart meter's id and password, which is called smart meter's credentials. It means in order to authenticate smart meter by head-end, smart meter needs to send its id and its password to AMI head-end. By sending these credentials, AMI head-end will know which smart meter is requesting to be authenticated by head-end. Then, based on smart meter's credentials, AMI head-end will verify smart meter and will authenticate smart meter if the credentials are the same as they were expected to be. We

elaborate authentication in the following UML diagrams including Use case, class, sequence, and state chart diagrams.

In use case diagrams, we have a use case diagram and use case specification for authentication. This use case is re-used in some of the other use case diagrams.

In class diagrams, in AMI head-end class diagram, there are some classes for authentication security design. We use security pattern for designing smart meter authentication in AMI head-end class diagram [47]. In authentication security pattern a subject is the one to be authenticated. In AMI head-end class diagram, the subject to be authenticated is smart meter. Since smart meter is not part of AMI head-end class diagram, therefore the credentials of smart meter is the subject requests to be authenticated. Authenticator is the entity to authenticate the subject. In this case, smart meter controller authenticates smart meter's credentials. Authenticator will authenticate the subject based on some authentication information. Here, the authentication information is credentials of smart meter. If the authentication process is successful, authenticator creates some proof of identity. Here, the proof of identity is session. Because after smart meter is authenticated, the session will be created. In order to create session, smart meter controller calls the session manager to create the session.

In sequence diagrams, there is authentication sequence diagram for showing more details about how the authentication process is performed. In some of the other sequence diagrams, we re-use this authentication sequence diagram. Finally, in state chart diagrams, there are some states related to authentication process.

5.2.2 Authorization

Authorization is another security mechanism to handle security requirements. Authorization means what a user or subject can do. For example in smart grids, in AMI head-end side, smart meter needs to be authorized. It means when smart meter sends a package to AMI head-end, AMI head-end must check the package code of smart meter to see if the package code is in the list of allowed packages to be sent by smart meter. For example, smart meter can send packages such as meter readings, internal meter switch verification messages in remote meter connect/disconnect or some other type of packages. Smart meter is not allowed to send malicious packages. The reason for designing authorization and other security mechanisms is to handle and address security requirements of AMI head-end system. These requirements have already been discussed in pervious sections.

The approach for authorization design in AMI head-end class diagram is based on Role Based Access Control (RBAC). In RBAC, there are some users (subjects) that are assigned to roles. In other words, a role in RBAC should belong to group of actors. For example, in AMI system, group of smart meters

form a role. These group of smart meters can have different roles. Some of them can be grouped as normal meters or have a role of normal meters. Some others can have the role of suspicious smart meters and others can be industrial smart meters. These roles are defined in RBAC to specify what access rights belong to which roles. Therefore, in RBAC method, the access rights are given to the roles according to their functions. For example, in AMI system, the access rights of normal smart meters can be different with access rights of industrial smart meters.

One way to implement RBAC is to use PDP-PEP structure. PDP is policy decision point and PEP is policy enforcement point. PDP receives authorization requests sent from PEP. The, PDP returns authorization decision to PEP. We assume that the policy is predefined in the XML file or in a data base. In this database, which policy is defined, we can specify the subject, roles, access rights and the resources that can be accessed. It means that this subject belongs to this role and this role can have these access rights to this resource. The access rights of subjects or roles are already defined in the PDP, which are stored in the database or XML file. Therefore, when PDP needs to check these rights, it can query the database to get the access rights of that subject or that role. For example, in AMI system, in AMI head-end side, when smart meter needs to be authorized by AMI head-end, smart meter has some access rights that shows what actions that specific smart meter can do or what packages that smart meter can send to AMI head-end. These are defined as access rights of smart meter in the data base or XML file. When policy decision point needs to authorize smart meter, it will check the access rights of smart meter by querying the data base or XML file.

In this thesis, we applied authorization in all types of UML diagrams. For example, in use case diagrams, we have a use case called “Authorize” for specifying authorization process.

In class diagrams, we use security patterns for designing authorization in AMI head-end class diagram [47]. In AMI head-end class diagram, the subject to be authorized is smart meter. We use RBAC approach. The PDP in this class diagram is session object. The PEP is smart meter controller. Smart meter controller sends authorization request to session. Session will check the access rights of smart meter. Then it will return authorization decision to smart meter controller. This decision can be either the smart meter is authorized or not authorized. As we assumed, the access rights of smart meter are predefined in the policy, in database or XML file. Whenever session needs to check access rights of smart meter, it will query the database to get the access rights. In AMI head-end class diagram, since smart meter has already been authenticated by AMI head-end, authentication is a prerequisite for authorization. In other words, authorization depends on authentication and is done after authentication. Because in smart meter authentication, the goal is to make sure that the smart meter, which sends package to AMI head-end is the one that was supposed to send the package. In authorization, the goal is to check and make sure the authenticated smart meter is sending packages, which are in the list of

access rights of smart meter. The approach for designing authorization in smart meter class diagram is different. In this thesis, we do not focus about authorization in the smart meter side.

In sequence diagrams, we elaborate the detailed steps of authorization process in Authorization sequence diagram from the AMI head-end side. This sequence diagrams is re-used in some of the other sequence diagrams by using ref structure. Finally, in state chart diagrams of AMI head-end side, there are some steps related to authorization.

5.2.3 Encryption and Decryption

Encryption and decryption are the other security mechanisms for handling the security requirements. They are cryptographic methods to secure communications between entities in a system [48]. In smart grid, this communication is between AMI head-end and smart meter. In encryption, when one party wants to send a sensitive message or data to another party, the sender encrypts the package before sending. In decryption, when one party receives a package from the other party, it decrypts the encrypted package to read the content of the package. There are some cryptographic algorithms for encrypting and decrypting the package. Cryptographic algorithms can be symmetric meaning there is only one secret key used for both encryption and decryption. These algorithms can also be asymmetric meaning there are two keys public and private keys for encrypting and decrypting the package, respectively [49].

There are some assumptions about the key exchange process that it could be done already before the system is deployed. The selection of cryptographic algorithm depends on different factors such as data criticality or sensitivity, value, etc. [3]. For example, in smart grid some information is very sensitive such as remote meter connect/disconnect commands that their security is very important. In this case, it is better to use asymmetric cryptography. Because the keys used in this scheme is much longer than the keys used in symmetric algorithms. Therefore, long keys provide better security in comparison with short keys used in symmetric approach [50]. We use asymmetric method for encrypting/decrypting sensitive information or commands such as remote meter connect/disconnect command in our thesis for AMI system.

On the other hand, in smart grid or AMI system, some information has a big size. For example, in periodic meter reading, the size of meter read data can be large. Therefore, in this case, using asymmetric method is not very useful. Because using this method for large amounts of data requires a lot of time and resource to do encryption. Symmetric cryptography is much faster than asymmetric cryptography. The advantage of symmetric method is not consuming too much computing power [50]. Therefore, asymmetric method needs more computing power especially if the data is big, it is not a useful method for encryption and decryption processes. For these kind of big data, the encryption and

decryption mechanisms need to be more efficient in term of providing better security. One cryptographic solution to address security requirements of large information is to use hybrid encryption and decryption scheme. In this approach, both symmetric and asymmetric cryptography algorithms are used. Asymmetric key is used to encrypt the symmetric key and symmetric key is used to encrypt the data. We use hybrid approach in our thesis as encryption/decryption mechanism for large amount of information such as periodic meter read data in AMI system.

We model all kinds of UML diagrams with using encryption and decryption mechanisms for designing security aspects of smart grid. In use case diagrams, we have some use cases for encryption and decryption. These use cases can be re-used in some of the other use case diagrams.

In class diagrams, we use security pattern. In this pattern, there is application component class, which encrypts or decrypts the package. There is data class, which is the class to be encrypted or decrypted [47]. There are also other classes used for transmission of packages.

In sequence diagrams, we have some sequence diagrams for encryption and decryption. These sequence diagrams can be re-used in some of the other sequence diagrams. Finally, in state chart diagrams, there are some states related to encryption and decryption, which can be re-used in other state chart diagrams.

5.3 Use cases of AMI Head-End

5.3.1 Table of Actors for AMI Head-End Use Cases

Table 1 Table of Actors for AMI head-End Use Cases

Name	Description	Actor/Subsystem
AMI Head-End	AMI head-end is the back office in the smart grid. It controls the advanced metering infrastructure.	Subsystem
Smart Meter Controller	Smart meter controller is part of AMI head-end. Since there are many smart meters in the system, smart meter controller is the one to handle and manage all the smart meters. It deals with smart meters.	Actor
Session Manager	Session manager is used to manage sessions. It performs tasks such as create session and look up session.	Actor
Session	Session is used in authorization process. It acts as a policy decision point. It has the access rights of smart meter. When smart meter controller sends authorization request to session, it checks the access rights of smart meter and returns the authorization decision to smart meter controller.	Actor
Smart Meter	Smart meter is used for measuring the electricity consumption of consumer. Smart meter communicates with AMI head-end.	Actor

5.3.2 AMI Head-End's Initialization Use Case Diagram

Summary

This is the first use case diagram from the AMI head-end side. There is only one use case and one actor in this use case diagram. The use case is named initialization and the actor is AMI head-end. This use case is used as a pre-condition for other following use cases. In this use case, AMI head-end will be initialized. It means there are some steps for making AMI head-end ready to listen to connection requests from smart meter.

Description of Use Cases

As mentioned, there is only one use case in this use case diagram, which is named initialization use case. The actor for performing this use case is AMI head-end. In order for AMI head-end to be initialized, there are some small steps inside the initialization use case. First AMI head-end needs to

create a server socket. The role of server socket is to listen to the network and waiting for receiving packages or requests from smart meter. Since server socket is a part of AMI head-end, we do not need to show it in the use case diagram. A server socket is created with the configured port number. In creation of server socket, there are two conditions. The server socket can be created successfully or the creation of server socket can be failed. If it is created successfully, it means the server socket will wait for receiving connection requests from smart meter. Then it makes conditions ready for establishing two-way connection between AMI head-end and smart meter. Otherwise, if the creation of server socket fails, then AMI head-end will show an error message. Then it should try again to create the server socket.

The Use Case Diagram

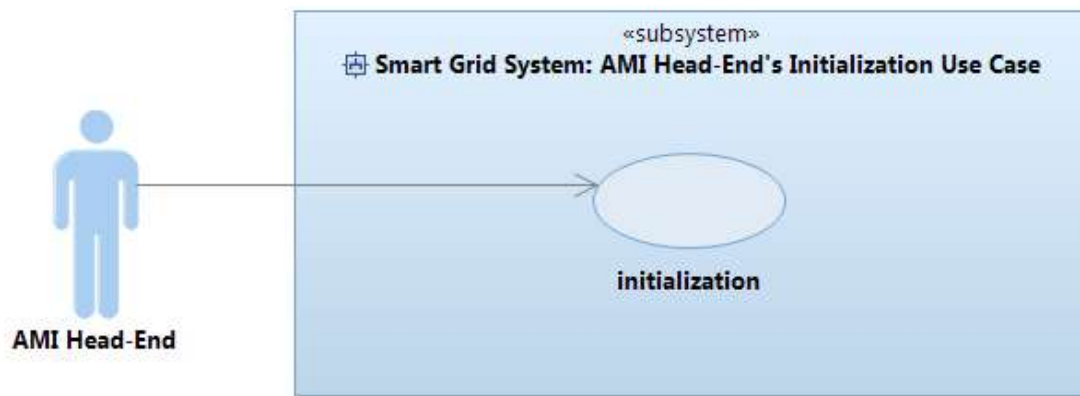


Figure 11 AMI Head-End's Initialization Use Case Diagram

Use Case

Table 2 AMI Head-end's Initialization Use Case

AMI Head-End's Initialization	
Use Case ID	UC Initialization
Use Case Name	Initialization
Description	This use case shows how AMI Head-end is initialized. For example, this use case should contain all the steps to make AMI head-end ready to listen to connection requests from smart meters.
Precondition	AMI head-end is configured.
Primary Actor	AMI head-end
Secondary Actors	None
Dependencies	None
Basic Flow	Steps:
	1 AMI head-end creates a server socket with the configured port number.

	2	IF server socket is created successfully THEN DO
	3	The server socket is waiting for connection from smart meters.
	4	A smart meter connects to the sever socket, which returns a client socket to that smart meter.
	5	AMI head-end creates a concurrent process to handle the establish connection (UC Establish Two-way Connection) to that smart meter MEANWHILE AMI head-end continues to wait for connection from smart meters. UNTIL server socket is closed.
	Post Condition	AMI head-end continues waiting for connection requests from smart meter.
Specific Alternative Flow	RFS Basic Flow 2	
	Steps:	
	1	ELSEIF server socket is not created successfully because the port is in use THEN
	2	AMI head-end shows an error message.
	3	AMI head-end is reconfigured with an unused port number.
	4	RESUME Step 1
	5	ENDIF
	Post Condition	AMI head-end is reconfigured.

5.3.3 AMI Head-End Establishes Connection with Smart Meter Use Case

Summary

There are three use cases in this use case diagram. The main use case in this diagram is “Establish Two-way Connection”. The other two use cases “Receiving Package from Smart Meter” and “Sending Package to Smart Meter” are other use cases, which are included from the main use case. The actors are AMI head-end and smart meter.

Description of Use Cases

After AMI head-end is initialized, the next step is to establish a connection between AMI head-end and smart meter. This is a two-way connection between AMI head-end and smart meter. There are also some steps in the process of establishing connection. First, AMI head-end accepts the connection request sent by smart meter. Then AMI head-end will send the connected acknowledgment message to smart meter to show that AMI head-end has accepted the connection request sent by smart meter.

The Use Case Diagram

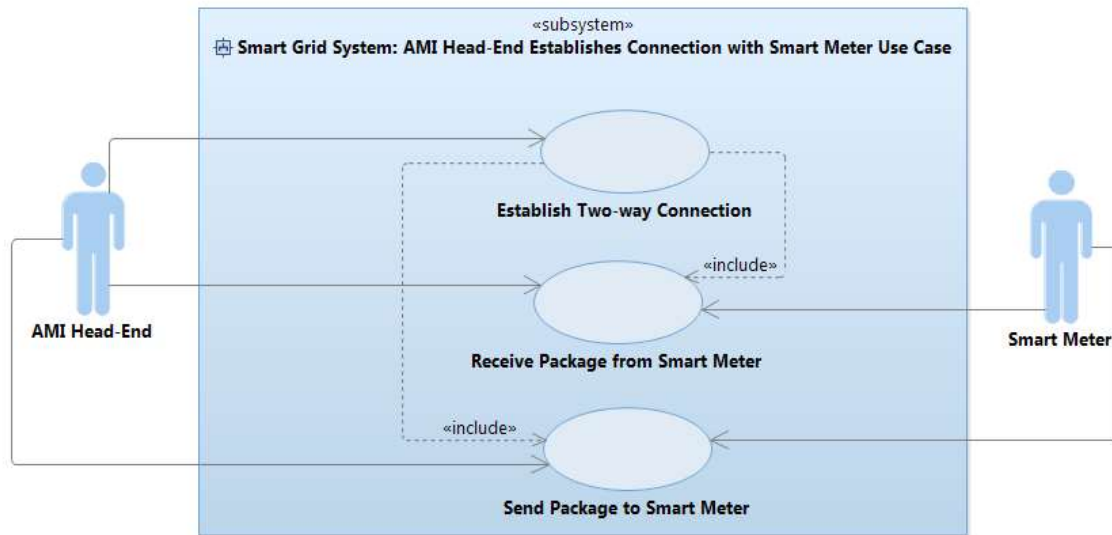


Figure 12 AMI Head-End Establishes Connection with Smart Meter Use Case Diagram

Use Case

Table 3 AMI Head-End Establishes Connection with Smart Meter Use Case

AMI Head-End Establishes Connection with Smart Meter	
Use Case ID	UC EstablishTwo-wayConnection
Use Case Name	Establish Two-way Connection
Description	This is the first step in registration process of smart meter. In this step AMI head-end establishes connection with smart meter.
Precondition	A client socket for the connection to the requesting meter has been created.
Primary Actor	AMI head-end
Secondary Actors	Smart Meter
Dependencies	INCLUDE USE CASE Receiving Package from Smart Meter INCLUDE USE CASE Sending Package to Smart Meter
Basic Flow	<p>Steps:</p> <ol style="list-style-type: none"> 1 A new concurrent process uses the client socket (i.e., the connection to the smart meter) for constantly waiting to receive packages sent from smart meter (UC_ReceivingPackageFromMeter) MEANWHILE the client socket can also be used for sending packages from AMI head-end to smart meter. 2 The client socket is used for sending a package with connected acknowledgment message to smart meter (UC_SendingPackageToMeter). <p>Post Condition</p> <p>There is a two-way connection between AMI head-end and smart meter, i.e., head-end side is ready to receive packages from smart meter as well as to send packages from head-end to smart meter.</p>

5.3.4 Receiving Package from Smart Meter Use Case

Summary

This use case is a general use case for receiving all kinds of packages from smart meter. The main use case in this use case diagram is “Receive Package from Smart Meter”. This use case includes “Decrypt Package”, “Response to Smart Meter” and “Send Package to Smart Meter” use cases. The actors are smart meter controller and smart meter.

Description of Use cases

AMI head-end waits for receiving package from smart meter. Then smart meter controller, which is part of AMI head-end will receive a package from smart meter. After that, smart meter controller will verify the received package from smart meter and will decrypt the received package. If the package is from expected smart meter, then it will process the package. Otherwise, it will send an error message to smart meter.

Use Case Diagram

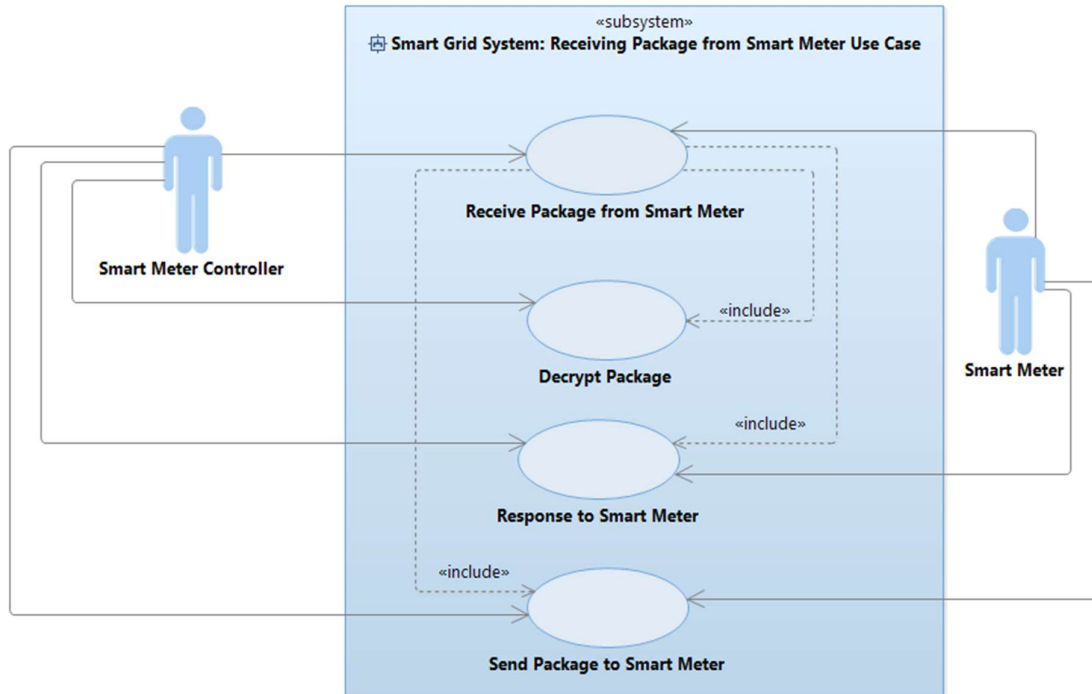


Figure 13 Receiving Package from Smart Meter Use Case Diagram

Use Case

Table 4 Receiving Package from Smart Meter Use Case

Receiving package from smart meter	
Use Case ID	UC_ ReceivingPackageFromMeter
Use Case Name	Receive Package from Smart Meter
Description	This is a general use case for specifying how AMI head-end receives packages from smart meter.
Precondition	A client socket for the connection to the requesting meter has been created.
Primary Actor	Smart meter controller
Secondary Actors	Smart Meter
Dependencies	INCLUDE USE CASE Decrypt Package INCLUDE USE CASE Response to Smart Meter INCLUDE USE CASE Send Package to Smart Meter
Basic Flow	Steps:
	1 DO
	2 AMI head-end is waiting to receive package from the smart meter.
	3 Smart meter controller receives a package from the smart meter.
	4 IF smart meter controller, which is part of AMI head-end verifies the digital signature and decrypts the received package from the smart meter (UC_DecryptPackage) THEN
	5 Smart meter controller creates a new concurrent process (thread) to process the decrypted package (UC_ResponseToMeter) MEANWHILE AMI head-end continues waiting for receiving packages.
	6 UNTIL the client socket is closed.
Post Condition	The package is received from smart meter to head-end.
Specific Alternative Flow	RFS Basic Flow Step 3
	Steps:
	1 ELSEIF smart meter controller does not verify the digital signature or decrypt the received package from the smart meter THEN
	2 Smart meter controller creates a new concurrent process (thread) to send a package with error message/code to the smart meter (see UC_SendPackageToMeter).
	3 ABORT
	4 ENDIF
Post Condition	A package with error message/code is sent to the meter.

5.3.5 Sending Package to Smart Meter Use Case

Summary

This is a use case for sending package from smart meter to AMI head-end. There are two use cases in this use case diagram. First use case is “Send Package to Smart Meter”. The other use case is “Encrypt Package”. This use case is included from the first use case. The actors are smart meter controller and smart meter.

Description of Use Cases

AMI head-end wants to send a package to smart meter. Smart meter controller creates a data package and sends a package to smart meter. Before sending a package to smart meter, smart meter controller first should encrypt and sign the package for the security purposes. Then smart meter controller will send the package to smart meter.

The Use case Diagram

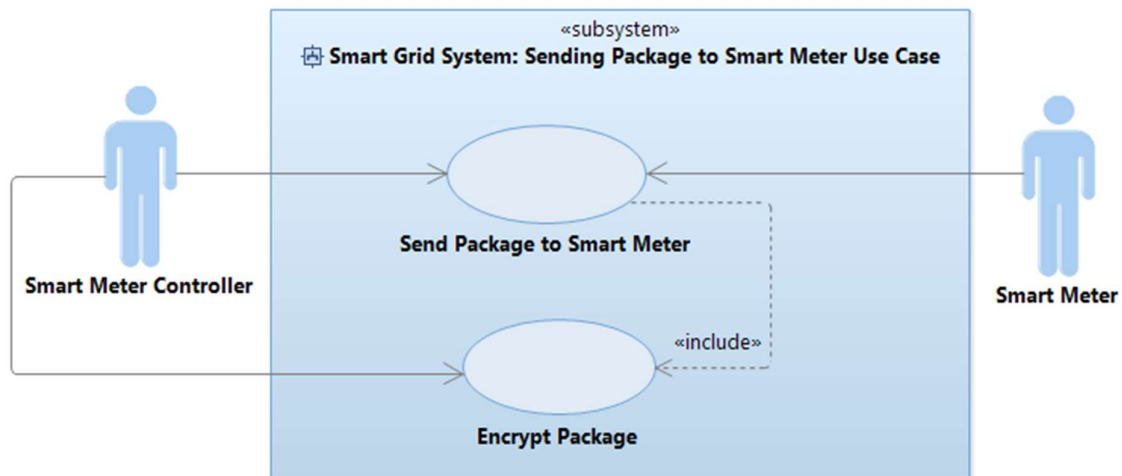


Figure 14 Sending Package to Smart Meter Use Case Diagram

Use Case

Table 5 Sending Package to Smart Meter Use Case

Sending package to smart meter	
Use Case ID	UC_ SendingPackageToMeter
Use Case Name	Send Package to Smart Meter
Description	This is a general use case for specifying how AMI head-end sends packages to smart meter.
Precondition	A client socket for the connection to the requesting meter has been created.
Primary Actor	Smart meter controller
Secondary Actors	Smart meter
Dependencies	INCLUDE USE CASE Encrypt Package
Basic Flow	Steps:
	1 Smart meter controller creates a package for sending to the smart meter.
	2 IF smart meter controller encrypts and signs the package (UC_EncryptPackage) THEN
	3 Smart meter controller sends the package to the smart meter via client socket.
	Post Condition The package is sent to the smart meter.
Specific Alternative Flow	RFS Basic Flow 2
	Steps:
	1 ELSEIF smart meter controller does not sign the package THEN
	2 Smart meter controller creates a new concurrent process (thread) to send a package with error message/code to the smart meter (see UC_SendPackageToMeter).
	3 ABORT
	4 ENDIF
	Post Condition A package with error message/code is sent to the meter.

5.3.6 Decrypting Package Use Case

Summary

The use case in this use case diagram is called “Decrypt Package”. The actor to perform this use case is smart meter controller. In this use case, when AMI head-end receives a package from smart meter, it should first decrypt the package. There are different types of cryptographic algorithms for encrypting and decrypting the package. For example, there are symmetric, asymmetric and hybrid approaches. In symmetric cryptography, there is only one secret key for both encrypting and decrypting the packages. In Asymmetric approach, there are two keys, one public key and one private key for encrypting and decrypting the package, respectively. Hybrid cryptography is the combination of both symmetric and asymmetric approaches. In our thesis, depend on the criticality and sensitivity of data we use the

appropriate method. For example in case of sensitive information such as remote meter connect/disconnect commands, we use asymmetric cryptographic solution. Because, it uses the long keys, which provide better security. However, in case of large data, such as periodic meter read data, we use hybrid approach. Because asymmetric method is not useful and time consuming due to requirement for larger keys to encrypt and decrypt large data. We assume that the key exchange process has been done already before system is deployed.

In the process of package decryption from the AMI head-end side, before decrypting the received package, AMI head-end should first verify the digital signature of received package. Digital signature are used to prove that the specific person has sent the specific message. The digital signature solution is based on encrypting the hash of the message by using private key and adding this signature to the message itself [49]. In this use case diagram, the sender of the message is smart meter and the receiver is AMI head-end. If AMI head-end has the public key of smart meter, smart meter controller, which is part of AMI head-end can check and verify if the digital signature of the received package matches with the hash of the message. If it is verified, then smart meter controller can decrypt the received package. If not, then it means the message is modified or the signature is created with the different key [49]. One reason that smart meter controller cannot verify the digital signature of received package might be due to uncertainty in the system. Some attackers can introduce some uncertainties in the AMI system. We discussed about this issue in section for examples of smart grid uncertainties in the thesis.

Description of Use Cases

In “Decrypt Package” use case when AMI head-end receives a package, first smart meter controller should verify the digital signature of received package sent by smart meter. Then smart meter controller decrypts the package to read the content of the package.

The Use Case Diagram

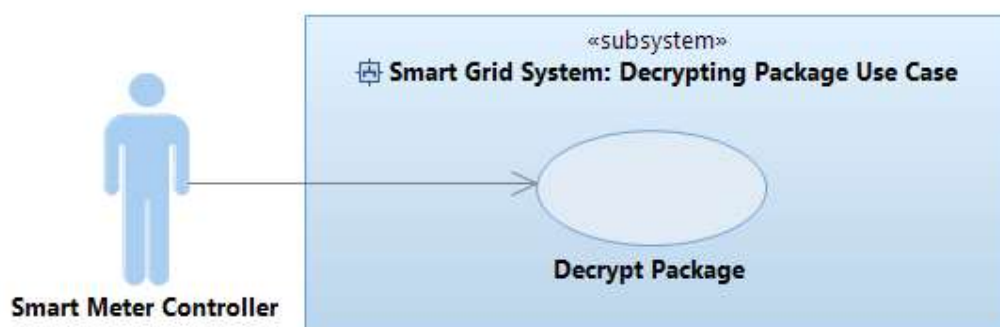


Figure 15 Decrypting Package Use Case Diagram

Use Case

Table 6 Decrypt Package Use Case

Decrypting Package									
Use Case ID	UC_DecryptPackage								
Use Case Name	Decrypt Package								
Description	This is the use case used for the security purposes. When AMI head-end receives a package from smart meter, it decrypts the package to read the content of the package.								
Precondition	AMI head-end has received the package from smart meter.								
Primary Actor	Smart meter controller								
Secondary Actors	None								
Dependencies	None								
Basic Flow	<p>Steps:</p> <table border="1"> <tr> <td>1</td> <td>IF Smart meter controller verifies the digital signature of received package by having the public key of smart meter THEN</td> </tr> <tr> <td>2</td> <td>Smart meter controller decrypts the received package.</td> </tr> </table> <p>Post Condition The received package has been decrypted.</p>	1	IF Smart meter controller verifies the digital signature of received package by having the public key of smart meter THEN	2	Smart meter controller decrypts the received package.				
1	IF Smart meter controller verifies the digital signature of received package by having the public key of smart meter THEN								
2	Smart meter controller decrypts the received package.								
Specific Alternative Flow	<p>RFS Basic Flow 1</p> <p>Steps:</p> <table border="1"> <tr> <td>1</td> <td>ELSEIF smart meter controller does not verify the digital signature of received package THEN</td> </tr> <tr> <td>2</td> <td>Smart meter controller sends an error message to smart meter.</td> </tr> <tr> <td>3</td> <td>ABORT</td> </tr> <tr> <td>4</td> <td>ENDIF</td> </tr> </table> <p>Post Condition A package with error message/code is sent to meter.</p>	1	ELSEIF smart meter controller does not verify the digital signature of received package THEN	2	Smart meter controller sends an error message to smart meter.	3	ABORT	4	ENDIF
1	ELSEIF smart meter controller does not verify the digital signature of received package THEN								
2	Smart meter controller sends an error message to smart meter.								
3	ABORT								
4	ENDIF								

5.3.7 Response to Smart Meter Use Case

Response to smart meter use case is a general use case to show how AMI head-end will respond to smart meter after it received the package from smart meter. Depending the package code, AMI head-end will respond to smart meter differently. There are different use cases in this use case diagram. The main use case is “Response to Smart Meter”. Other use cases are included from this use case.

Description of Use Cases

The main use case in this use case diagram is “Response to Smart Meter”. AMI head-end after receiving the packages will process the packages depending on the package code and will respond to smart meter. For example, if the package code is for the authentication of smart meter, then AMI head-end will authenticate the smart meter, otherwise it will act differently. For example, if the package code is periodic meter reading from smart meter, then AMI head-end will process the meter read data.

The Use Case Diagram

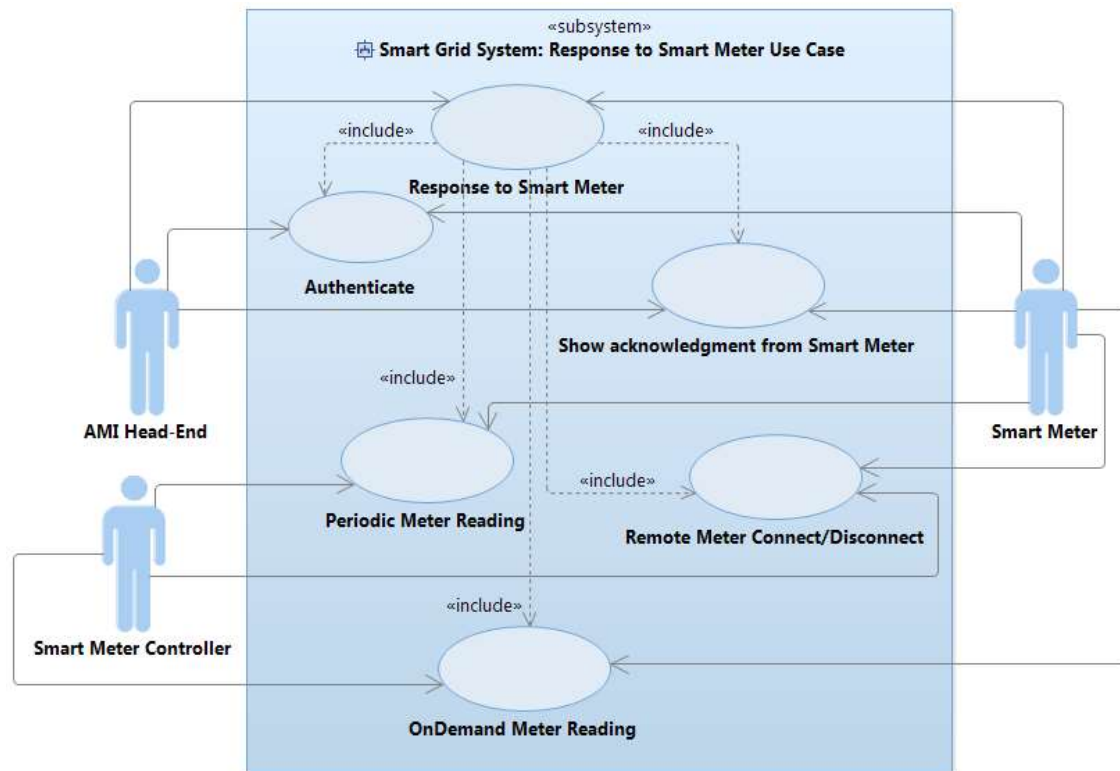


Figure 16 Response to Smart Meter Use Case Diagram

Use Case

Table 7 Response to Smart Meter Use Case

Response to smart meter	
Use Case ID	UC_ResponseToMeter
Use Case Name	Response to Smart Meter
Description	This is a general use case for specifying how head-end side processes a received package from smart meter.
Precondition	A package from meter has been received and decrypted successfully.
Primary Actor	Smart meter controller
Secondary Actors	AMI head-end, Smart Meter
Dependencies	INCLUDE USE CASE Authenticate INCLUDE USE CASE Show Acknowledgment from Smart Meter INCLUDE USE CASE Periodic Meter reading INCLUDE USE CASE Remote Meter Connect/Disconnect INCLUDE USE CASE On-Demand Meter Reading
Basic Flow	Steps:
	1 Smart meter controller reads the decrypted package to check the package code.

2	IF the package code is for the authentication of smart meter THEN
3	AMI head-end authenticates the smart meter. (UC_Authentication).
4	ELSEIF the package code is the ACK message of meter that it is configured successfully THEN
5	AMI head-end shows the message. (UC_ShowACKfromMeter)
6	ELSEIF the package code is the Periodic Meter Readings from meter THEN
7	Smart meter controller processes the Meter Readings data. (UC_PeriodicMeterReading)
8	ELSEIF the package code is internal meter switch verification message from meter THEN
9	Smart meter controller processes the internal meter switch verification message.(UC_RemoteMeterConnect/Disconnect)
10	ELSEIF the package code is on-demand meter read data from smart meter THEN
11	Smart meter controller processes the on-demand meter read data. (UC_On-DemandMeterReading)
12	ELSEIF the package code is unknown to the headend side THEN
13	AMI head-end sends to smart meter a package with the error message that the package code is unknown (UC_SendPacakgeToMeter).
14	MMB shows the error message.
15	ENDIF
Post Condition	The package from meter has been processed, i.e., a response from headend side is sent back to smart meter.

5.3.8 Encrypting Package Use case

Summary

The use case in this use case diagram is called “Encrypt Package”. The actor to perform this use case is smart meter controller.

Description of Use Cases

In the process of encryption package, the sender first should encrypt the package before sending the package to another party. Here, AMI head-end wants to send a package to smart meter. Therefore, smart meter controller encrypts and signs the package before sending it to smart meter by using cryptographic algorithms. Then smart meter controller sends the encrypted package to smart meter.

The Use Case Diagram

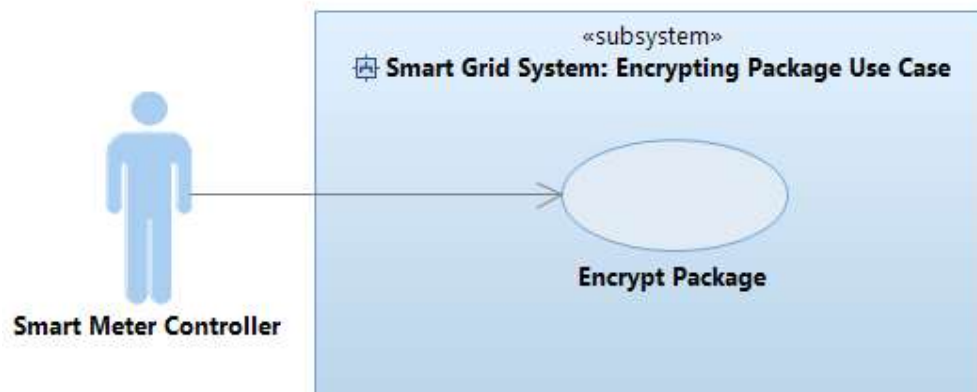


Figure 17 Encrypting Package Use Case Diagram

Use Case

Table 8 Encrypting Package Use Case

Encrypting Package	
Use Case ID	UC_EncryptPackage
Use Case Name	Encrypt Package
Description	This is a use case used for the security purposes. When AMI head-end wants to send a package to smart meter, it first encrypts and signs the package before sending. Then, it sends the package to smart meter.
Precondition	AMI head-end has created a package for sending to the smart meter.
Primary Actor	Smart meter controller
Secondary Actors	None
Dependencies	None
Basic Flow	Steps:
	1 Smart meter controller encrypts the package by using cryptographic algorithms and the key for encryption.
	2 IF Smart meter controller signs the package using digital signatures THEN
	3 Smart meter controller sends the encrypted and signed package to smart meter.
Post Condition	The package for sending has been encrypted.
Specific Alternative Flow	RFS Basic Flow 2
	Steps:
	1 ELSEIF smart meter controller does not sign the package THEN
	2 Smart meter controller sends the error message to smart meter.
	3 ABORT
	4 ENDIF
Post Condition	A package with error message/code is sent to meter.

5.3.9 Showing Acknowledgment from Smart Meter

Summary

There are two actors in this use case diagram. AMI head-end acts as a primary actor, which is responsible for performing the use case. Smart meter is secondary actor. The use case in this use case diagram is called “Show Acknowledgment from Smart Meter”.

Description of Use Cases

There is only one use case in this use case diagram. In this use case, after smart meter has been configured successfully, it sends an acknowledgment message to AMI head-end that shows that smart meter has been configured. AMI head-end will show this acknowledgment from smart meter.

The Use Case Diagram

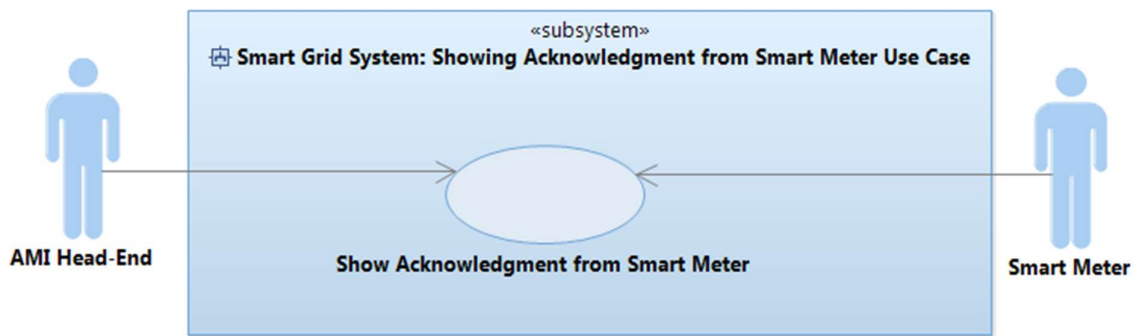


Figure 18 Showing Acknowledgment from Smart Meter Use Case Diagram

Use Case

Table 9 Showing Acknowledgment from Smart Meter Use Case

Showing Acknowledgment from Smart Meter	
Use Case ID	UC_ShowACKfromMeter
Use Case Name	Show Acknowledgment from Smart Meter
Description	This is the use case for showing the acknowledgment from smart meter that smart meter has been configured successfully.
Precondition	Smart meter has been configured successfully.
Primary Actor	AMI head-end
Secondary Actors	Smart meter
dependencies	None
Basic Flow	Steps:

	1	After smart meter is configured, it sends the acknowledgment to AMI head-end that means smart meter is configured.
	2	AMI head-end shows the acknowledgment from smart meter.
	Post Condition	The acknowledgment message has been shown by AMI head-end.

5.3.10 Authentication Use case

Summary

Authenticate use case is a security related use case for sending or receiving packages to/from smart meter. This use case includes different use cases such as “Decrypt Package”, “Create New Session”, “Authorize”, and “Send Package to Smart Meter”.

Description of Use Cases

When the connection is established between smart meter and AMI head-end, smart meter will send its credentials (id and password) to be authenticated by AMI head-end. In the process of smart meter authentication, after smart meter sends its credentials to AMI head-end, smart meter controller verifies smart meter’s credentials. If the verification process is successful, AMI head-end authenticates smart meter. Then smart meter controller creates a session and sends session id to the smart meter. Smart meter saves this session id to use it in later processes. If the authentication process is not successful, smart meter controller sends the error message to smart meter.

The Use Case Diagram

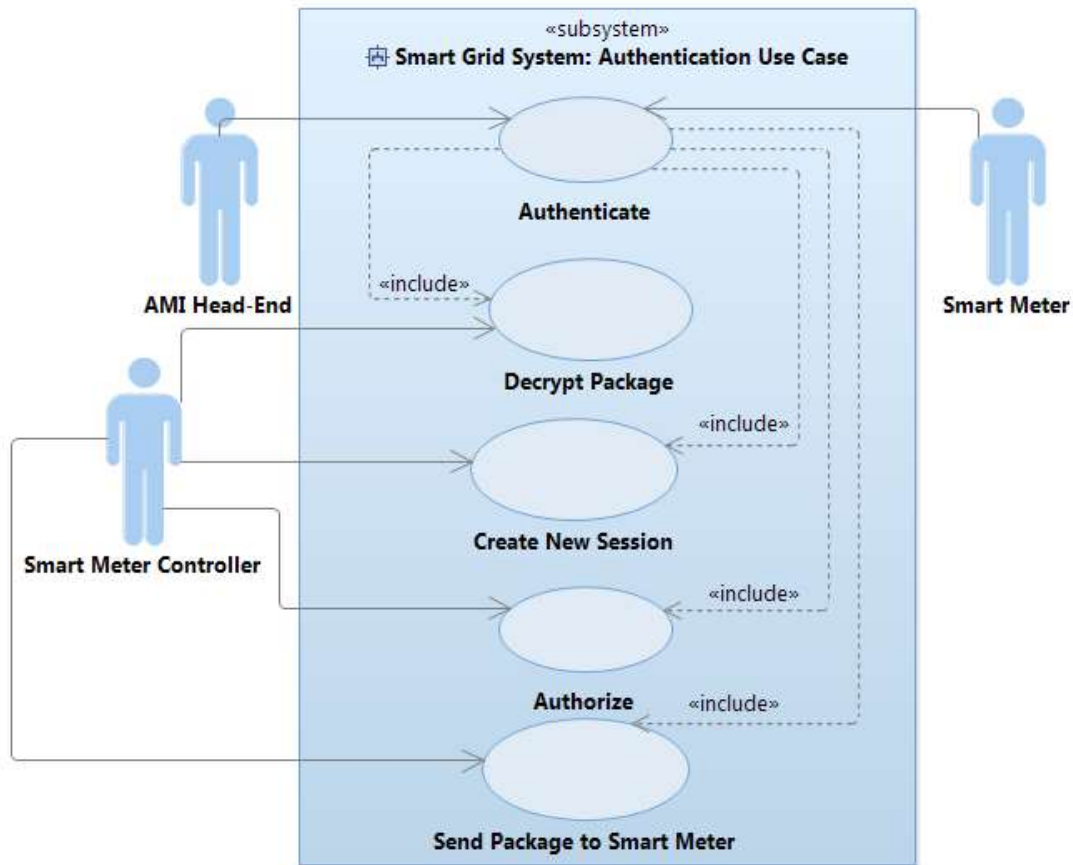


Figure 19 Authentication Use Case Diagram

Use Case

Table 10 Authenticate Use Case

Authenticate	
Use Case ID	UC_Authentication
Use Case Name	Authenticate
Description	This use case is the second step in smart meter registration process. This is a security-related use case. To send a data or message from smart meter to AMI head-end and vice-versa, AMI head-end needs to authenticate smart meter.
Precondition	There is a connection established between AMI head-end and smart meter. Additionally, smart meter has sent its credentials to AMI head-end to be authenticated by AMI head-end.
Primary Actor	Smart meter controller
Secondary Actors	Smart Meter, AMI head-end
Dependencies	INCLUDE USE CASE Decrypt Package INCLUDE USE CASE Create New Session

	INCLUDE USE CASE Authorize INCLUDE USE CASE Send Package to Smart Meter	
Basic Flow	Steps:	
	1	The package code, which includes smart meter's credentials for the authentication of smart meter is received.
	2	Smart meter controller decrypts the received package. (UC_DecryptPackage)
	3	Smart meter controller verifies smart meter's credentials (id and password), which is sent by smart meter.
	4	IF smart meter controller verifies smart meter's credentials THEN
	5	Smart meter controller creates a session object with session id 1 (SID1). (UC_CreateNewSession)
	6	Smart meter controller adds the permissions of smart meter to this session object. (UC_Authorization)
	7	Smart meter controller sends encrypted authentication acknowledgment to smart meter together with the session id (SID1) and configuration parameters. (UC_SendPackageToMeter)
	Post condition	AMI head-end authenticates smart meter.
Specific Alternative Flow	RFS Basic Flow 3	
	Steps:	
	1	ELSEIF smart meter controller does not verify smart meter's credentials THEN
	2	Smart meter controller sends encrypted error message to smart meter. (UC_SendPackageToMeter)
	3	ABORT
	4	ENDIF
	Post condition	AMI head-end does not authenticate smart meter.

5.3.11 Creating New Session Use case

Summary

The use case in this use case diagram is called "Create New Session". The actors are smart meter controller as primary actor and session manager as secondary actor.

Description of Use Cases

This use case is part of or a sub-use case of authentication use case. In the process of creating new session, after smart meter controller verifies the smart meter's credentials in the authentication process, smart meter controller calls the session manager to create the session. Session manager is an actor responsible to handle and manage sessions. Then session manager will create a new session for security purposes and for the later use. AMI head-end will send this session to smart meter.

The Use Case Diagram

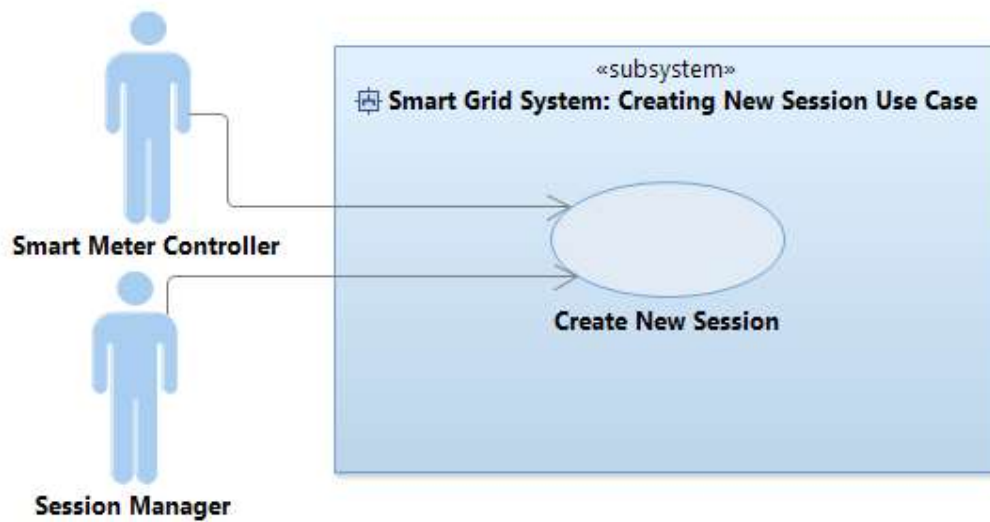


Figure 20 Creating New Session Use Case Diagram

Use Case

Table 11 Creating New Session Use Case

Creating New Session	
Use Case ID	UC_CreateNewSession
Use Case Name	Create New Session
Description	This use case is one of the sub-use cases in the smart meter authentication process. AMI head-end after verifies the smart meter’s credentials, will create the session and will send the session ID to smart meter.
Precondition	AMI head-end has verified and authenticated the smart meter successfully.
Primary Actor	Smart meter controller
Secondary Actors	Session manager
Dependencies	None
Basic Flow	Steps:
	1 Smart meter controller calls the session manger to create the session.
	2 Session manger creates the session.
Post Condition	The session has been created.

5.3.12 Authorization Use Case

Summary

The use case in this use case diagram is “Authorize”. The primary actor is Smart meter controller. The secondary actor is session.

Description of Use cases

Authorization process is part of security design in smart grid system. In this process when receiver receives a package from sender, the receiver should check and make sure if the package, which comes from the legal party, is in the list of allowed packages to be sent. It means the sender should only send the packages that are allowed to be sent. In this case, when AMI head-end receives a package from smart meter, it should make sure if the package that received from the expected smart meter is secure package, not the package with malicious purposes. For example, smart meter can send packages to AMI head-end such as “periodic meter reading”, “on-demand meter reading” or “remote meter connect/disconnect”.

The Use Case Diagram

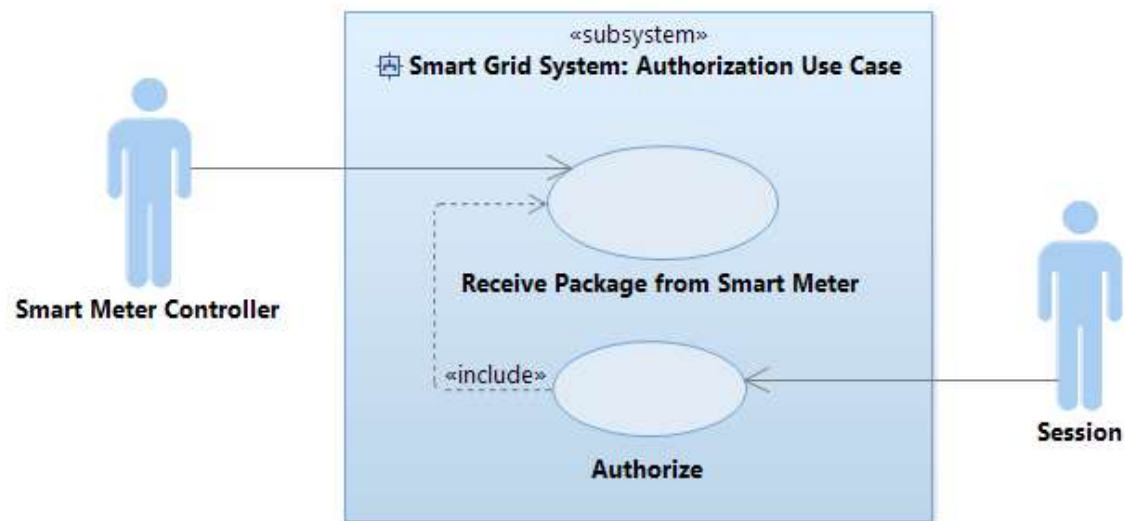


Figure 21 Authorization Use Case Diagram

Use Case

Table 12 Authorization Use Case

Authorize	
Use Case ID	UC_Authorization
Use Case Name	Authorize
Description	This is a use case used for the security purposes. In the process of authorization, when AMI head-end receives a package from smart meter, AMI head-end must check if the package code received from smart meter is in the list of allowed package codes.
Precondition	The session has been created by AMI head-end.

Primary Actor	Smart meter controller	
Secondary Actors	Session	
Dependencies	INCLUDE USE CASE Receive Package from Smart Meter	
Basic Flow	Steps:	
	1	Smart meter controller receives a package from smart meter. (UC_ ReceivingPackageFromMeter)
	2	IF smart meter is authenticated by AMI head-end THEN
	3	Smart meter controller checks the session id of the received package.
	4	Smart meter controller calls the session manager to look up for the session.
	5	Session manager calls the session to get the session id of the session, which is created by AMI head-end.
	6	Session returns the session id of the session, which is created by AMI head-end to smart meter controller.
	7	Smart meter controller matches the received package's session id with created session id.
	8	ENDIF
	9	IF there is a match between the received package's session id and created session id THEN
	10	Smart meter controller sends authorization request to session object.
	11	Session checks the access rights of smart meter to see if the received package code is in the list of access rights of smart meter.
	12	Session returns the authorization decision to smart meter controller.
	13	IF the authorization decision equals to package authorized THEN
	14	Smart meter controller processes the received package.
	15	ELSEIF the authorization decision equals to package not authorized THEN
	16	Session sends error message to smart meter controller.
	17	Smart meter controller shows the error message.
	18	ENDIF
	Post Condition	The received package, which is sent from smart meter, has been authorized.
Specific Alternative Flow	RFS Basic Flow 9	
	Steps:	
	1	ELSEIF there is no match between the received package's session id and created session id THEN
	2	Session sends error message to smart meter controller.
	3	Smart meter controller shows the error message.
	4	ABORT
	5	ENDIF
	Post Condition	The received package, which is sent from smart meter, has not been authorized.

5.3.13 Periodic Meter Reading Use Case

Summary

There is one main and high-level use case in this use case diagram, which is called “Periodic Meter Reading”. This use case includes other use cases such as “Authenticate”, “Receive Package from Smart Meter” and “Authorize” use cases. The primary actor in this use case diagram is AMI head-end. The secondary actors are smart meter controller and smart meter.

Description of Use Cases

Periodic meter reading is the high-level use case. In the process of periodic meter reading from the AMI head-end side, smart meter controller receives a periodic meter read data from smart meter. However, before receiving the meter read data, AMI head-end first should check if the smart meter has already been authenticated by AMI head-end or not. If the session is active, it means smart meter has been authenticated by AMI head-end. In this cases, smart meter controller can receive the package from smart meter. Otherwise, if the session is timed-out, smart meter needs to be re-authenticated by AMI head-end. Therefore, in this case after smart meter has been authenticated again by AMI head-end, smart meter controller can receive the meter read data. After receiving meter read data, it should also be authorized.

The Use Case Diagram

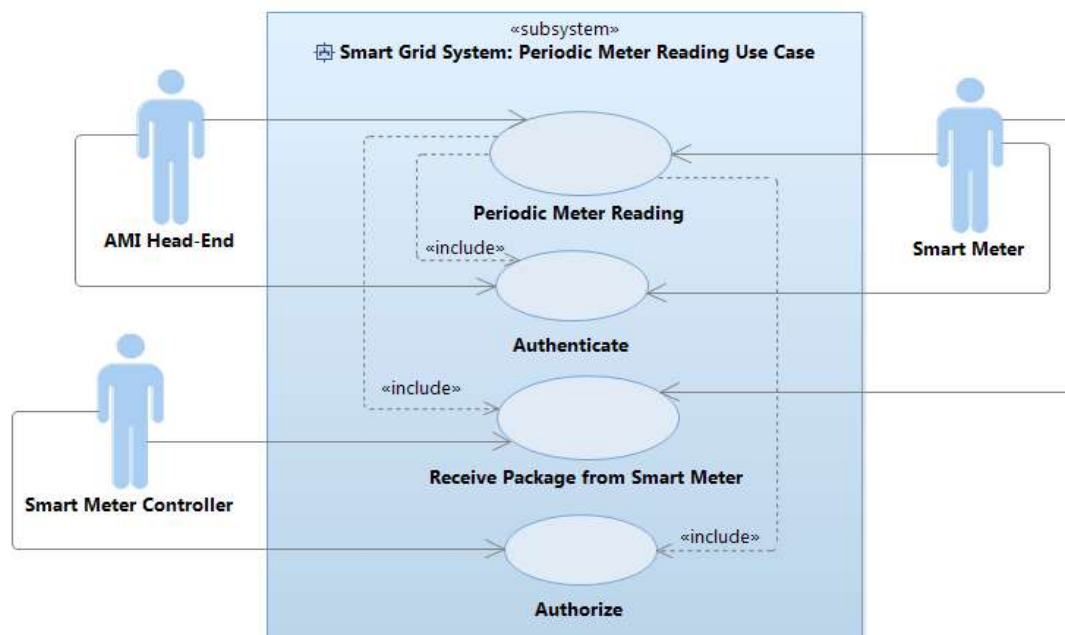


Figure 22 Periodic Meter Reading Use Case

Use Case

Table 13 Periodic Meter Reading Use case

Periodic Meter Reading	
Use Case ID	UC_PeriodicMeterReading
Use Case Name	Periodic Meter Reading
Description	Periodic meter reading is a high-level use case. After the record service, records the 15 minutes electrical usage data, meter metrology board collects the meter read data every 4 hours. Then NIC packages meter read data. NIC sends the meter read data to AMI head-end.
Precondition	Smart meter has sent the periodic meter read data to AMI head-end.
Primary Actor	AMI head-end
Secondary Actors	Smart meter controller, Smart meter
Dependencies	INCLUDE USE CASE Authenticate INCLUDE USE CASE Receive Package from Smart Meter INCLUDE USE CASE Authorize
Basic Flow	Steps:
	1 In order for AMI head-end to receive the package including meter read data from smart meter, AMI head-end needs to check if the smart meter has already authenticated by AMI head-end or if the session is still active or it is timed out. (UC_Authentication)
	2 IF the session is active THEN
	3 Smart meter controller receives the encrypted meter read data from smart meter. (UC_ReceivingPackagefromMeter)
	4 AMI head-end authorizes smart meter. (UC_Authorization)
	Post Condition AMI head-end receives the meter read data.
Specific Alternative Flow	RFS Basic Flow 2
	Steps:
	1 ELSEIF the session is timed out THEN
	2 AMI head-end re-authenticates smart meter by re-using authenticate use case.
	3 RESUME STEP 3
	4 ENDIF
Post Condition -	

5.3.14 Remote Meter Connect/Disconnect Use Case

Summary

There is one main and high-level use case in this use case diagram, which is called “Remote Meter Connect/Disconnect” use case. All other use cases in this use case diagram are reused by this main use case. These use cases are “Send Package to Smart Meter”, “Receive Package from Smart Meter”,

“Authorize” and “Authenticate” use cases. AMI head-end is the primary actor. Smart meter controller and smart meter are the secondary actors in this use case diagram.

Description of Use Cases

In the process of remote meter connect/disconnect, first CIS sends remote meter connect/disconnect request message to AMI head-end. Then head-end needs to send remote meter connect/disconnect message to smart meter. However, before sending the message, smart meter should be already authenticated by AMI head-end. Therefore, AMI head-end checks the session status. If the session is active, smart meter controller sends the remote meter connect/disconnect message to smart meter. Then smart meter controller receives the closed/opened internal meter switch verification message from smart meter. This package is authorized. If the session is timed out, AMI head-end re-authenticates the smart meter. AMI head-end sends the encrypted message to CIS at the end.

The Use Case Diagram

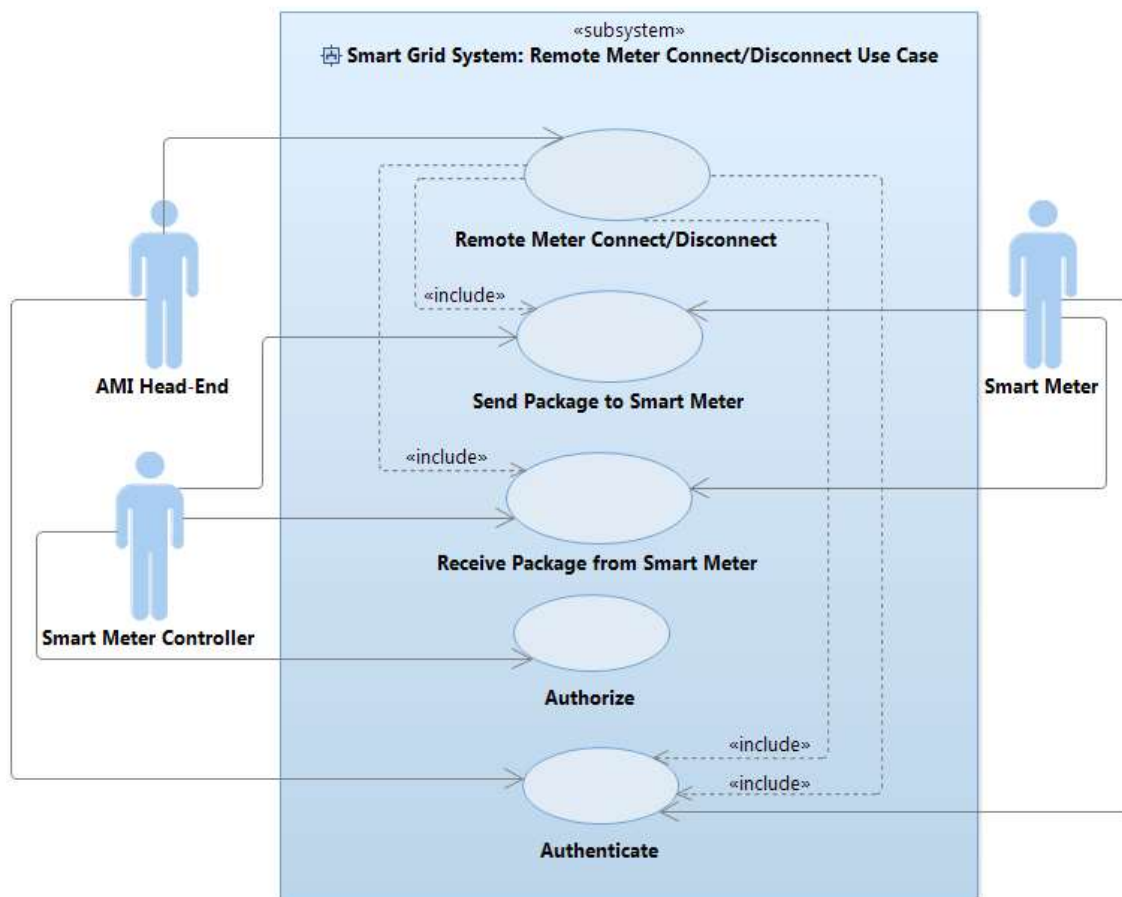


Figure 23 Remote Meter Connect/Disconnect Use Case Diagram

Use Case

Table 14 Remote Meter Connect/Disconnect Use case

Remote Meter Connect/Disconnect	
Use Case ID	UC_RemoteMeterConnect/Disconnect
Use Case Name	Remote Meter Connect/Disconnect
Description	This use case is a high-level use case, which connects smart meter or disconnects smart meter remotely.
Precondition	The smart meter has been registered by AMI head-end already.
Primary Actor	AMI head-end
Secondary Actors	Smart meter controller, Smart meter
Dependencies	INCLUDE USE CASE Send Package to Smart Meter INCLUDE USE CASE Receive Package from Smart Meter INCLUDE USE CASE Authorize INCLUDE USE CASE Authenticate
Basic Flow	Steps:
	1 CIS sends remote meter connect/disconnect request message to AMI head-end.
	2 AMI head-end checks the session status.
	3 IF the session is active THEN
	4 Smart meter controller sends the encrypted remote meter connect/disconnect message to smart meter. (UC_SendingPackageToMeter)
	5 Smart meter controller receives the package from smart meter including closed/opened Internal meter switch verification message. (UC_ReceivingPackageFromMeter)
	6 AMI head-end authorizes smart meter. (UC_Authorization)
	7 AMI head-end sends encrypted closed/opened Internal meter switch verification message to CIS.
	Post Condition The remote meter connect/disconnect message has been sent to smart meter.
Specific Alternative Flow	RFS Basic Flow 3
	Steps:
	1 ELSEIF the session is timed out THEN
	2 AMI head-end authenticates smart meter. (UC_Authentication)
	3 RESUME STEP 4
	4 ENDIF
Post Condition -	

5.3.15 On-Demand Meter Reading Use case

Summary

“On-Demand Meter Reading” use case includes “Send Package to Smart Meter”, “Authorize”, and “Authenticate” use cases. The primary actor is AMI head-end and the secondary actors are smart meter controller and smart meter in this use case diagram.

Description of Use Cases

In the process of on-demand meter reading, first CIS sends on-demand meter read request message to AMI head-end. AMI head-end will send this message to smart meter. However, since this message is sensitive, AMI head-end needs to authenticate smart meter. Therefore, AMI head-end checks the session status. If the session is active, then there is no need to authentication. Smart meter controller sends the on-demand meter read request message to NIC, which is a part of smart meter. NIC sends on-demand meter read data to smart meter controller. The meter read data is authorized. At the end, AMI head-end sends the on-demand meter read data to CIS. On the other hand, if the session is timed-out, it means there is a need for re-authentication process. Therefore, AMI head-end will re-authenticate the smart meter.

The Use Case Diagram

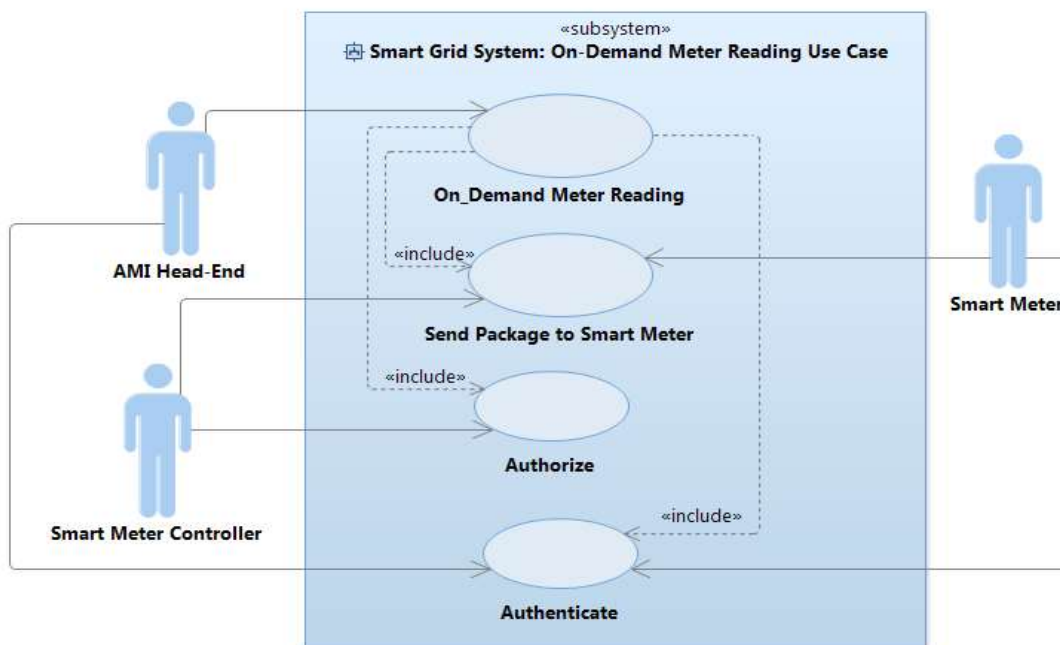


Figure 24 On-Demand Meter Reading Use Case Diagram

Use Case

Table 15 On-Demand Meter Reading Use case

On-Demand Meter Reading	
Use Case ID	UC_On-DemandMeterReading
Use Case Name	On-Demand Meter Reading
Description	On-demand meter reading is a high-level use case. The description of this use case is like this: CIS requests on-demand meter reading, which means reading meter read data based on a demand in the specific date and time or schedule. On-demand meter read request message is sent by CIS to AMI head-end and. AMI head-end sends the on-demand meter read request message to NIC. At the end meter read data will be sent to AMI head-end by smart meter. Then, AMI head-end will send the meter read data to CIS.
Precondition	Smart meter has been registered by AMI head-end.
Primary Actor	AMI head-end
Secondary Actors	Smart meter controller, Smart meter
Dependencies	INCLUDE USE CASE Send Package to Smart Meter INCLUDE USE CASE Authorize INCLUDE USE CASE Authenticate
Basic Flow	Steps:
	1 CIS sends on-demand meter read request message to AMI head-end.
	2 AMI head-end checks the session status.
	3 IF the session is active THEN
	4 Smart meter controller sends the encrypted on-demand meter read request message to smart meter.(UC_SendPackageToMeter)
	5 Smart meter controller receives the on-demand meter read data from smart meter. (UC_ReceivePackageFromMeter)
	6 AMI head-end authorizes smart meter. (UC_Authorization)
	7 AMI head-end sends on-demand meter read data to CIS.
	Post Condition AMI head-end receives on-demand meter read data.
Specific Alternative Flow	RFS Basic Flow 3
	Steps:
	1 ELSEIF the session is timed out THEN
	2 AMI head-end re-authenticates smart meter. (UC_Authentication)
	3 RESUME STEP 4
	4 ENDIF
	Post Condition -

5.4 Use cases of Smart Meter

5.4.1 Table of Actors for Smart Meter Use Cases

Table 16 Table of Actors for Smart Meter Use Cases

Name	Description	Actor/Subsystem
Smart Meter	Smart meter is used for measuring the electricity consumption of consumer. Smart meter communicates with AMI head-end. Smart meter consists of some small parts such as Meter Metrology Board (MMB), NIC, and Internal Meter Switch.	Actor
Meter Metrology Board	Meter Metrology Board is a part of smart meter, which is responsible for performing some operations or functions such as recording the meter’s electrical usage data in formatted table, controlling the internal meter switch to close/open, and retrieving meter read data in formatted table.	Actor
NIC	NIC is the other part of smart meter, which is responsible for transmitting or sending data from Meter Metrology Board to AMI head-end or from AMI head-end to Meter Metrology Board.	Actor
Internal Meter Switch	Internal Meter Switch is a part of smart meter. It executes the RCD (remote connect/disconnect) command to close or open the meter switch. When there is a connect command, it closes the meter switch. Otherwise, when there is a disconnect command it opens the meter switch.	Actor
AMI Head-End	AMI head-end is the back office in the smart grid. It controls the advanced metering infrastructure.	Subsystem

5.4.2 Smart Meter Establishes Connection with AMI Head-End Use Case

Summary

There are three use cases in this use case diagram. The main use case in this diagram is “Establish Two-way Connection”. The other two use cases are “Receive Package from AMI Head-End”, and “Authorize”, which are included from the main use case. The primary actor is Meter metrology board. The secondary actors are NIC and AMI head-end in this use case diagram.

Description of Use Cases

In this use case, smart meter wants to establish a connection with AMI head-end. There is a two-way connection between smart meter and AMI head-end. First, smart meter requests to establish connection with AMI head-end. Then if the connection is ok, NIC will receive a connected

acknowledgment message from AMI head-end that means the connection is established between smart meter and AMI head-end. Otherwise, if there are connection problems, NIC will show error message.

The Use Case Diagram

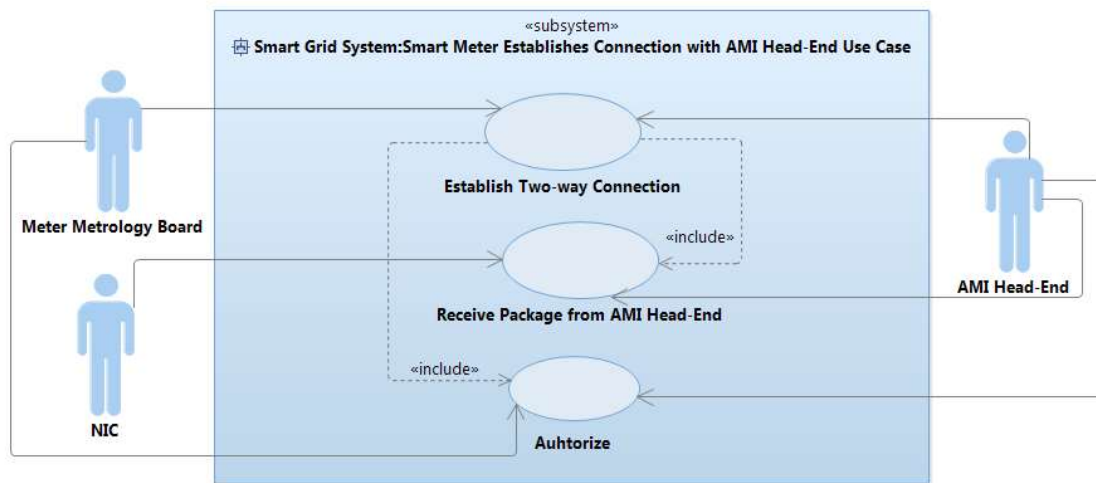


Figure 25 Smart Meter Establishes Connection with AMI Head-End Use Case Diagram

Use Case

Table 17 Smart Meter Establishes Connection with AMI Head-End Use Case

Smart Meter Establishes Connection with AMI Head-End	
Use Case ID	UC_EstablishTwo-wayConnection
Use Case Name	EstablishTwo-wayConnection
Description	This is the first step in registration process of smart meter. In this step, smart meter establishes connection with AMI head-end.
Precondition	AMI head-end has been initialized.
Primary Actor	Meter metrology board
Secondary Actors	NIC, AMI head-end
Dependencies	INCLUDE USE CASE Receive Package from AMI Head-End INCLUDE USE CASE Authorize
Basic Flow	<p>Steps:</p> <ol style="list-style-type: none"> 1 Meter metrology board performs necessary configuration. For example, meter metrology board needs to know the IP address and server port number of AMI Head-end. 2 NIC creates a client socket to connect to the server socket of Head-end with the configured IP and port number. 3 IF the client socket is created successfully THEN 4 NIC creates an output stream for sending package to Head-end MEANWHILE NIC also creates an input stream and a process to

		constantly wait for receiving package from Head-end.
	5	NIC receives the encrypted connected acknowledgment message from AMI head-end. (UC_ReceivingPackageFromHead-End)
	6	MMB authorizes AMI head-end. (UC_Authorization)
	Post condition	There is a connection established between smart meter and AMI head-end, i.e., an acknowledgment message from Head-end is received by smart meter.
Specific Alternative Flow	RFS Basic Flow 3	
	1	ELSEIF there is connection error between smart meter and AMI head-end THEN
	2	NIC shows connection error message.
	3	NIC is reconfigured.
	4	Resume Step 2
	5	ENDIF
	Post condition	NIC is reconfigured.

5.4.3 Sending Package to AMI Head-End Use Case

Summary

The main use case in this use case diagram is “Send Package to AMI Head-End”. The other use case “Encrypt Package” is re-used by this main use case. Meter metrology board is the primary actor. NIC and AMI head-end are the secondary actors.

Description of Use Cases

In the process of sending package to AMI head-end, first smart meter needs to create a package. Then smart meter will encrypt the package using cryptographic algorithms. After that, it will sign the package using digital signatures. At the end, smart meter will send the package to AMI head-end.

In signing package with using digital signature, it could be the case that smart meter in general or meter metrology board in detail, cannot sign the package. One reason that MMB cannot sign the package is due to uncertainty. The attacker can introduce uncertainty related to package signing in the AMI system. We explain this issue in the section for examples of smart grid uncertainties.

The Use Case Diagram

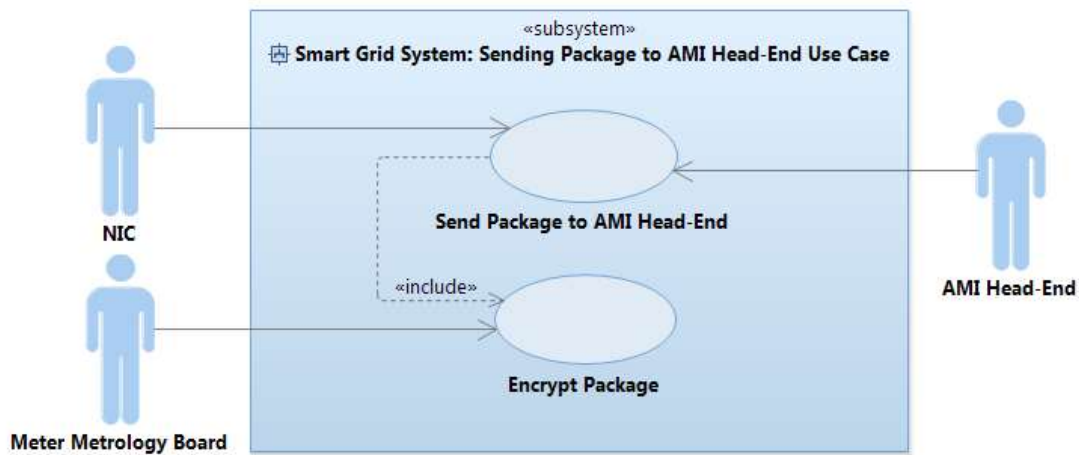


Figure 26 Sending Package to AMI Head-End Use Case Diagram

Use Case

Table 18 Sending Package to AMI Head-End Use Case

Sending package to AMI head-end	
Use Case ID	UC_SendingPackageToHead-End
Use Case Name	Send Package to AMI Head-End
Description	This is a general use case for specifying how smart meter sends packages to AMI head-end.
Precondition	A two-way communication channel from smart meter to head-end has been created.
Primary Actor	Meter Metrology Board
Secondary Actors	NIC, AMI head-end
Dependencies	INCLUDE USE CASE Encrypt Package
Basic Flow	<p>Steps:</p> <ol style="list-style-type: none"> 1 Meter metrology board creates a package to send to AMI head-end. 2 Meter metrology board encrypts the package. (UC_Encrypt Package) 3 Meter metrology board signs the package. (UC_Encrypt Package) 4 IF meter metrology board signs the package THEN 5 NIC sends the package to AMI head-end. <p>Post Condition The package is sent to AMI head-end.</p>
Specific Alternative Flow	<p>RFS Basic Flow 3</p> <p>Steps:</p> <ol style="list-style-type: none"> 1 ELSEIF meter metrology board does not sign the package THEN 2 Meter metrology board creates a new concurrent process (thread) to send a package with error message/code to AMI head-end.

		(UC_SendPackageToHead-End)
	3	ABORT
	4	ENDIF
	Post Condition	A package with error message/code is sent to AMI head-end.

5.4.4 Receiving Package from AMI Head-End Use Case

Summary

This use case is a general use case for receiving all kinds of packages from AMI head-end. The main use case in this use case diagram is “Receive Package from AMI Head-End”. This use case includes “Decrypt Package”, “Response to AMI Head-End”, and “Send Package to AMI Head-End” use cases. The primary actor is NIC and the secondary actors are meter metrology board and AMI head-end.

Description of Use cases

In the process of receiving package, smart meter waits for receiving package from AMI head-end. Then NIC will receive a package from AMI head-end. After that, Meter metrology board, which is part of smart meter, will verify the received package. It will decrypt the received package. If the package is from expected AMI head-end, then it will process the package, otherwise it will send an error message to AMI head-end.

The Use Case Diagram

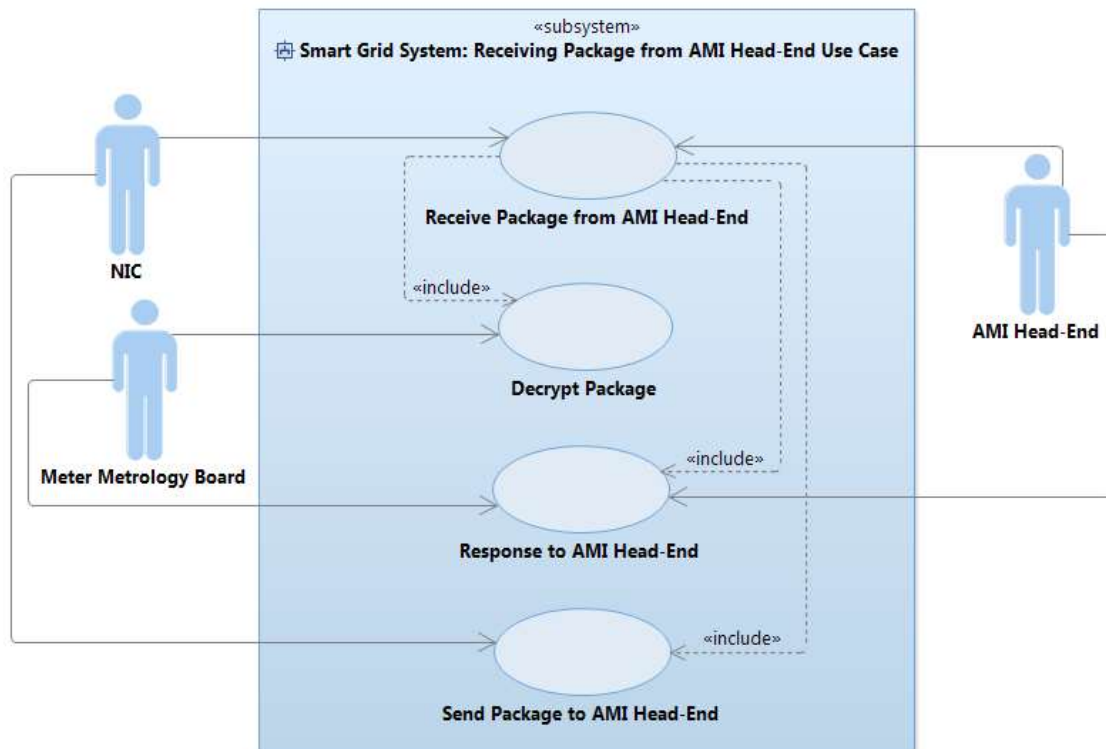


Figure 27 Receiving Package from AMI Head-End Use Case Diagram

Use Case

Table 19 Receiving Package from AMI Head-End Use Case

Receiving package from AMI head-end	
Use Case ID	UC_ReceivingPackageFromHead-End
Use Case Name	Receive Package from AMI Head-End
Description	This is a general use case for specifying how smart meter receives packages from AMI head-end.
Precondition	A two-way communication channel from smart meter to head-end has been created.
Primary Actor	NIC
Secondary Actors	Meter metrology board, AMI head-end
Dependencies	INCLUDE USE CASE Decrypt Package INCLUDE USE CASE Response to AMI Head-End INCLUDE USE CASE Send Package to AMI Head-End
Basic Flow	Steps:
	1 DO
	2 Smart meter is waiting to receive package from AMI head-end.
	3 NIC, which is part of smart meter, receives a package from AMI head-

		end.
	4	Meter Metrology Board (MMB), which is part of smart meter, verifies the package's digital signature. (UC_DecryptPackage)
	5	MMB decrypts the received package. (UC_DecryptPackage)
	6	IF MMB verifies the package's digital signature THEN
	7	MMB creates a new concurrent process (thread) to process the decrypted package (UC_ResponseToHead-End) MEANWHILE smart meter continues waiting for receiving packages from AMI head-end.
	8	UNTIL the server socket is closed.
	Post Condition	The package is received from AMI head-end.
Specific Alternative Flow	RFS Basic Flow Step 4	
	Steps:	
	1	ELSEIF MMB does not verify the digital signature or decrypt the received package from AMI head-end THEN
	2	MMB creates a new concurrent process (thread) to send a package with error message/code to AMI head-end. (UC_SendPackageToHead-End)
	3	ABORT
	4	ENDIF
	Post Condition	A package with error message/code is sent to head-end.

5.4.5 Encrypting Package Use case

Summary

The use case in this use case diagram is called "Encrypt Package". The actor to perform this use case is meter metrology board.

Description of Use Cases

In the process of encrypting package, the sender first should encrypt the package before sending the package to another party. Here, smart meter wants to send a package to AMI head-end. Therefore, MMB encrypts and signs the package before sending it to AMI head-end by using cryptographic algorithms. Then MMB sends the encrypted package to AMI head-end.

The Use Case Diagram

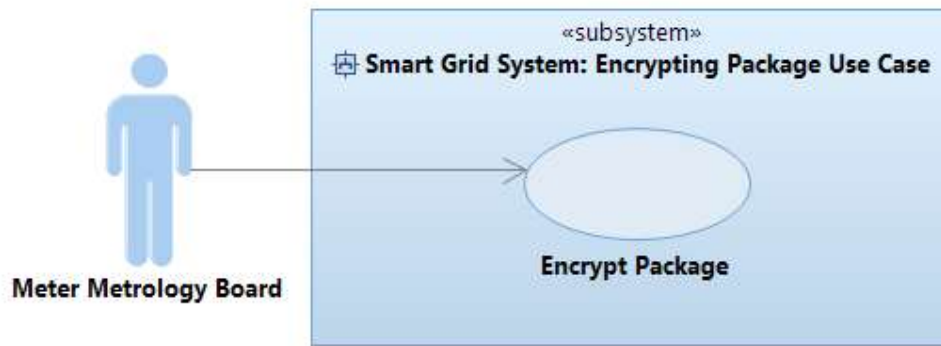


Figure 28 Encrypting Package Use Case Diagram

Use Case

Table 20 Encrypting Package Use Case

Encrypting Package	
Use Case ID	UC_ EncryptPackage
Use Case Name	Encrypt Package
Description	This is a use case used for the security purposes. When smart meter wants to send a package to AMI head-end, it first encrypts and signs the package before sending. Then, it will send the package to AMI head-end.
Precondition	Smart meter has created a package for sending to AMI head-end.
Primary Actor	Meter metrology board (MMB)
Secondary Actors	None
Dependencies	None
Basic Flow	<p>Steps:</p> <ol style="list-style-type: none"> 1 MMB encrypts the package by using the cryptographic algorithms. 2 MMB signs the package using digital signature. 3 IF MMB signs the package THEN 4 MMB sends the package to AMI head-end via NIC. <p>Post Condition The package for sending has been encrypted.</p>
Specific Alternative Flow	<p>RFS Basic Flow 3</p> <p>Steps:</p> <ol style="list-style-type: none"> 1 ELSEIF MMB does not sign the package THEN 2 MMB sends the error message to AMI head-end via NIC 3 ABORT 4 ENDIF <p>Post Condition A package with error message/code is sent to head-end.</p>

5.4.6 Decrypting Package Use Case

Summary

The use case in this use case diagram is called “Decrypt Package”. The actor to perform this use case is meter metrology board.

Description of Use cases

In “Decrypt Package” use case when smart meter receives a package, first meter metrology board should verify the digital signature of received package sent by AMI head-end. Then meter metrology board decrypts the package to read the content of the package. After that, meter metrology board can process the package based on the package code.

The Use Case Diagram

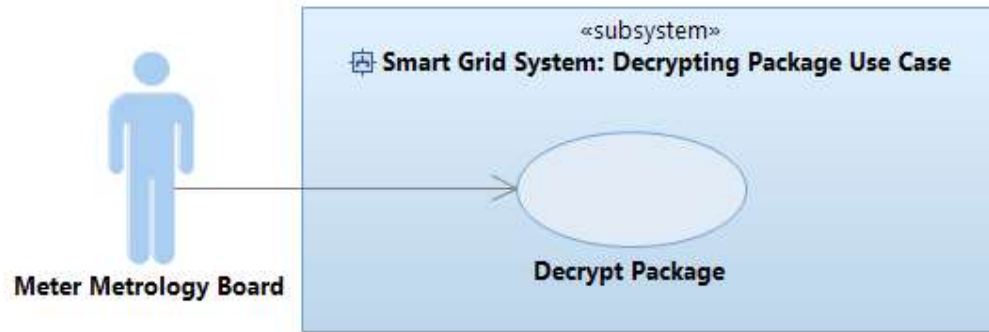


Figure 29 Decrypting Package Use Case Diagram

Use case

Table 21 Decrypting Package Use Case

Decrypting package	
Use Case ID	UC_ DecryptPackage
Use Case Name	Decrypt Package
Description	This is the use case used for the security purposes. When smart meter receives a package from AMI head-end, it decrypts the package to read the content of the package.
Precondition	Smart meter has received the package from AMI head-end.
Primary Actor	Meter metrology board (MMB)
Secondary Actors	None
Dependencies	None
Basic Flow	Steps:
	1 MMB verifies the digital signature of received package.

	2	IF MMB verifies the digital signature of received package THEN
	3	MMB decrypts the received package.
	4	MMB processes the received package.
	Post Condition	The received package has been decrypted.
Specific Alternative Flow	RFS Basic Flow 1	
	Steps:	
	1	ELSEIF MMB does not verify the digital signature of received package THEN
	2	MMB sends an error message to AMI head-end via NIC.
	3	ABORT
	4	ENDIF
	Post Condition	A package with error message/code is sent to AMI head-end.

5.4.7 Response to AMI Head-End Use Case

Summary

“Response to AMI head-End” use case is a general use case to show how smart meter will respond to AMI head-end after it received the package from AMI head-end. Depending the package code, smart meter will respond to AMI head-end differently. There are different use cases in this use case diagram. The main use case is “Response to AMI Head-End”. Other use cases are included from this use case. These use cases are “Authenticate”, “Remote Meter Connect/Disconnect”, and “On-Demand Meter Reading”. Meter metrology board is the primary actor and AMI head-end is the secondary actor in this use case diagram.

Description of Use Cases

The main use case in this use case diagram is “Response to AMI Head-End”. Smart Meter after receiving the packages will process the packages depending on the package code and will respond to AMI head-end. For example, if the package code is the acknowledgment message of AMI head-end that the connection is established between smart meter and AMI head-end, then meter metrology board sends smart meter’s credentials to be authenticated by AMI head-end.

The Use Case Diagram

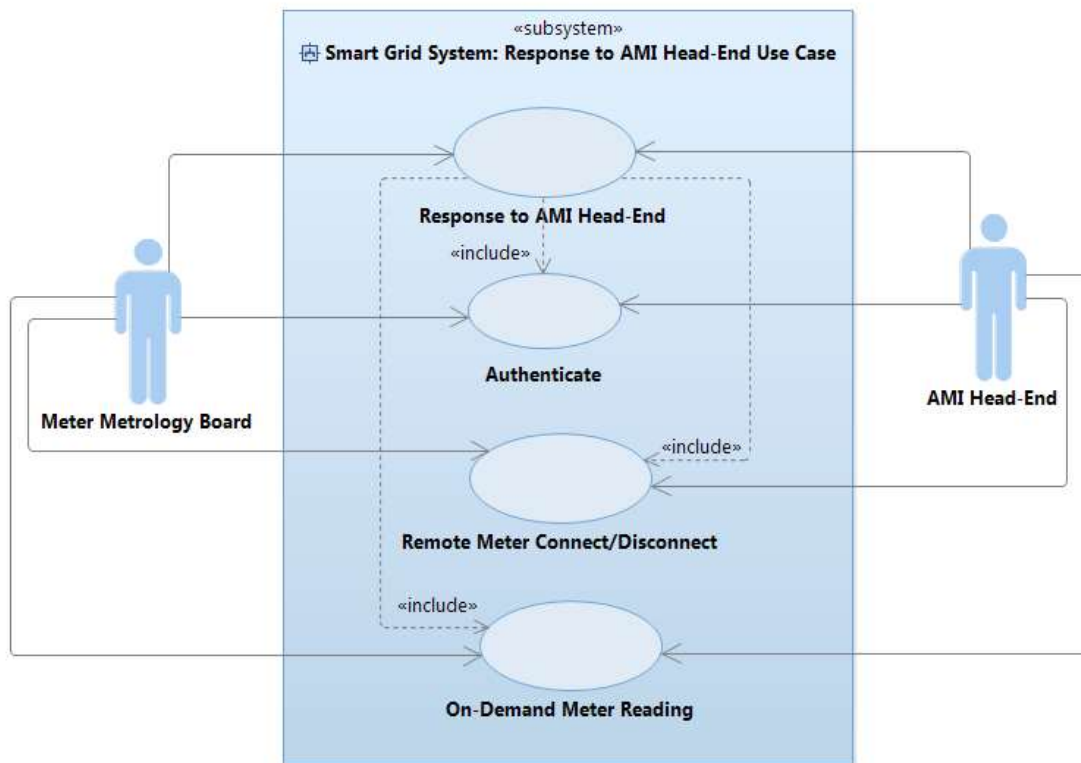


Figure 30 Response to AMI Head-End Use Case Diagram

Use Case

Table 22 Response to AMI Head-End Use Case

Response to AMI head-end	
Use Case ID	UC_ResponseToHead-End
Use Case Name	Response to AMI Head-End
Description	This is a general use case for specifying how smart meter processes a received package from AMI head-end.
Precondition	A package from AMI head-end has been received and decrypted.
Primary Actor	Meter metrology board (MMB)
Secondary Actors	AMI head-end
Dependencies	INCLUDE USE CASE Authenticate INCLUDE USE CASE Remote Meter Connect/Disconnect INCLUDE USE CASE On-Demand Meter Reading
Basic Flow	Steps:
	1 Meter metrology board reads the decrypted package to check the package code.
	2 IF the package code is the ACK message of AMI head-end that the connection is established between smart meter and AMI head-end

		THEN
	3	Meter metrology board sends smart meter's credentials to AMI head-end to be authenticated by AMI head-end. (UC_Authnetication)
	4	ELSEIF the package code is ACK message from AMI head-end that meter is authenticated THEN
	5	Meter metrology board saves the session id sent by AMI head-end. (UC_Authentication)
	6	ELSEIF the package code is remote meter connect/disconnect message THEN
	7	Meter metrology board controls internal meter switch. (UC_RemoteMeterConnect/Disconnect)
	8	ELSEIF the package code is on-demand meter read request message THEN
	9	Meter metrology board retrieves meter read data in formatted table. (UC_On-DemandMeterReading)
	10	ENDIF
	Post Condition	The package from AMI head-end has been processed. A response from smart meter is sent back to AMI head-end.

5.4.8 Authenticate Use Case

Summary

Authenticate use case is a security related use case for sending or receiving packages to/from smart meter. This use case includes use cases such as "Receive Package from AMI Head-End", "Authorize", and "Send Package to AMI Head-End".

Description of Use Cases

When the connection is established between smart meter and AMI head-end, smart meter will send its credentials (id and password) to be authenticated by AMI head-end. In the process of smart meter authentication, after smart meter sends its credentials to AMI head-end, head-end verifies smart meter's credentials. If the verification process is successful, smart meter receives the acknowledgment from head-end that meter is authenticated together with the session id. Then AMI head-end creates a session and sends session id to the smart meter. Smart meter saves this session id to use it in later processes. If the authentication process is not successful, AMI head-end sends the error message to smart meter and smart meter shows this error message.

The Use Case Diagram

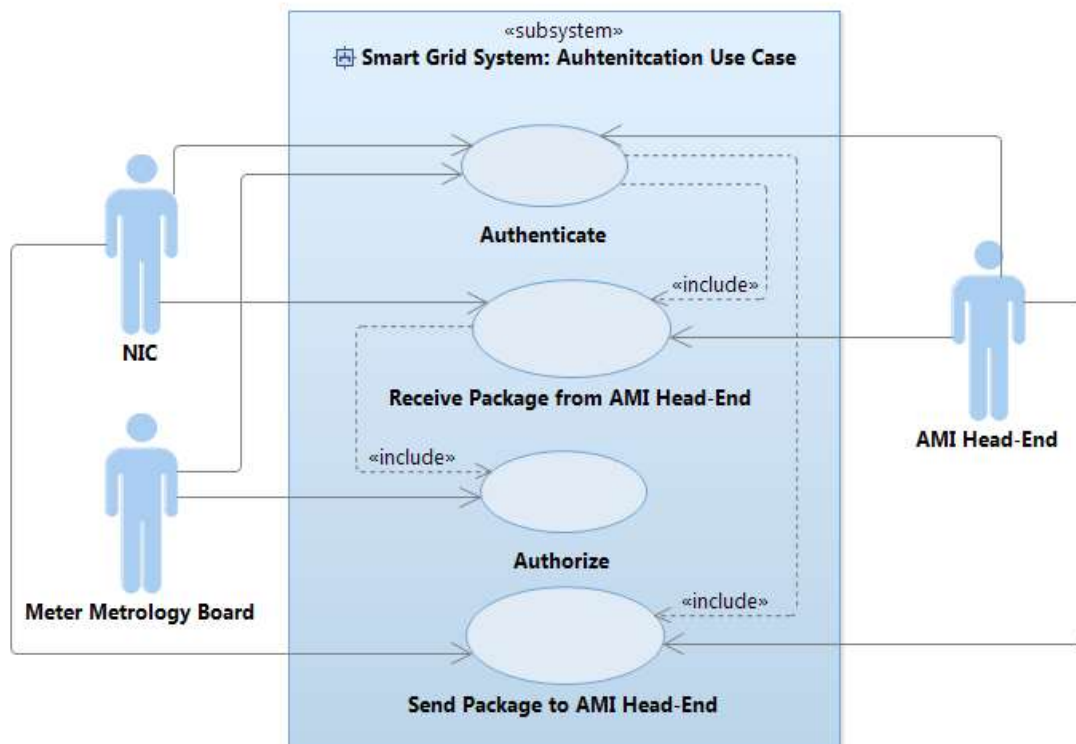


Figure 31 Authentication Use Case Diagram

Use Case

Table 23 Authenticate Use Case

Authenticate	
Use Case ID	UC_Authentication
Use Case Name	Authenticate
Description	This use case is the second step in smart meter registration process. This is a security-related use case. To send a data or message from smart meter to AMI head-end and vice-versa, AMI head-end needs to authenticate smart meter.
Precondition	There is a connection established between smart meter and AMI head-end.
Primary Actor	NIC
Secondary Actors	Meter metrology board, AMI head-end
Dependencies	INCLUDE USE CASE Receive Package from AMI Head-End INCLUDE USE CASE Authorization INCLUDE USE CASE Send Package to AMI Head-End
Basic Flow	Steps:
	1 NIC receives encrypted connected acknowledgment message from AMI head-end. (UC_ReceivingPackageFromHead-End)
	2 Meter metrology board authorizes AMI head-end. (UC_Authorization)

	3	NIC sends smart meter's credentials (smart meter's id and password), which are encrypted to AMI head-end (UC_SendingPackageToHead-End)
	4	IF AMI head-end authenticates smart meter THEN
	5	NIC receives the encrypted acknowledgment message from head-end that meter is authenticated together with the session id (SID1). (UC_ReceivingPackageFromHead-End)
	6	Meter metrology board saves the session id (SID1), which AMI head-end sends.
	Post Condition	AMI head-end has authenticated smart meter.
Specific Alternative Flow	RFS Basic Flow 2	
	Steps:	
	1	ELSEIF AMI head-end does not authenticate smart meter THEN
	2	NIC receives error message from AMI head-end. (UC_ReceivingPackageFromHead-End)
	3	Meter metrology board shows the error message.
	4	ABORT
	5	ENDIF
	Post Condition	AMI head-end has not authenticated smart meter.

5.4.9 Periodic Meter Reading Use Case Diagram

Summary

This use case diagram shows the main use cases related to periodic meter reading. It shows the process of recording meter read data by Meter Metrology Board. It also shows the process of encrypting meter read data before sending it to AMI head-end. The main and high-level use case in this use case diagram is "Periodic Meter Reading". This use case includes "Record the Meter's Electrical Usage Data", "Encrypt Package" and "Send Package to AMI Head-End" use cases. Meter metrology board is the primary actor. NIC and AMI head-end are the secondary actors in this use case diagram.

Description of use cases

In the process of periodic meter reading, first meter metrology board records the power electricity consumption for 15 minutes interval. After that, it collects the meter read data in formatted table. Every 4 hours Meter Metrology Board sends the meter read data (including 15 minutes interval) to NIC. NIC encrypts the meter read data for security purposes. Then, NIC sends the encrypted meter read data to AMI head-end.

Use case diagram

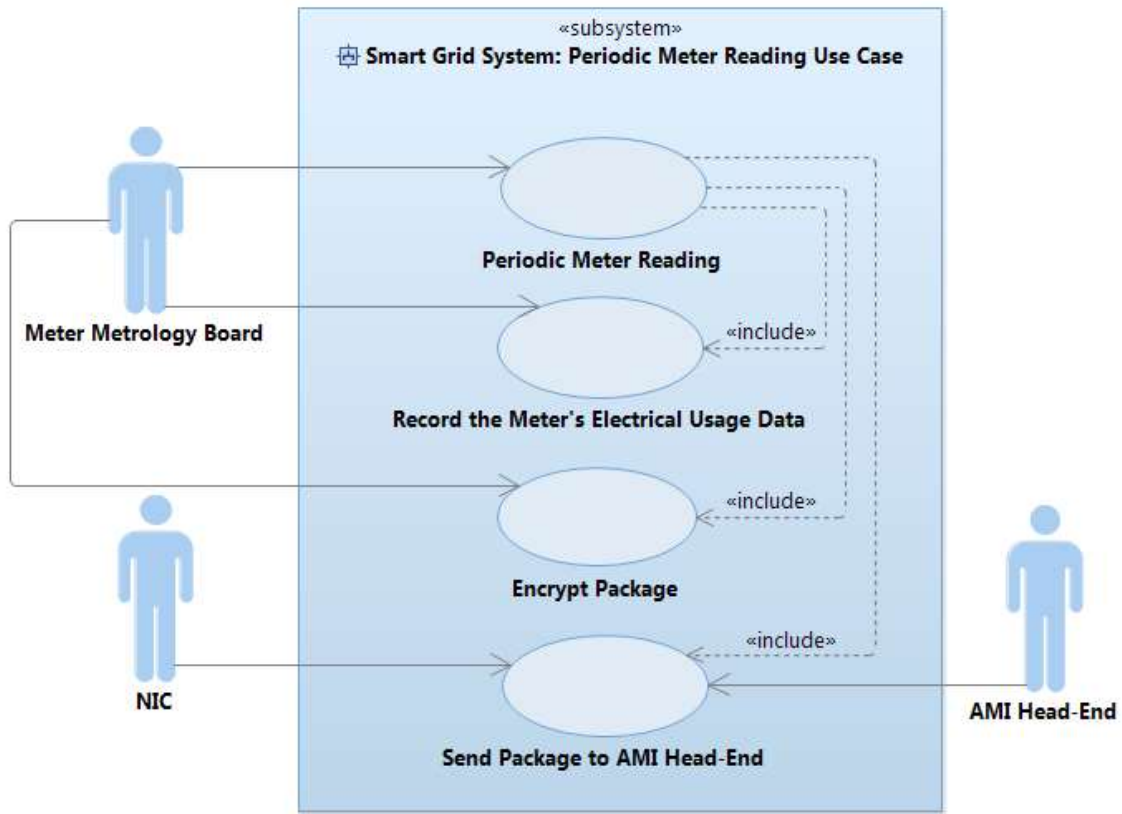


Figure 32 Periodic Meter Reading Use Case Diagram

Use Case

Table 24 Periodic Meter Reading Use Case

Periodic Meter Reading	
Use Case ID	UC_PeriodicMeterReading
Use Case Name	Periodic Meter Reading
Description	Periodic meter reading is a high-level use case. After Meter metrology board records the meter electrical usage data, it collects the meter read data every 4 hours. Then NIC packages meter read data. NIC sends the meter read data to AMI head-end.
Precondition	Record service has recorded meter read data.
Primary Actor	Meter Metrology Board
Secondary Actors	NIC, AMI head-end
Dependencies	INCLUDE USE CASE Record the Meter's Electrical Usage Data INCLUDE USE CASE Encrypt Package INCLUDE USE CASE Send Package to AMI Head-End
Basic Flow	Steps:
	1 DO
	2 Meter metrology board records meter's electrical usage data every 15

		minutes. (UC_RecordingMeterElectricalData)
3		UNTIL recording process has been done for 15 minutes interval.
4		DO
5		Every 4 hours, Meter Metrology Board, collects meter read data in formatted table, which has been recorded so far in (UC_RecordingMeterElectricalData).
6		Meter metrology board encrypts meter read data before sending to AMI head-end. (UC_EncryptPackage)
7		NIC packages meter read data.
8		NIC sends the encrypted meter read data to AMI head-end. (UC_SendingPackageToHead-End)
9		UNTIL the server socket is closed.
	Post Condition	The meter read data has been sent to AMI head-end.

5.4.10 Recording the Meter's Electrical Usage Data Use Case Diagram

Summary

The use case in this use case diagram is called "Record the Meter's Electrical Usage Data". This use case is a sub-use case of "Periodic Meter Reading" use case. The actor to perform this use case is meter metrology board.

Description of Use Cases

In this use case, meter metrology board creates the record service. The, record service will record the meter's electrical usage data for 15 minutes interval.

The Use Case Diagram

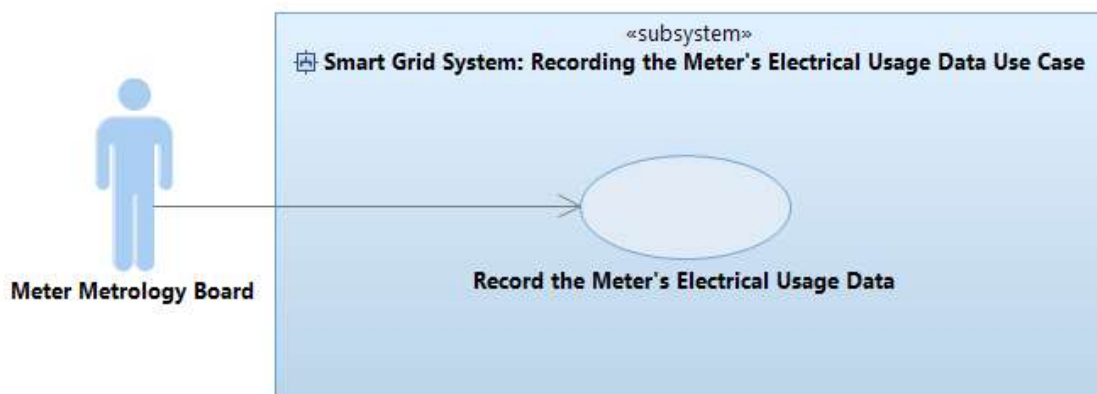


Figure 33 Recording the Meter’s Electrical Usage Data Use case Diagram

Use Case

Table 25 Record the Meter’s Electrical Usage Data Use case

Record the Meter’s Electrical Usage Data	
Use Case ID	UC_RecordingMeterElectricalData
Use Case Name	Record the Meter’s Electrical Usage Data
Description	After registering the smart meter by AMI head-end, Meter Metrology Board can record the electricity power usage for 15 minutes interval locally in the system.
Precondition	The smart meter has been registered by AMI head-end. Additionally the smart meter has got the interval that it can record the electric usage.
Primary Actor	Meter Metrology Board
Secondary Actors	None
Dependencies	None
Basic Flow	Steps:
	1 Meter Metrology Board creates the record service object.
	2 DO
	3 Record Service, which is part of meter metrology board records the meter’s electrical usage data for 15 minutes interval.
	4 UNTIL recording process has been done for 15 minutes interval.
Post Condition	The meter’s electrical usage data has been recorded.

Security Requirements of Periodic Meter Reading Use Case

One of the security requirements for Periodic Meter Reading use case diagram is **Privacy** or **Confidentiality** of data. That means the data should be private and not public to other customers. Only the customer who wants to read his/her electricity consumption should read this data. For example, in periodic meter reading, smart meter transmits the data to AMI head-end. It should send the data to the expected AMI head-end [3]. Of course, privacy is little bit different form confidentiality. Privacy is more from the users or consumers point of view. It means to send data to private users not to public. Confidentiality is more about the data. To keep the data confidential means not to disclose data to everybody. To protect confidentiality and privacy of data, one solution is to use **Encryption**. By using **Cryptographic Algorithms** for encryption, it is possible to secure the data. Additionally, it makes data more confidential and private. When the data is encrypted, there is some coding. Therefore, if attackers try to access data and read it, it would be impossible or difficult. Because the data is encrypted [3].

The **Integrity** of meter read data is also important in periodic meter reading. When smart meter transmits data to AMI head-end, the data should be not modified [3]. There are some techniques for

protecting integrity of data. One of these techniques is **Output Validation**. It is performed at the receiving end to confirm the data is not changed during transmission [51].

Authentication is also important in transmitting meter data from one utility to another utility. For example, when smart meter sends data to AMI head-end, it should be proved that it is the same smart meter that was supposed to send the data. Therefore, the smart meter should be authenticated. Authentication means to prove that the source is the same as it is expected to be [51]. Additionally, all the data transmitted in the smart grid must be authenticated [52]. To have authentication in the system, there are some solutions. **Ids, Passwords**, or some kinds of **keys** like **Cryptographic Keys** are some solutions. For example, when smart meter sends data to AMI head-end, smart meter can have id or password. It shows which smart meter is supposed to send data to AMI head-end. Based on these id and password, the AMI head-end can recognize the smart meter.

5.4.11 Remote Meter Connect/Disconnect Use Case Diagram

Summary

In this use case, the smart meter can be connected or disconnected for applying some changes in the system. For example, if the consumer does not pay the cost, the smart meter is disconnected. The main and high-level use case in this use case diagram is “Remote Meter Connect/Disconnect”. This use case includes “Authenticate”, “Receive Package from AMI Head-End”, “Authorize”, “Encrypt Package”, and “Send Package to AMI Head-End” use cases. The primary actor is NIC. The secondary actors are meter metrology board, internal meter switch and AMI head-end.

Description of use cases

The main use case in this use case diagram is “Remote meter Connect/Disconnect”. NIC receives remote meter connect/disconnect message sent by AMI head-end. Then Meter Metrology Board controls Internal Meter Switch. Internal Meter Switch, which is part of smart meter, executes the RCD command to close/open the meter switch. Meter Metrology Board creates Internal Meter Switch Verification message. Meter metrology board encrypts the message. Then NIC sends the encrypted Internal Meter Switch Verification message to AMI head-end.

Use Case Diagram

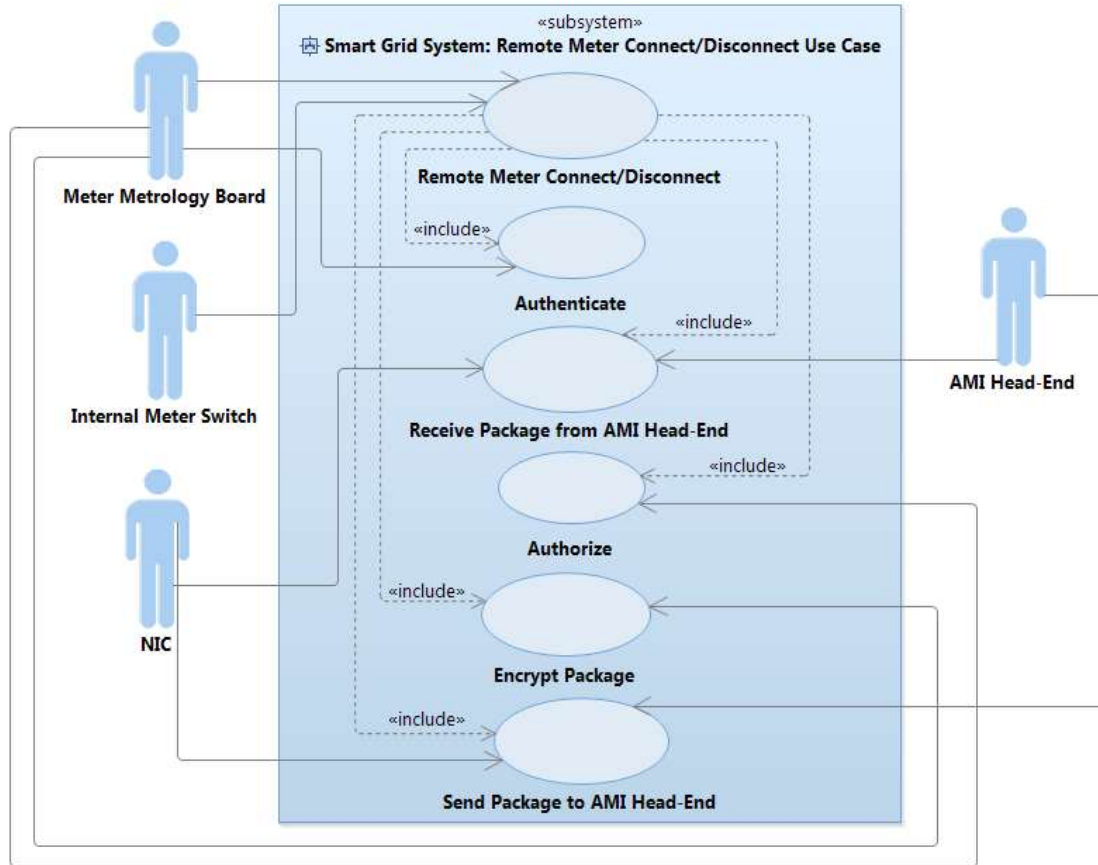


Figure 34 Remote Meter Connect/Disconnect Use case Diagram

Use Case

Table 26 Remote Meter Connect/Disconnect Use case

Remote Meter Connect/Disconnect	
Use Case ID	UC_RemoteMeterConnect/Disconnect
Use Case Name	Remote Meter Connect/Disconnect
Description	This use case is a high-level use case, which connects smart meter or disconnects smart meter remotely.
Precondition	AMI head-end has sent the remote meter connect/disconnect message to smart meter.
Primary Actor	NIC
Secondary Actors	Meter Metrology Board, Internal Meter Switch, AMI Head-End
Dependencies	INCLUDE USE CASE Authenticate INCLUDE USE CASE Receive Package form AMI Head-End INCLUDE USE CASE Authorize INCLUDE USE CASE Encrypt Package

INCLUDE USE CASE Send Package to AMI Head-End	
Basic Flow	Steps:
	1 Meter metrology board saves the session id received by AMI head-end. (UC_Authentication)
	2 NIC receives the encrypted remote meter connect message from AMI head-end. (UC_ReceivingPackageFromHead-End)
	3 Meter metrology board authorizes AMI head-end. (UC_Authorization)
	4 Meter metrology board controls the Internal Meter Switch.
	5 Internal meter switch, which is part of smart meter, executes the RCD command to close/open the meter switch (when there is a connection the meter switch is closed, otherwise it is opened).
	6 Internal meter switch closes/opens the meter switch.
	7 Internal meter switch sends RCD executed message to meter metrology board to show that it has executed the RCD command.
	8 Meter metrology board encrypts the Internal meter switch verification message before sending to AMI head-end. (UC_EncryptPackage)
	9 NIC sends the encrypted closed/opened Internal meter switch verification message to AMI head-end. (UC_SendingPackageToHead-End)
Post Condition	AMI head-end receives the closed/opened Internal meter switch verification message.

Security Requirements of Remote Meter Connect/Disconnect Use Case

One of the security requirements for Meter remote connect/disconnect use case is **Integrity**. The remote connect/disconnect message should not be changed [53]. For example, if the goal is to connect smart meter, the message should be remote meter connect message, not disconnect message. If the disconnect message is sent instead of connect message, it can lead to some problems. For example, the customer might think that the smart meter is corrupted or not working. Therefore, he/she can decide to buy new smart meter and other equipment. Additionally, remote meter connect/disconnect message should be protected against deletion. It means the message should not be deleted. Other purpose of integrity is to prevent fake messages or fake senders and receivers [53]. To prevent manipulation and preserve the integrity, some **Cryptographic methods** are used. **Cipher-Based Message Authentication Code (CMAC)** is an example of cryptographic methods [3].

The other security requirement for meter remote connect/disconnect use case is **Availability**. The smart meter and AMI system should be always available [53]. For example, if the goal is to send remote meter connect/disconnect message from AMI head-end to smart meter, smart meter should be available. If there is no smart meter, the message cannot be transmitted further to other parts. This means remote meter connect/disconnect operation is performed unsuccessfully. Of course, only a single meter is not a big deal. The operator can detect if a smart meter is having any problem, and fix

it. That is why there is the **Power Outage Notification** as well. To solve lack of availability, we can use mechanisms such as **Redundancy of Recourses**. By this technique, we will have more resources that help to make the resources available [7].

Availability of AMI Head-End is more important than availability of a single smart meter. Because AMI Head-End has to serve thousands or millions of meters. However, in case of AMI system, we can say that we prioritize Confidentiality and Integrity.

Remote Access Policy and Procedures is another security requirement for this use case. This means that the data or message can be accessible only by authorized authorities. For example, when AMI head-end sends remote meter connect/disconnect message to smart meter, this message should be accessible only by authorized smart meter. There are mechanisms such as **Monitoring access activities**. Monitoring prevents remote access by unauthorized smart meter or AMI head-end [3].

Remote Access is another security requirement for remote meter connect/disconnect use case. Remote access is any access to a smart grid through external device such as internet. To protect remote access against wireless access, **Authentication** and **Encryption** are used. **Cryptography** is also used to prevent lack of confidentiality and lack of integrity of remote access [3].

5.4.12 On-Demand Meter Reading Use Case Diagram

Summary

On-demand meter reading is similar to periodic meter reading. The difference is that in on-demand meter reading, the process of meter reading is based on the demand. There is a demand on a specific date and time. The main use case in this use case diagram is “On-Demand Meter Reading”. This use case includes “Authenticate”, “Receive Package from AMI Head-End”, “Authorize”, “Encrypt Package”, and “Send Package to AMI Head-End” use cases. NIC is the primary actor. Meter metrology board and AMI head-end are the secondary actors.

Description of use cases

In “On-Demand Meter Reading” use case, NIC receives the on-demand meter read request message from AMI head-end. Then meter metrology board retrieves meter read data in formatted table. NIC encrypts meter read data and it sends the on-demand meter read data to AMI head-end.

Use Case Diagram

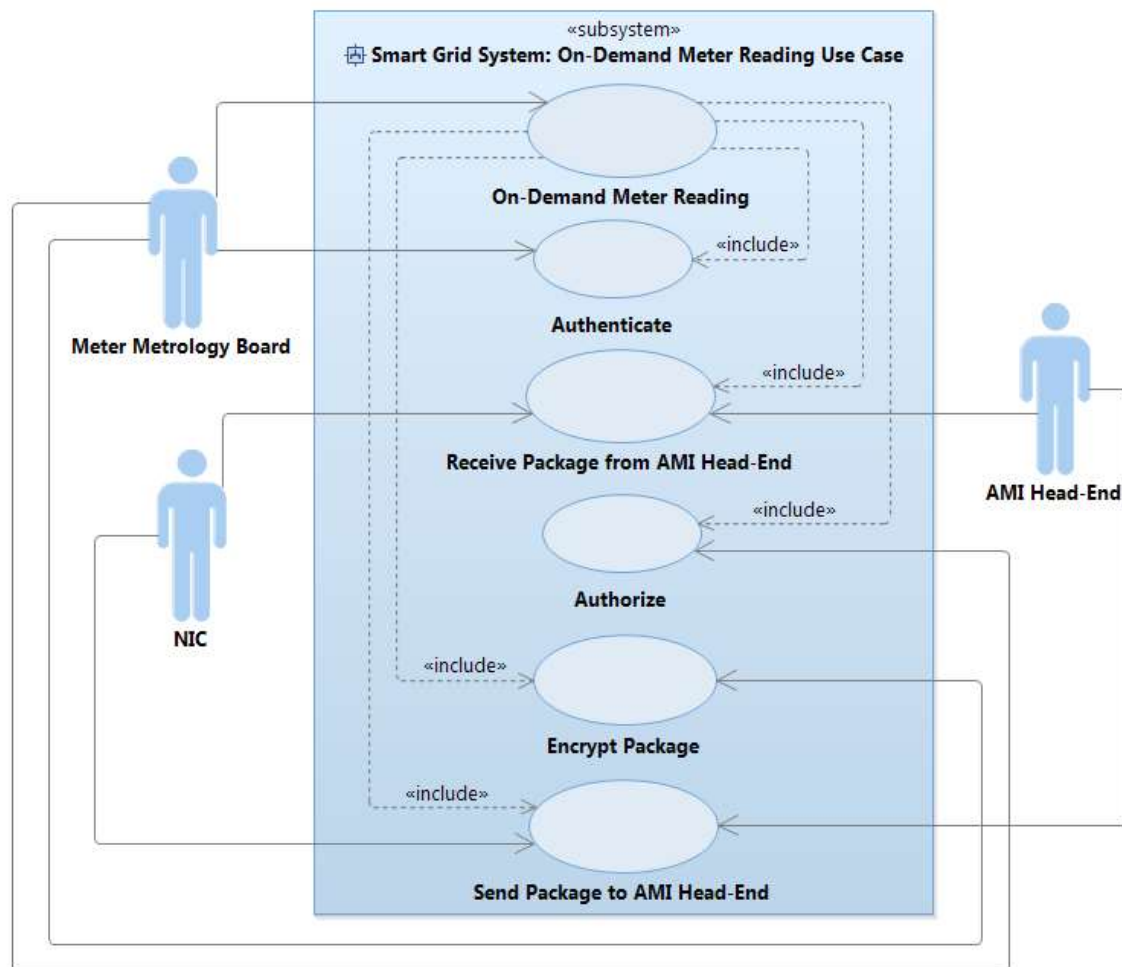


Figure 35 On-Demand Meter Reading Use case Diagram

Use case

Table 27 On-Demand Meter Reading Use case

On-Demand Meter Reading	
Use Case ID	UC_On-DemandMeterReading
Use Case Name	On-Demand Meter Reading
Description	The description of this use case is like this: After NIC receives the on-demand meter read request message from AMI head-end, meter metrology board retrieves meter read data. Then, NIC sends the on-demand meter read data to AMI head-end.
Precondition	AMI head-end has sent the on-demand meter read request message to smart meter.
Primary Actor	NIC

Secondary Actors	Meter Metrology Board, AMI head-end
Dependencies	INCLUDE USE CASE Authenticate INCLUDE USE CASE Receive Package from AMI Head-End INCLUDE USE CASE Authorize INCLUDE USE CASE Encrypt Package INCLUDE USE CASE Send Package to AMI Head-End
Basic Flow	Steps:
	1 Meter metrology board saves the session id received by AMI head-end. (UC_Authentication)
	2 NIC receives the encrypted on-demand meter read request message from AMI head-end. (UC_ReceivingPackageFromHead-End)
	3 Meter metrology board authorizes AMI head-end. (UC_Authorization)
	4 Meter metrology board retrieves meter read data in formatted table.
	5 Meter metrology board encrypts meter read data before sending to AMI head-end. (UC_EncryptPackage)
	6 NIC sends the encrypted on-demand meter read data to AMI head-end. (UC_SendingPackageToHead-End)
Post Condition AMI head-end receives the on-demand meter read data.	

Security Requirements of On-Demand Meter Reading Use Case

One of the security requirements for on-demand meter reading is **Confidentiality**. Meter read data should be confidential and not accessible to other customers. For example, when smart meter sends meter data to AMI head-end, it should be sent to the expected AMI head-end. Additionally, protecting customer information is also important. In on-demand meter reading, the meter read data will be sent to the customer at the end. It is important to reach data to the customer who demanded it [53]. One way to solve lack of confidentiality problem is **Encryption**. Additionally, using **Authenticated Sequence Numbers** is the other way to protect confidentiality [54].

The **Integrity** of meter read data is another important security requirement. Meter data should be without any modification when it is sent between different systems. For example, when smart meter sends meter data to AMI head-end, it is important that the data is not changed during transmission. Therefore, it is important to protect data against manipulation and deletion [53]. **Message Authentication Codes** is a security solution for integrity [54].

5.5 Security Related Uncertainties of Smart Grid

5.5.1 Examples of Smart Grid Uncertainties

Example 1: City Blackouts Uncertainties

One of the main examples of uncertainties in smart grid can be city blackouts or power outage where it creates instabilities in power supply [55]. These kinds of blackouts have been increased in the last decades due to demand's growth. A large percentage of population tends to use smart grid technologies, which leads to occurrence of transmission and distribution losses and subsequent problems such as economic losses [55]. Some factors could indicate that blackouts or power outage can be counted as an uncertainty problem. For example, lack of knowledge about technical issues in this case is an indicator of uncertainty.

One reason that leads to city blackouts, which is security related uncertainty is malicious shutdown commands. Attackers can send these commands to thousands of smart meters at the same time. It means smart meters will be shut down by receiving shut down commands and it causes the black out for the city. There could be some mutation operators, which introduce the malicious packages into the AMI system. When the smart meter receives these malicious packages, they are in fact the remote disconnect messages that disconnect the smart meter remotely and that leads to blackout. There can be different kinds of attacks such as structural attacks or cyber/internet attacks [55].

Attackers can execute malicious shut down commands. Each attacker has a property, which is called attack action. The attack action describes the activity of attacker, which is performed on a system. For example, in city blackout case, the attacker performs terminate action, which leads to shutting down the smart grid. The attacker needs to get into the AMI head-end system in order to capture the remote connect package in remote meter connect case. If the attacker can capture remote meter connect message, it can change this message to remote meter disconnect and send remote meter disconnect message to the smart meter and cause power outage or blackouts. We assume that attacker can get in to the AMI system and capture the remote meter connect message by Eavesdropping. This is another attack action of attacker. By eavesdropping, the attacker can do passive monitoring of data stream to obtain the information [56]. Now, attacker listens to the network communication and tries to capture the remote meter connect message sent from AMI head-end to smart meter. If the attacker gets remote meter connect message sent from AMI head-end, it can change this message to remote meter disconnect message by performing modify attack action. Then the attacker will send the remote meter disconnect message to the smart meter and perform the terminate action in order to shut down the smart grid, which leads to city blackouts. By sending these malicious remote meter disconnect

commands or malicious shut down commands via attackers to smart meters, attackers can cause harms or damages to the smart grid. For example, forwarding malicious shut down commands can have effect on integrity of the smart grid. Because, the attacker changes the command from remote meter connect to remote meter disconnect. It can also have effects on availability. Because, by sending shut down commands, smart meter will be disconnected remotely and the smart grid will not be available.

City blackouts can have major impacts on people's lives, economic conditions and the whole society. Some examples of these effects can be classified as below:

People can be affected for example by using transportation services such as train. Trains need electricity power to work. Therefore, by electricity cut-off, many passengers would be in trouble. The other example can be even more critical related to people's lives. Patients who are hospitalized in hospitals could suffer due to lack of electrical facilities at that time [55].

In case of economic effects many organizations, which are mostly dependent to use of smart grid and electricity can challenge with financial problems. They cannot produce their products and make profits.

Example 2: Demand and Supply Uncertainty

Another example of uncertainty could be uncertainty related to demand and supply. Due to population growth and tending people to use smart grids, the electricity demand is expected to increase drastically. However, it is not predictable the amount of electrical resources to be demanded by consumers especially in a long-term perspective of time. Since the demand percentage is not predictable, it means the electricity providers cannot make strong decisions about supplying the resources. This unpredictability causes uncertainty. Therefore, we can count demand and supply as a problem of uncertainty.

In short-term perspective, it is rather predictable to determine how much electricity will be needed. However, on the other hand, the peak load could happen unexpectedly and make the situation uncertain.

One reason that can cause demand and supply as an uncertainty problem is attacks on the integrity of AMI system or smart grids. It can lead to uncertainty in the optimization of energy distribution, in the real-time pricing, and in the load balancing of demand and supply. The rapid changes in demand can be caused by attacks [55]. Attackers can break the integrity of smart grid and the balance of power supply by consecutive turning on and turning off the smart meter. It relates to integrity since the unauthorized entity has illegal access to the system to modify the information and commands. The state of smart meter is changing overtime by receiving turn on and turn of commands from malicious

parties or attackers. That makes a big problem for real-time calculation in the system. Because the attacker manages to interfere with the real-time calculation or real-time control of load balancing in the smart grid. Therefore, demand and supply will be uncertain. One of the main aspects of the smart grid is the way you can control the real-time control and optimization of energy distribution.

There should be proper capacity management to prevent occurrence of such events. If the demand is exceeded the capacity limit, it should be controlled properly and avoided [55].

Example 3: Communication Uncertainty

One of the uncertainties in smart grid, which is related to two-way communication between AMI head-end and smart meter is communication uncertainties. There could be problems in the way AMI head-end and smart meter communicate with each other, for example in transmitting data or package. This could make communication uncertain. One example can be when AMI head-end sends a package or request to smart meter and waits for getting response from smart meter. In on-demand meter reading case, AMI head-end sends an on-demand meter read request message to smart meter. Based on this command, smart meter should retrieve meter read data and send the meter read data to AMI head-end. If the response time from smart meter to AMI head-end takes a long time, it means there is a communication problem between smart meter and AMI head-end. Smart meter can be suspicious and this can make the communication uncertain. The reason for late response time could be that there is an attacker or a malicious party that can create many connection requests to connect to AMI head-end. The reason for creating many fake requests is that in this case all the resources of AMI head-end will need to spent time for those fake requests. For instance, it will ask the AMI head-end to create thousands of smart meter controllers at the same time. Therefore, it will make the AMI head-end difficult to handle the real requests from the real smart meter. It can be called DOS attack or the denial of service attack. That is some kind of uncertainty. Other reason can be there is not sufficient bandwidth to make communication easier and faster. The other reason that leads to the long response time from smart meter to AMI head-end is due to uncertain security process at the smart meter. The example is too long authentication, authorization or decryption process.

This example of uncertainty is based on use case diagrams and their specifications. It can include use cases such as on-demand meter reading, remote meter connect/disconnect or smart meter authentication.

It is necessary to design communication technologies in the way that provide fast response time, better security and high bandwidth to extend the power of retrieving [57].

Example 4: Packet Loss Uncertainty

Packet loss can be an uncertainty issue in smart grids in package transmission. The packet can be lost because of different factors. Packet congestion can be a reason for packet loss. When there is a huge mass of data to be distributed, the packet can be lost [58].

The other reason for packet loss can be that hackers can access to the data due to incorrect access control and make some data breach. Packet loss can be counted as incomplete information. When the packet is lost, the demand will be increased. For example, when AMI head-end sends on-demand meter reading request message to smart meter, if there is a packet loss of meter read data, the demand for getting meter read data would be increased.

This uncertainty problem can be related to many use cases such as periodic meter reading, remote meter connect disconnect, and on-demand meter reading. There is a packet exchanged in these use case between AMI head-end and smart meter.

Example 5: Verifying Package's Digital Signature Uncertainty

One of the uncertainties related to smart grid is uncertainty in verifying digital signatures when there is a transmission between utilities. For example, when smart meter sends a sensitive message to AMI head-end, smart meter needs to encrypt the message for security purposes. Then when AMI head-end receives the package from smart meter, it needs to decrypt the package. However, before decrypting the package, AMI head-end should verify the digital signature of the package to make sure there is no integrity breach of data. If AMI head-end can verify the digital signature of the package, it can decrypt the package and reads the content inside the package. However, sometimes there are cases that make verifying digital signature scheme difficult or there are some errors or uncertainties on the way. One of the examples is related to hash functions or algorithms. MD5 hash function is an example. It is used as an integrity checking for the file in digital signature schemes [59]. It has been shown that MD5 hash function is no longer secure in checking file's integrity and verifying digital signature of the package. It makes uncertainty problem. The reason is that MD5 hash function can create the same hashes for two different packages with identical names. This is called colliding messages. The same hash sums means that two different packages will have the same hash values when checking. It means the integrity of data cannot be checked. This can be a case for attackers. For example, when smart meter sends a secure package to AMI head-end, attackers can change the package content with keeping the same name that produce the same hash value that it was expected to be produced. Therefore, there is uncertainty in the way. It cannot be clarified if the package received is the one that was expected to be received or if it is the package containing malicious contents.

Example 6: Signing Package Uncertainty

Another example related to uncertainty is package-signing uncertainty in encryption scheme. For example, in AMI system case, when smart meter sends a sensitive package to AMI head-end, it should encrypt the package before sending it to AMI head-end. After encryption, smart meter should sign the package. If the package can be signed successfully, the package will be sent to AMI head-end. However, some cases can make problems or errors in the way of signing package. The example can be attacker or any malicious party that can make uncertainties in signing package. For example, the attacker can make a pair of packages with equal MD5 sums, one containing original files another one with malicious contents. Then the attacker sends the original package for signing. This package is signed successfully by using MD5 hash function. However, attacker decides to replace the secure package with flawed package, which both of them are equal in MD5 sum and digital signature. Since, they are equal, it takes a long time to discover the attack [59].

Example 7: Uncertainty Related to Periodic Meter Reading

One of the uncertainty problems in smart grid can be uncertainty related to periodic meter reading case. For example, in periodic meter reading, the smart meter records the consumer's electrical usage data called meter read data. It records this information in frequent intervals for example every 15 minutes. Then at the end, this data will be sent to customer for billing purposes. Since meter read data is very sensitive data, its privacy is very important and it should be considered carefully. It should not be published publicly. Only the authorized entities must have rights to access the meter read data. Due to meter read data's sensitivity and privacy issues, it is a good target for attackers to hack it for billing and account management purposes. An example of this kind of attack can be called electricity or power theft attack for financial purposes. Power theft is detected when there is a mismatch between the reported energy consumption and real energy consumption without any information manipulation [60]. The attackers can manipulate the meter readings for example to make financial profits. The other example can be that the attackers by breaching confidentiality of meter read data, can access to consumer's accounts and do some malicious actions such as account management.

5.5.2 Mutation Operators

Mutation operators are kinds of operators used in genetic algorithms. Genetic algorithms are one of the most popular evolutionary algorithms used for optimization techniques. The idea of mutation operator is to replace a gene with another value. There are two important factors for applying mutation operators: the first factor is the population of mutants meaning the number of mutants to be applied. The second factor is the strength of mutation operators meaning the disorder proportion produced in a chromosome [61].

We can test the software by introducing mutation analysis or mutation testing. The idea behind this scheme is to introduce mutants. Mutants are artificial faults or faulty programs seeded into the original program in order to detect the faults in the original program. By designing test cases, the defects can be revealed [62]. The roles of mutation operators are to do modifications such as to replace, insert, or delete variables and expressions [63].

We can define some mutation operators in the context of our thesis. In our case, these mutation operators can be used for introducing uncertainties into CPSs in general. For example, we can define one generic mutation operator to mutate a CPS element that it will send a false (fake) signal. Then we can apply the mutation operator to make the smart meter to send fake shutdown command. In order to define mutation operators in our thesis, we can generalize some of the examples of uncertainties, which are defined above. For example, some of the uncertainty problems in the AMI system relates to violation of integrity security requirement. For example, shut down commands and sending fake meter reading commands can be categorized into one group. Therefore, we can introduce one general mutation operator for both of these cases, which this mutation operator is defined to violate the integrity of information and introduce uncertainty in the AMI system.

6 Modeling

In this chapter, we provide the modeling and designing of three different types of UML diagrams, which are class, sequence and state chart diagrams. In section 6.1, we design two class diagrams. One of them is called AMI head-end class diagram and the other one is called smart meter class diagram. In section 6.2, we have different sequence diagrams. Some of them are related to main functionalities of AMI head-end. Some others are for security design of AMI system. We have also some sequence diagrams for security related uncertainties of AMI system. Finally, in section 6.3, we provide some state chart diagrams.

6.1 Class Diagram

6.1.1 Class Diagram Description

The class diagrams are designed based on use cases in the previous section. In order to design the class diagrams, use case diagrams are used for determining the classes, especially the name of the classes. The actors, which perform an operation in the use case diagrams, are usually classes in the class diagram. The methods in the class diagrams can be determined by looking at step's part of use case specifications. Attribute part of classes in the class diagram, can be filled simply based on the class name and what each class performs.

There are two different class diagrams below. One is a class diagram related to AMI head-end. The other one is a class diagram related to smart meter.

6.1.2 AMI Head-End Class Diagram

Summary

In the AMI Head-End class diagram, there are different classes, which are related to AMI head-end. One of the main classes in this class diagram is AMI head-end class. There are other classes such as smart meter controller, connection handler, server socket, thread, etc. Since there are many smart meters in the system, to handle these smart meters, there is a need to have a class that controls smart meters. Therefore, smart meter controller is used to handle the management of smart meters. Connection handler, which is part of smart meter controller is used to handle the connection between AMI head-end and smart meter. Server socket is used for accepting connection between AMI head-end and smart meter. Thread class is used for receiving packages from smart meter. "Is" used in the thread class stands for input stream. Data package class is for showing packages exchanged between

AMI head-end and smart meter. Each data package has a package code, which represents what kind of package is exchanged between AMI head-end and smart meter.

In this class diagram, some of the classes are used for showing the basic functionalities of AMI head-end system. However, besides these classes there are other classes, which are used to show the security functionalities of AMI head-end system. For example, there are some classes used for Authentication design. Some classes are used for Authorization part and some of them for encryption and decryption security design. We use “Security Patterns” for the security design [47].

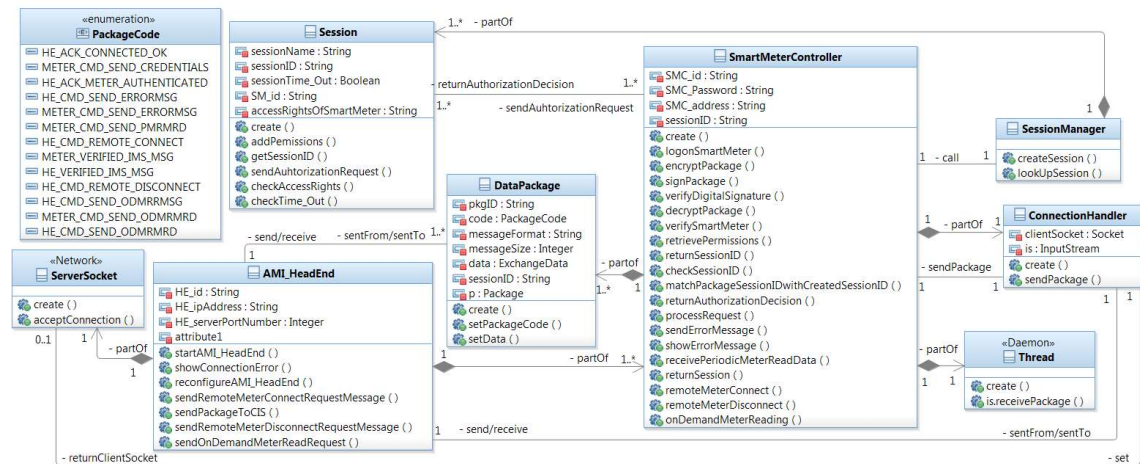


Figure 36 AMI Head-End Class Diagram with Security Functionalities

Documentation

Table 28 Class summary for AMI Head-End Class Diagram with Security Functionalities

Class Summary	
Class	Description
AMI_HeadEnd	This class is the main class in AMI head-end class diagram. The AMI head-end is as a back office that controls the advanced metering infrastructure.
SmartMeterController	Since there are many smart meters in the system, therefore to control these smart meters, there is a need to have smart meter controller. Smart meter controller is a class, which controls the smart meters. There is a composition relationship between smart meter controller and AMI head-end classes. It shows that smart meter controller is a part of AMI head-end. The multiplicity is 1 to many. Each AMI head-end has many smart meter controllers. Smart meter controller is also a class of security pattern design for Authentication. It is used as an authenticator to authenticate smart meter. Since smart meter is not part of AMI head-end class diagram as mentioned above, therefore smart

	meter controller will authenticate credentials of smart meter, which are some properties of smart meter. Smart meter controller can be used for Authorization and encryption/decryption security patterns, as well. In authorization process, smart meter controller acts as Policy Enforcement Point (PEP), which sends authorization request to session object. It will process the package after being authorized. In encryption/decryption process, smart meter controller encrypts and decrypts the package.
SessionManager	Session manager class is as a part of security design both in authentication and authorization processes. It is used to manage the session class. Smart meter controller will call the session manager to create session and look up for the session.
Session	Authentication, when smart meter controller verifies smart meter's credentials, if the verification process is successful, then smart meter controller will create session via session manager. In Authorization, session can act as PDP. It has access rights of smart meter, which are defined in XML file or data base. Session object receives authorization requests from PEP, which is smart meter controller here. Then, it will check the access rights of smart meter and return authorization decision.
Thread	Thread is a class used for receiving the packages, which are sent by smart meter to AMI head-end.
DataPackage	Data package class means the messages or data exchanged between AMI head-end and smart meter.
ConnectionHandler	Connection handler handles the connections between AMI head-end and smart meter. It is used for sending packages from AMI head-end to smart meter. It is as a part of encryption and decryption security design.
PackageCode	Package code is a code, which is defined in data package class as an attribute or property. When AMI head-end sends or receives packages from smart meter, each package should have a package code, which defines what type of package is transmitted.
ServerSocket	Socket is a class used for connecting AMI head-end and smart meter.

Class AMI_HeadEnd

Table 29 Attribute Summary for Class AMI_HeadEnd in AMI Head-End Class Diagram

Attribute Summary	
Attribute	Description
HE_id: String	ID stands for identification. It is a unique identity that distinguishes each device from another one. Almost each class has this attribute. Here, AMI head-end has an ID called HE_id, which distinguishes it from other AMI

	head-ends. Id's data type is String.
HE_ipAddress: String	Ip-Address is internet protocol address. Here, since AMI head-end is in a network (It has a communication with NIC and AMI network), then there should be an ip Address for AMI head-end. This address shows where AMI head-end is located in the network. Its data type is String.
HE_serverPortNumber:Integer	Since AMI head-end is in the server side and also it is in a network with NIC and AMI network, therefore it has a server port. Each port has a number, which is combination of digits. Therefore, AMI head-end has a server port number with data type of integer.

Table 30 Method Summary for Class AMI_HeadEnd in AMI Head-End Class Diagram

Method Summary	
Method	Description
startAMI_HeadEnd (in HE_serverPortNumber: Integer): void	This is a first step or method in Establish connection with smart meter. AMI head-end first needs to be started in order to perform other tasks. Parameter for this method is HE-serverPortNumber. The type of this parameter is integer. This parameter is as an input to this method. The input parameter is specified by using "in" keyword. The type of the return value is void.
showConnectionError ()	In establish connection, when server socket cannot be created successfully, AMI head-end shows connection error. It means the establish connection fails between AMI head-end and smart meter.
ReconfigureAMI_HeadEnd (in HE_serverPortNumber: Integer): void	In establish connection process, when there is a connection error between AMI head-end and smart meter, AMI head-end will be reconfigured with unused port number. This step will be done after AMI head-end shows the connection error.
sendRemoteMeterConnectRequestMessage (in HE_ipAddress: String): void	In remote meter connect, CIS sends remote meter connect request message to AMI head-end. The parameter is HE-ipAddress. Because the message will be sent to AMI head-end's address.
sendPackageToCIS (in p: Package, CIS_address: String): void	In remote meter connect/disconnect, AMI head-end sends the closed/opened internal meter switch verification message to CIS.
sendRemoteMeterDisconnectRequestMessage (in HE_ipAddress: String): void	In remote meter disconnect, CIS sends remote meter disconnect request message to AMI head-end.
sendOnDemandMeterReadRequest (in HE_ipAddress: String): void	In On-demand meter reading, CIS sends on-demand meter read request message to AMI head-end.

Table 31 Attribute Summary for Class SmartMeterController in AMI Head-End Class Diagram

Attribute Summary	
Attribute	Description
SMC_id: String	SMC_id is the identification of smart meter controller.
SMC_password: String	SMC_Password is the password of smart meter controller. Since there are many smart meter controllers, which control smart meter, each of them should have different passwords. Password data type is string.
SMC_address: String	SMC_Address is the address of smart meter controller. It means where smart meter controller is located in the network and in the system.

Table 32 Method Summary for Class SmartMeterController in AMI Head-End Class Diagram

Method Summary	
Method	Description
create (in s: ServerSocket): void	This is a method for creating smart meter controller by AMI head-end. The parameter for this method is s. The type of this parameter is server socket.
logonSmartMeter (in s: ServerSocket): void	After AMI head-end accepts the establish connection request sent by smart meter, and establishes a connection with smart meter, smart meter controller starts to logon to smart meter.
encryptPackage (in p: Package): void	Smart meter controller encrypts the package before sending it to smart meter for the security purposes.
signPackage (in p: Package): void	Smart meter controller signs the package after encrypting it and before sending it to smart meter.
verifyDigitalSignature (in p: Package): boolean	In decryption process, smart meter controller verifies the digital signature of the received package before decrypting and reading the content of the package. The reason for verifying digital signature is security purposes to make sure that the package is received from secure source.
decryptPackage (in p: Package): void	Smart meter controller decrypts the received package after verifying the digital signature in order to read the package's content.

<p>verifySmartMeter (in SM_id: String, in SM_password: String): boolean</p>	<p>When smart meter sends its credentials (its id and password) to AMI head-end, smart meter controller verifies the smart meter's credentials. The input parameters for this method are SM_id and SM_password (smart meter's id and password). The type of return value is boolean. It returns the result in "r". "R" can be either r=MeterAuthenticated or r=MeterNotAuthenticated.</p>
<p>retrievePermissions ()</p>	<p>In smart meter authentication process, when smart meter is authenticated by AMI head-end, smart meter controller retrieves the permissions of smart meter.</p>
<p>returnSessionID ()</p>	<p>In smart meter authentication, after adding permissions, session returns the session id to smart meter controller. Additionally, in authorization process, session object returns the session id of the session, which is created by AMI head-end to smart meter controller.</p>
<p>checkSessionID (in p.sessionID: String): void</p>	<p>In authorization process, when AMI head-end receives a package from smart meter, smart meter controller first has to check the session id of that package. The reason for this checking is to see if this package's session id matches with the session id that AMI head-end has already sent to smart meter in authentication process. If they match, it means this smart meter that sends the package, has already been authenticated by AMI head-end. The parameter is "p.sessionID". It is the package's session id.</p>
<p>matchPackageSessionIDwithCreatedSessionID (in p.sessionID: String, in sessionID: String): Boolean</p>	<p>After checking the received package's session id in authorization process, smart meter controller will match this package's session id with the session id that AMI head-end has already sent to smart meter. This method returns a Boolean value. If there is a match between session ids, it means the smart meter, which sends the package to AMI head-end, has already been authenticated by AMI head-end. Therefore, this package will be checked to see if it can be authorized or not.</p>
<p>returnAuthorizationDecision (in p: Package): Boolean</p>	<p>In authorization process, if there is a match between received session id and created session id, smart meter controller sends authorization request to session object. Session object will check the access rights to see if the received package code is in the</p>

	list of access rights of smart meter. Then, session will return authorization decision to smart meter controller. Authorization decision is a Boolean value. It can be either the received package, which is sent by smart meter is authorized or it is not authorized.
processRequest (in p:Package): Boolean	If the package is authorized, then smart meter controller can process the received package by process request method.
sendErrorMessage ()	In authorization process, if the package is not authorized, session sends error message to smart meter controller.
showErrorMessage ()	Smart meter controller shows the error message if the package is not authorized.
receivePeriodicMeterReadData ()	This method is as a trigger in periodic meter reading. It is called from AMI head-end to smart meter controller.
returnSession ()	Session manager will return the session to smart meter controller after look up session method in checking the session state in periodic meter reading, remote meter connect/disconnect and in on-demand meter reading processes.
remoteMeterConnect (in SM_id: String, in SMC_address: String): void	AMI head-end calls the method remote meter connect of smart meter controller as a trigger in remote meter connect.
remoteMeterDisconnect (in SM_id: String, in SMC_address: String): void	AMI head-end calls the method remote meter disconnect of smart meter controller as a trigger in remote meter disconnect.
onDemandMeterReading (in SM_id: String, in SMC_address: String): void	AMI head-end calls on-demand meter reading method of smart meter controller. This method is a trigger in on-demand meter reading process.

Class Session Manager

Table 33 Method Summary for Class SessionManager in AMI Head-End Class Diagram

Method Summary	
Method	Description
createSession ()	Smart meter controller calls the session manager to create the session.
lookUpSession ()	This method is used to determine the state of session when AMI head-end wants to send or receive package from smart meter. Smart meter controller calls the session manger to perform look up session method.

Class Session

Table 34 Attribute Summary for Class Session in AMI Head-End Class Diagram

Attribute Summary	
Attribute	Description
sessionName: String	It is the name of session. For example the session can be logon session or it can be logoff session. The data type is string.
sessionID: String	It is the identification of session. The data type is String.
sessionTime_Out: Boolean	This property shows if the session times out or not. The data type is Boolean. Because it has two values.
SM_id: String	Session should also have some information related to the subject. One of this information is smart meter id.
accessRightsofSmartMeter: String	The other information related to smart meter is access rights of smart meter.

Table 35 Method Summary for Class Session in AMI Head-End Class Diagram

Method Summary	
Method	Description
create ()	This is a method for creating session by Smart meter controller.
addPermissions ()	Smart meter controller adds the permissions of smart meter to the session object.

getSessionID (out sessionID: String): void	In Authorization process, when session manager looks up for the session, it calls the get session id method of session to get the session id of the created session by AMI head-end. By this method, session id will be obtained and then it will be returned to smart meter controller to be matched with the received package's session id.
sendAuthorizationRequest ()	In authorization process, if there is a match between received package's session id and created session id, smart meter controller sends the authorization request to session.
checkAccessRights (in SM_id: String, in accessRightsOfSmartMeter: String): void	Session after receives authorization request, checks the access rights of smart meter and compares it with the received package code. If the received package code is in the list of rights of smart meter, it means this package is allowed to be sent from smart meter to AMI head-end.
checkTime_Out (in sessionTime_Out: Boolean): boolean	Session will check the session timeout. If the session is timed out, it means the session is inactive.

Class Thread

Table 36 Method Summary for Class Thread in AMI Head-End Class Diagram

Method Summary	
Method	Description
create ()	Smart meter controller creates the thread class.
is.receivePackage ()	The thread class receives the packages sent by smart meter. "is" stands for input stream.

Class DataPackage

Table 37 Attribute Summary for Class DataPackage in AMI Head-End Class Diagram

Attribute Summary	
Attribute	Description
pkgID: String	pkgID is the identification of data package. It means which package it is. The data type is string.
code: PackageCode	Code refers to enumeration class called "PackageCode". In this class, there are different kinds of codes with their names. The example is HE_CMD_REMOTE_DISCONNECT. This code means AMI head-end will perform command (CMD) remote disconnect.
messageFormat: String	Message format shows the format of the particular message. For example,

	the format for meter read data could be table data.
messageSize: Integer	Message size shows the size of message. The units for size can be like kilo bytes, or megabytes and etc. the data type is integer. Because the size is numerical.
data: ExchangeData	It shows the data, which is going to be exchanged between AMI head-end and smart meter.
sessionID: String	During receiving or sending messages, session id is also attached to that message. Therefore we need to have this session id in this class.
p: Package	“P” stands for package. It is a property of class data package. Its data type is

Table 38 Method Summary for Class DataPackage in AMI Head-End Class Diagram

Method Summary	
Method	Description
create ()	Smart meter controller creates the class data package.
setPackageCode (in code:PackageCode): void	Data package sets package code for transmitting the packages.
setData (out sessionID: String): void	In smart meter authentication when AMI head-end wants to send acknowledgment message to smart meter to show that it has authenticated the smart meter, it sends session id to smart meter as well. Therefore, before sending the session id, the data package sets the session id as a data to be sent to smart meter. Here, session id is as an out parameter. Because it will be sent to smart meter.

Class ConnectionHandler

Table 39 Attribute Summary for Class ConnectionHandler in AMI Head-End Class Diagram

Attribute Summary	
Attribute	Description
clientSocket: Socket	When AMI head-end receives a package from smart meter, connection handler needs to connect to client socket, which is smart meter socket in order to handle this receipt.
is: InputStream	When thread receives a package from Smart meter, it receives the package as an input stream by listening to the client socket.

Table 40 Method Summary for Class ConnectionHandler in AMI Head-End Class Diagram

Method Summary	
Method	Description
create ()	Smart meter controller creates the connection handler with this method.

sendPackage (in p: Package, NIC_address: String): void	When smart meter controller sends package to smart meter, it sends the package through connection handler.
---	--

Class PackageCode

Table 41 Enumeration Summary for Class Package Code in AMI Head-End Class Diagram

Enumeration Summary	
Enumeration	Description
HE_ACK_CONNECTED_OK	AMI head-end acknowledges (ACK) that the connection between AMI head-end and smart meter is OK in establish connection process. AMI head-end sends reply to smart meter that it is connected to smart meter.
METER_CMD_SEND_CREDENTIALS	This package code means smart meter sends its credentials to AMI head-end in smart meter's authentication process.
HE_ACK_METER_AUTHENTICATED	In Smart meter's Authentication, AMI head-end sends acknowledgment to smart meter that shows the meter is authenticated by AMI head-end.
HE_CMD_SEND_ERRORMSG	When smart meter is not authenticated by AMI head-end, head-end sends error message to smart meter.
METER_CMD_SEND_ERRORMSG	In encryption and decryption process when smart meter cannot sign the package or verify the digital signature of the package, smart meter sends error message to AMI head-end.
METER_CMD_SEND_PMRMRD	In periodic meter reading, smart meter sends periodic meter read data to AMI head-end.
HE_CMD_REMOTE_CONNECT	In remote meter connect, AMI head-end sends remote meter connect message to smart meter.
METER_VERIFIED_IMS_MSG	In remote meter connect/disconnect, Smart meter sends closed/opened internal meter switch verification message to AMI head-end.
HE_VERIFIED_IMS_MSG	In remote meter connect/disconnect, AMI head-end sends closed/opened internal meter switch verification message to CIS.
HE_CMD_REMOTE_DISCONNECT	In remote meter disconnect, AMI head-end sends remote meter disconnect message to smart meter.
HE_CMD_SEND_ODMRRMSG	In on-demand meter reading, AMI head-end sends on-demand meter reading request message to smart meter.
METER_CMD_SEND_ODMRMRD	In on-demand meter reading, Smart meter sends on demand meter read data to AMI head-end.
HE_CMD_SEND_ODMRMRD	In on-demand meter reading, AMI Head-end sends on demand meter read data to CIS.

Table 42 Method Summary for Class ServerSocket in AMI Head-End Class Diagram

Method Summary	
Method	Description
create ()	Server socket is created by AMI head-end in order to have connection with smart meter.
acceptConnection ()	When smart meter requests to establish connection with AMI head-end, AMI head-end will accept this request. Therefore, server socket is responsible for performing accept connection method.

6.1.3 Smart Meter Class Diagram

Summary

There are some classes in smart meter class diagram. The most important classes in this diagram are classes, which are part of smart meter such as meter metrology board, NIC, internal meter switch. There are other classes as well. Meter metrology board is part of smart meter that is mostly responsible for processing messages received by AMI head-end. NIC plays the role of connection handler in AMI head-end class diagram. It is a network interface part of smart meter. NIC is responsible for handling packages sent from meter metrology board to AMI head-end. It is the mediator between meter metrology board and AMI head-end. Internal meter switch is another part of smart meter. It is used in remote meter connect/disconnect. There are some classes similar to the ones in AMI head-end class diagram such as data package, package code, and thread. The purpose of having these classes is the same as in the AMI head-end class diagram. In this class diagram, instead of server socket, there is another class named client socket. The reason is smart meter is in client side, not server side. Record service and record classes are used in periodic meter reading to record meter electrical usage data. Some of the classes in this class diagram are designed for basic functionalities of smart meter. Other classes are designed for representing security functionalities of smart meter system. These functionalities are Authentication, Authorization, Encryption, and Decryption. We use “Security Patterns” to design security functionalities.

Class Diagram

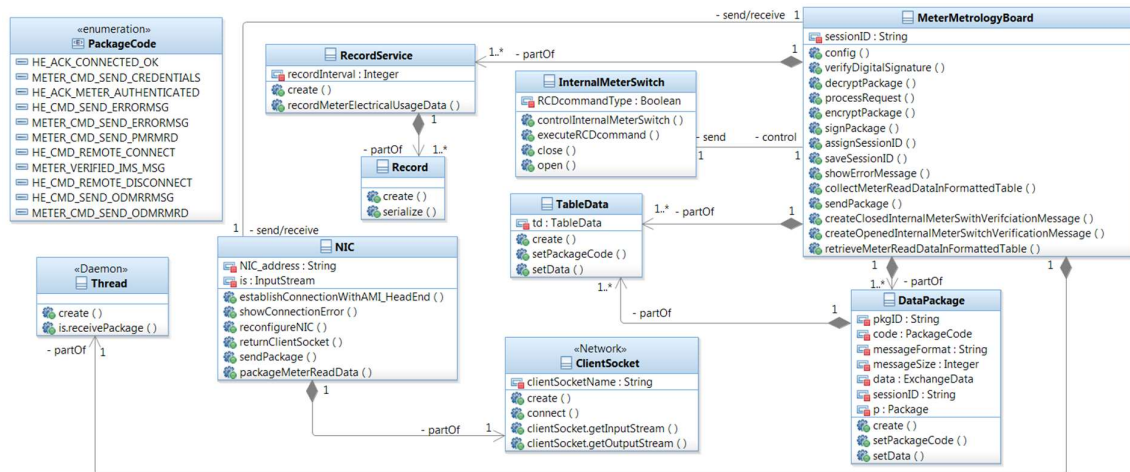


Figure 37 Smart Meter Class Diagram with Security Functionalities

Documentation

Table 43 Class Summary for Smart Meter Class Diagram with Security Functionalities

Class Summary	
Class	Description
MeterMetrologyBoard	Meter Metrology Board is a part of smart meter. It is responsible for recording meter read data, retrieving meter read data in formatted table and so on. It is also a class for Encryption and decryption security design. When smart meter wants to send a package to AMI head-end, meter metrology board is the component, which encrypts the package before sending. Additionally, in decryption process, meter metrology board will decrypt the received package from AMI head-end and will process the received package.
NIC	NIC is the other part of smart meter. It is as a transmitter between AMI head-end and Meter Metrology Board. It is also used in encryption and decryption process for sending package to AMI head-end.
DataPackage	Data package class means the messages or data exchanged between smart meter and AMI head-end.
Thread	There is a thread always running to receive package if any package sent from Head-end.
InternalMeterSwitch	Internal Meter Switch is the other part of smart meter. Its task is to execute RCD (Remote Connect Disconnect) command to close/open the meter switch in remote meter connect/disconnect.
ClientSocket	Socket is a class used for connecting AMI head-end and smart meter.
RecordService	This is a class used for recording the meter electrical usage data in periodic meter reading.
Record	This class is part of record service class to do serialize method.
TableData	This is a format for meter read data in periodic meter reading and on-demand meter reading.

PackageCode	Package code is a code, which is defined in data package class as an attribute or property. When smart meter sends or receives packages from AMI head-end, each package should have a package code, which defines what type of package is transmitted.
-------------	--

Class MeterMetrologyBoard

Table 44 Attribute Summary for Class MeterMetrologyBoard in Smart Meter Class Diagram

Attribute Summary	
Attribute	Description
sessionID: String	In smart meter authentication process, when AMI head-end sends session id to smart meter, meter metrology board saves this session id in the session id property of itself.

Table 45 Method Summary for Class MeterMetrologyBoard in Smart Meter Class Diagram

Method Summary	
Method	Description
config (in HE_ipAddress: String, in HE_serverPortNumber: Integer): void	It is the first method when smart meter tries to establish connection with AMI head-end. First meter metrology board performs some configuration. It needs to know head-end IP address and the head-end server port number.
verifyDigitalSignature (in p: Package): boolean	In decryption process, meter metrology board verifies the digital signature of the received package before decrypting and reading the content of the package. The reason for verifying digital signature is security purposes to make sure that the package is received from secure source.
decryptPackage ()	Meter metrology board decrypts the received package after verifying the digital signature in order to read the package's content.
processRequest ()	Meter metrology board will process the received package after decrypting the package.
encryptPackage (in p: Package): void	Meter metrology board encrypts the package before sending it to AMI head-end for the security purposes.
signPackage (in p: Package): void	Meter metrology board signs the package after encrypting it and before sending it to AMI head-end.
assignSessionID (in sessionID: String= "p.sessionID"): void	In smart meter authentication process, MMB assigns the p.sessionID, which is received from the package to the property of session id of MMB class.

saveSessionID (in sessionID: String= “p.sessionID”): void	Meter metrology board saves the session id, which AMI head-end has sent in smart meter authentication process.
showErrorMessage ()	In Authentication process when head-end cannot authenticate smart meter, MMB shows error message.
collectMeterReadDataInFormattedTable ()	In periodic meter reading process, Meter Metrology Board collects the meter read data in formatted table after the recording meter electrical usage data has been done.
sendPackage ()	In remote meter connect/disconnect, internal meter switch will send an acknowledgment of RCD (remote connect disconnect) executed message to meter metrology board. It shows that internal meter switch has executed the RCD command.
createClosedInternalMeterSwitch VerificationMessage ()	Meter Metrology Board creates closed Internal Meter Switch verification message to verify that the smart meter is connected in remote meter connect.
createOpenedInternalMeterSwitch VerificationMessage ()	Meter Metrology Board creates opened Internal Meter Switch verification message to verify that the smart meter is disconnected in remote meter disconnect.
retrieveMeterReadDataInFormattedTable ()	Meter Metrology Board retrieves meter read data in formatted table in on-demand meter reading.

Class NIC

Table 46 Attribute Summary for Class NIC in Smart Meter Class Diagram

Attribute Summary	
Attribute	Description
NIC_address: String	It is the address of NIC.
is: InputStream	When thread receives a package from AMI head-end, it receives the package as an input stream by listening to the server socket.

Table 47 Method Summary for Class NIC in Smart Meter Class Diagram

Method Summary	
Method	Description
establishConnectionWithAMI_HeadEnd (in HE_ipAddress: String): void	MMB calls the establish connection with AMI head-end method to NIC.
showConnectionError ()	If the establish connection is failed between smart meter and AMI head-end, NIC will show connection error.

reconfigureNIC ()	In establish connection process, when there is a connection error between smart meter and AMI head-end, NIC will be reconfigured. This step will be done after NIC shows the connection error.
returnClientSocket ()	If the establish connection is ok between smart meter and AMI head-end, client socket will be returned to NIC.
sendPackage (in p: Package, in HE_ipAddress: String): void	Meter metrology board calls the method send package of NIC to send a package to AMI head-end via NIC.
packageMeterReadData (in p: Package): void	Before sending meter read data to AMI head-end NIC first packages the meter read data.

Class DataPackage

Table 48 Attribute Summary for Class DataPackage in Smart Meter Class Diagram

Attribute Summary	
Attribute	Description
pkgID: String	Package ID is the identification of Message. Its data type is string.
code: PackageCode	Code refers to enumeration class called “PackageCode”. In this class, there are different kind of codes with their names. The example is METER_CMD_SEND_CREDENTIALS. It means smart meter sends its credentials to AMI head-end.
messageFormat: String	Message format shows the format of the particular message. For example, the format for meter read data could be table data.
messageSize: Integer	This parameter shows the size of message. The units for size can be kilo bytes, or megabytes and etc. the data type is integer. Because the size is numerical.
data: ExchangeData	It shows the data, which is exchanged between smart meter and AMI head-end.
sessionID: String	Since when receiving or sending some messages, session id is also attached to that message, therefore we need to have this session id in this class.
p: Package	“P” stands for package. It is a property of data package. Its data type is package.

Table 49 Method Summary for Class DataPackage in Smart Meter Class Diagram

Method Summary	
Method	Description
create ()	Meter metrology board creates the class data package.

setPackageCode (in code: PackageCode): void	Data package sets package code for transmitting the packages.
setData ()	In authentication process, when smart meter wants to send its credentials to AMI head-end, it sets the data, which here the data is the credentials of smart meter. Additionally in periodic meter reading and on-demand meter reading, smart meter sets data to meter read data, which is in format of table data.

Class Thread

Table 50 Method Summary for Class Thread in Smart Meter Class Diagram

Method Summary	
Method	Description
create ()	Meter metrology board creates the thread class to receive packages from AMI head-end.
is.receivePackage ()	Thread receives messages from AMI head-end. “Is” stands for input stream.

Class InternalMeterSwitch

Table 51 Attribute Summary for Class InternalMeterSwitch in Smart Meter Class Diagram

Attribute Summary	
Attribute	Description
RCDcommandType: Boolean	RCD command has two types. It can be remote connect command or remote disconnect command.

Table 52 Method Summary for Class InternalMeterSwitch in Smart Meter Class Diagram

Method Summary	
Method	Description
controlInternalMeterSwitch ()	MMB controls the internal meter switch.
executeRCDcommand ()	Internal meter switch executes the RCD command.
close ()	Internal meter switch closes meter switch if there is a connection.
open ()	Internal meter switch opens the meter switch if there is a disconnection.

Class ClientSocket

Table 53 Attribute Summary for Class ClientSocket in Smart Meter Class Diagram

Attribute Summary

Attribute	Description
clientSocketName: String	It is the name of client socket.

Table 54 Method Summary for Class ClientSocket in Smart Meter Class Diagram

Method Summary	
Method	Description
create ()	NIC creates the client socket class.
connect (in HE_ipAddress: String, In HE_serverPortNumber: Integer): boolean	This method tries to connect smart meter to head-end. Client socket for connection needs to know head-end IP address and head-end sever port number.
clientSocket.getInputStream ()	This method is for listening to the socket and gets input stream or packages from head-end.
clientSocket.getOutputStream ()	This method is for listening to the socket and gets output stream or packages.

Class RecordService

Table 55 Attribute Summary for Class RecordService in Smart Meter Class Diagram

Attribute Summary	
Attribute	Description
recordInterval: Integer	This shows the interval for recording meter read data. Here, it is 15 minutes.

Table 56 Method Summary for Class RecordService in Smart Meter Class Diagram

Method Summary	
Method	Description
create (out t ₁ : Time= “15min”): int	MMB creates the record service class for recording meter electrical usage data in periodic meter reading. Its parameter is t ₁ of type Time and it is for 15 minutes. It shows every 15 minutes, record service will record the meter electrical usage data.
recordMeterElectricalUsageData ()	Record service records meter electrical usage data in periodic meter reading.

Class Record

Table 57 Method Summary for Class Record in Smart Meter Class Diagram

Method Summary

Method	Description
create ()	Record service creates record class.
serialize ()	Record class performs serialize method.

Class Table Data

Table 58 Attribute Summary for Class TableData in SmartMeter Class Diagram

Attribute Summary	
Attribute	Description
td: TableData	This parameter is used in periodic meter reading and on-demand meter reading. Table data is the format of meter read data.

Table 59 Method Summary for Class TableData in Smart Meter Class Diagram

Method Summary	
Method	Description
create ()	Data Package class creates Table Data class.
setPackageCode (in code: PackageCode): void	Table data class will set the package code in periodic meter reading and on-demand meter reading. This package code is set for sending meter read data.
setData (in td: TableData): void	Table Data class sets data in periodic meter reading and on-demand meter reading. The parameter for this method is “td”. The data type of this parameter is table data.

Class Package Code

Table 60 Enumeration Summary for Class PackageCode in Smart Meter Class Diagram

Enumeration Summary	
Enumeration	Description
HE_ACK_CONNECTED_OK	AMI head-end acknowledges (ACK) that the connection is OK. This command is performed in establish connection process when head-end sends reply to smart meter that it is connected to smart meter.
METER_CMD_SEND_CREDENTIALS	This package code means smart meter sends its credentials to AMI head-end.
HE_ACK_METER_AUTHENTICATED	Head-end sends acknowledgment to smart meter that shows the meter is authenticated by AMI head-end.
HE_CMD_SEND_ERRORMSG	In encryption and decryption process, when the package is not signed or digital signature is not verified, head-end

	sends error message to smart meter. Additionally, in authentication process, when smart meter is not authenticated by AMI head-end, AMI head-end sends error message to smart meter.
METER_CMD_SEND_ERRORMSG	In encryption and decryption process, when smart meter cannot sign the package or verify the digital signature, respectively, smart meter sends error message to AMI head-end.
METER_CMD_SEND_PMRMRD	Smart meter sends periodic meter read data to AMI head-end in periodic meter reading.
HE_CMD_REMOTE_CONNECT	AMI head-end sends remote meter connect message to smart meter.
METER_VERIFIED_IMS_MSG	Smart meter sends closed/opened internal meter switch verification message to AMI head-end
HE_CMD_REMOTE_DISCONNECT	AMI head-end sends remote meter disconnect message to smart meter.
HE_CMD_SEND_ODMRRMSG	AMI head-end sends on-demand meter reading request message to smart meter.
METER_CMD_SEND_ODMRMRD	Smart meter sends on demand meter read data to AMI head-end in on-demand meter reading.

6.2 Sequence Diagram

6.2.1 Package Encryption Sequence Diagram

Package Encryption Sequence Diagram (AMI Head-End Side)

Package Encryption sequence diagram from AMI head-end side is used when AMI head-end wants to send a package to smart meter. Therefore, AMI head-end needs to encrypt the package before sending to smart meter for the security reasons.

In each sequence diagram, there are some objects, which interact with each other through messages or method calls. The objects are shown in rectangles. The objects in this sequence diagram are smart meter controller, data package, and connection handler. There are some actions, which are performed in order in sequence diagram. These actions are linked to the steps of each use case in the use case specification. In this sequence diagram, first smart meter controller encrypts the package, which will be sent to smart meter using cryptographic algorithms. Then smart meter controller will sign the package using digital signature. If the package is signed successfully, smart meter controller will send

the package to smart meter. If the package cannot be signed, smart meter controller will send an error message to smart meter.

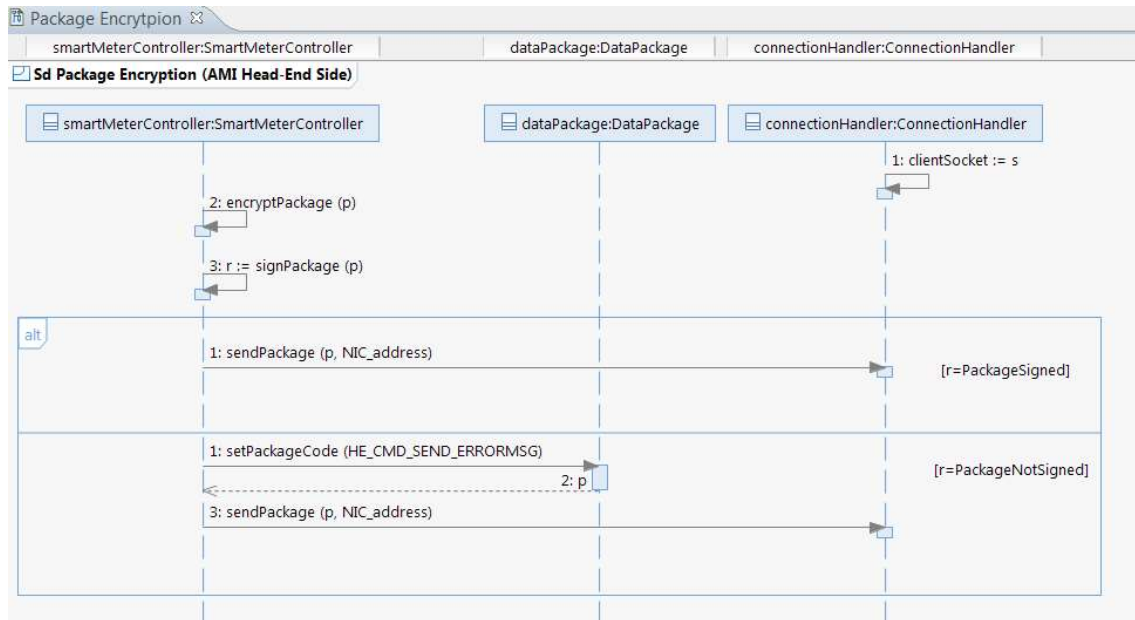


Figure 38 Package Encryption Sequence Diagram (AMI Head-End Side)

Package Encryption Sequence Diagram (Smart Meter Side)

Package Encryption sequence diagram from smart meter side is used when smart meter wants to send a package to AMI head-end. Therefore, smart meter should encrypt the package before sending to AMI head-end for the security purposes.

In this sequence diagram, first meter metrology board encrypts the package using cryptographic algorithms. Then, it will sign the package using digital signature. If the package is signed successfully, MMB will send the package to AMI head-end through NIC. If the package cannot be signed, MMB will send an error message to AMI head-end.

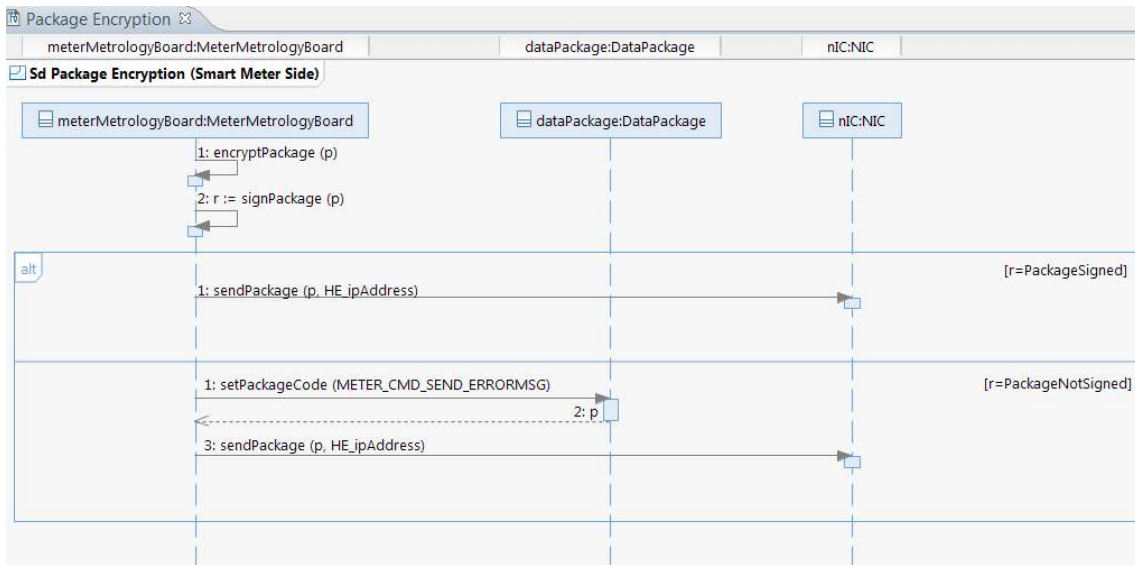


Figure 39 Package Encryption Sequence Diagram (Smart Meter Side)

6.2.2 Package Decryption Sequence Diagram

Package Decryption Sequence Diagram (AMI Head-End Side)

Package Decryption sequence diagram from AMI Head-end side is used when AMI head-end receives a package from smart meter. In this sequence diagram, when smart meter controller receives a package, it will first verify the digital signature of the package. The step of receiving package is done inside a loop. Because receiving package is a repetitive process, which requires listening to the network all the time. If smart meter controller can verify the digital signature, it will decrypt the package. Otherwise, if the verification process is failed, smart meter controller will send an error message to smart meter.

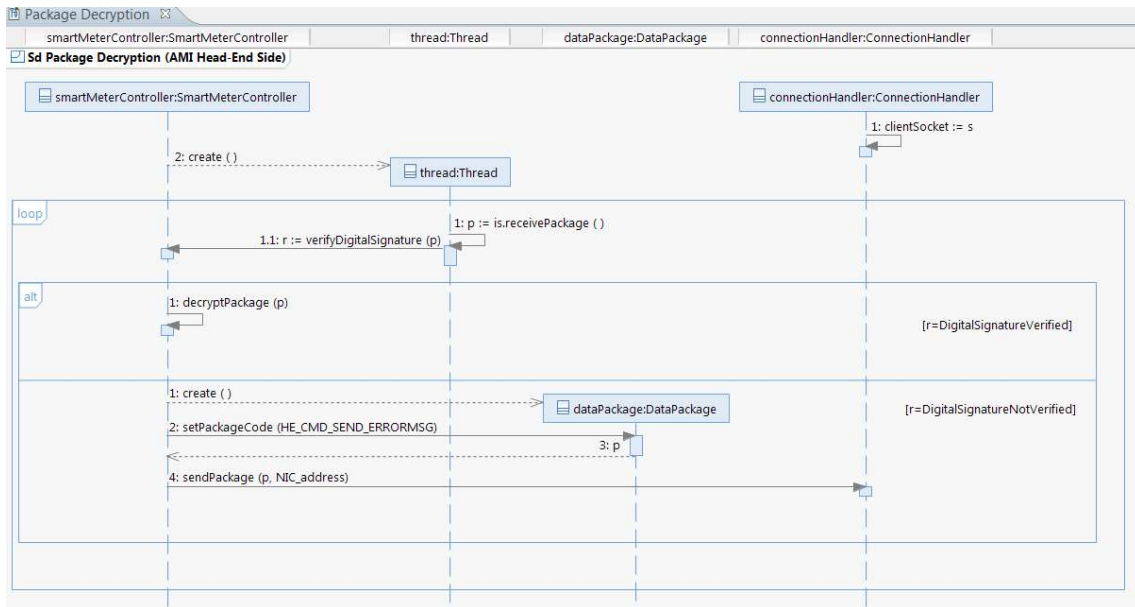


Figure 40 Package Decryption Sequence Diagram (AMI Head-End Side)

Package Decryption Sequence Diagram (Smart Meter Side)

Package Decryption sequence diagram from smart meter side is used when smart meter receives a package from AMI head-end. In this sequence diagram, when MMB receives a package, it will first verify the digital signature of the package. If it can verify the digital signature, it will decrypt the package and then it will process the package. Otherwise, if the verification process is failed, MMB will send an error message to AMI head-end through NIC.

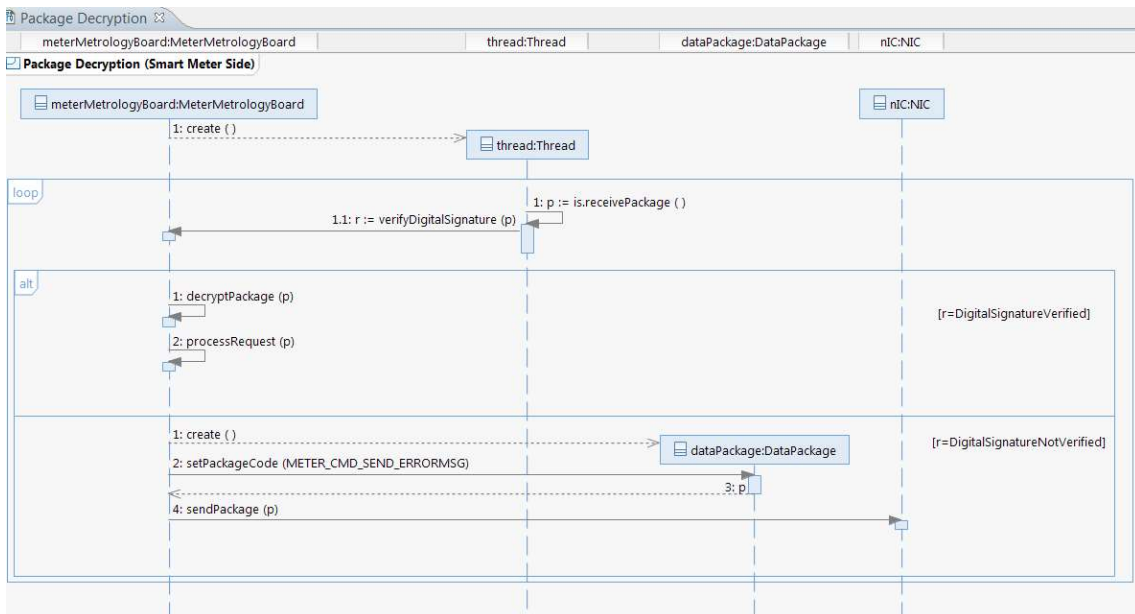


Figure 41 Package Decryption Sequence Diagram (Smart Meter Side)

6.2.3 Establish Connection Sequence Diagram

Establish Connection Sequence Diagram (AMI Head-End Side)

Establish connection is the first step in the smart meter registration process. The purpose of establish connection is to create a connection between AMI head-end and smart meter. In this sequence diagram, first AMI head-end starts. AMI head-end is responsible for responding the connection request sent by smart meter. Therefore, to accept the connection, AMI head-end should create a server socket, which listens to the messages sent by smart meter. Server socket can be created successfully or not successfully. Therefore, there is an alternative flow having two conditions showing that if server socket is created successfully or not. Creation of server socket depends on the server port number. For example, if the port number is already in use by another program, AMI head-end cannot create a server socket with that port number. If the server socket is created successfully, it accepts the connection. Then AMI head-end creates smart meter controller. The role of smart meter controller is since there are many smart meters in the system, smart meter controller can handle these smart meters. smart meter controller creates connection handler and data package. Connection handler handles the connection between AMI head-end and smart meter. Connection handler calls client socket to make connection with smart meter. Then smart meter controller calls “set the package code” method of data package. The package code here is “HE_ACK_CONNECTED_OK”. This code means the head-end acknowledges that there is a connection between head-end and smart meter. At the end, smart meter controller will send the package to NIC through connection handler. Here, for sending the package to smart meter there is a use of reference (ref) structure. This part is referenced to “Package Encryption (Smart Meter Side)” sequence diagram. It means to send a package from AMI head-end to smart meter, the package first should be encrypted based on the “Package Encryption (Smart Meter Side)” sequence diagram.

If the server socket cannot be created successfully, then AMI head-end will show an error message and AMI head-end will be reconfigured with an unused port number.

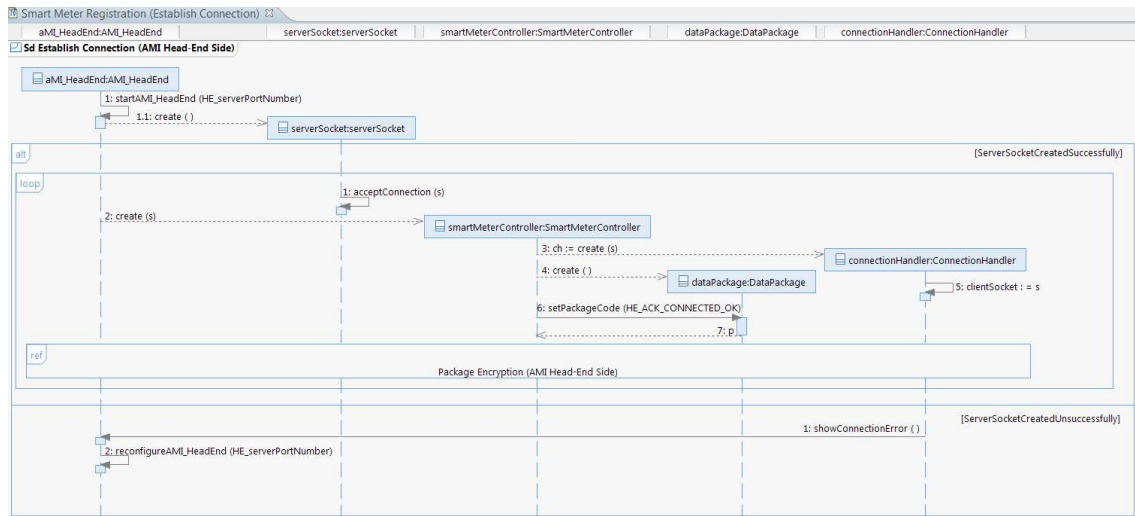


Figure 42 Smart Meter registration (Establish Connection) Sequence Diagram (AMI Head-End Side)

Establish Connection Sequence Diagram (Smart Meter Side)

The objects in this sequence diagram are meter metrology board, p1 type of Package, NIC, thread and client socket. In the process of establish connection, first meter metrology board performs some configuration. Then since smart meter needs to establish connection with AMI head-end, meter metrology board calls a method called “establishConnectionWithAMI_HeadEnd (HE_ipAddress)” to NIC. NIC creates client socket object. Client socket performs connect method. It listens to the network to see if there is any connection message reply from head-end side. Here, there is an alternative flow. In the alternative flow, if there is a connection error, NIC shows connection error and then NIC will be reconfigured. Otherwise, if the connection is ok, client socket returns the client socket to NIC. In NIC, there are two methods called “is: = clientSocket.getInputStream ()” and “os: = clientSocket.getOutputStream ()”. “Is” means input stream and “os” means output stream. NIC will create a thread object, which is responsible for receiving packages. This receiving package process from AMI head-end is performed by referring this part to “Package Decryption (Smart Meter Side)” sequence diagram. Because when smart meter receives a package from AMI head-end, it should first decrypt the package for the security reasons and then the received package will be processed by “processRequest (p)” method. These steps are performed inside the “Package Decryption (Smart Meter Side)” sequence diagram. If there is a connection between head-end and smart meter, smart meter will send its credentials such as its id and password to AMI head-end to be authenticated by AMI head-end.

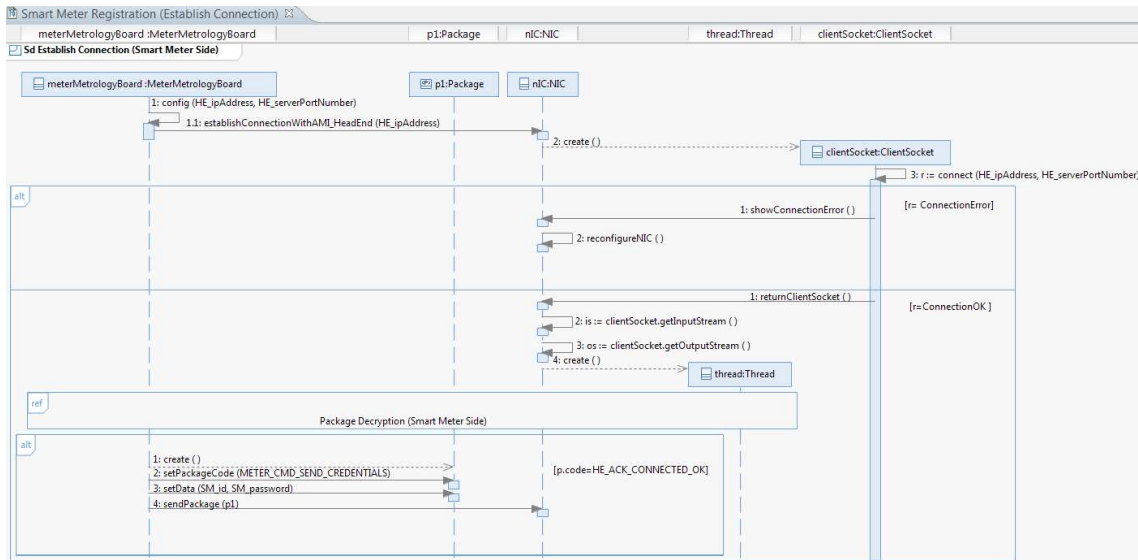


Figure 43 Smart Meter registration (Establish Connection) Sequence Diagram (Smart Meter Side)

6.2.4 Smart Meter Authentication Sequence Diagram

Smart Meter Authentication Sequence Diagram (Smart Meter Side)

In the smart meter authentication process, AMI head-end authenticates the smart meter. Authentication process is related to security design of the diagrams.

In this diagram, first smart meter receives a package from AMI head-end by re-using “Package Decryption (Smart Meter Side)” sequence diagram. The received package code is “HE_ACK_CONNECTED_OK”. After decryption, meter metrology board will send smart meter’s credentials (smart meter’s id and password) to AMI head-end. This sending process is performed by re-using “Package Encryption (Smart Meter Side)” sequence diagram. AMI head-end will verify smart meter’s credentials and will decide to either authenticate or not authenticate the smart meter. Therefore, smart meter will receive a package from AMI head-end. Meter metrology board processes the received package. If AMI head-end authenticates smart meter, then “p.code” is “HE_ACK_METER_AUTHENTICATED”. In this case, MMB assigns the p.sessionID, which is received from the package to the property of session id of MMB class. Then MMB will save the p.sessionID. On the other hand, if the authentication process is not successful, meter metrology board will show the error message.

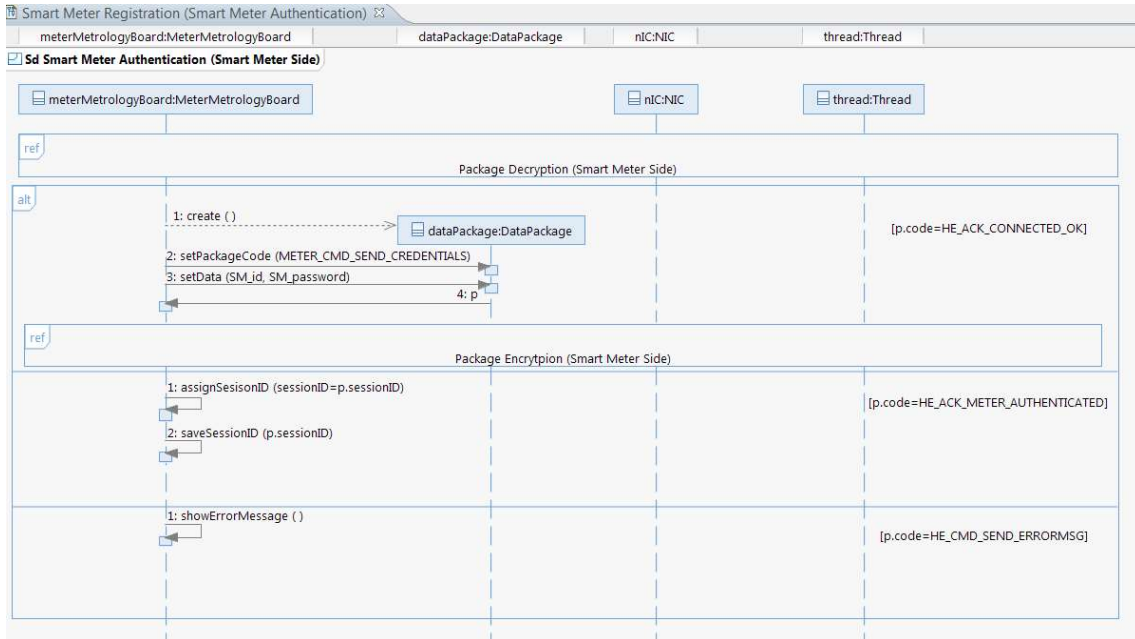


Figure 44 Smart Meter registration (Smart Meter Authentication) Sequence Diagram (Smart Meter Side)

Smart Meter Authentication Sequence Diagram (AMI Head-End Side)

The objects in this sequence diagram are smart meter controller, session manager, session, thread, data package and connection handler. Connection handler is responsible for creating connection between smart meter and AMI head-end. In this diagram, first AMI head-end receives a package from smart meter by referring this part to “Package Decryption (AMI Head-End Side)” sequence diagram. Based on the package code, there are different conditions. To show these conditions, we use an alternative flow. If the package code is “METER_CMD_SEND_CREDENTIALS” meaning smart meter sends its credentials to AMI head-end, smart meter controller will verify smart meter’s credentials (SM_id and SM_password). The result of verify smart meter method will be kept in a Boolean value called “r”. It has two conditions or results. If “r” equals to meter authenticated, smart meter controller will call the session manager to create session object. Then session manager creates the session object. Smart meter controller retrieves the permissions and adds permissions in the session object. Session object returns the session id to smart meter controller. Finally, smart meter controller will send the package including successful smart meter authentication code to smart meter by re-using “Package Encryption (AMI Head-End Side)” sequence diagram. If the result of verifying is “meter not authenticated”, smart meter controller will send the error message to smart meter.

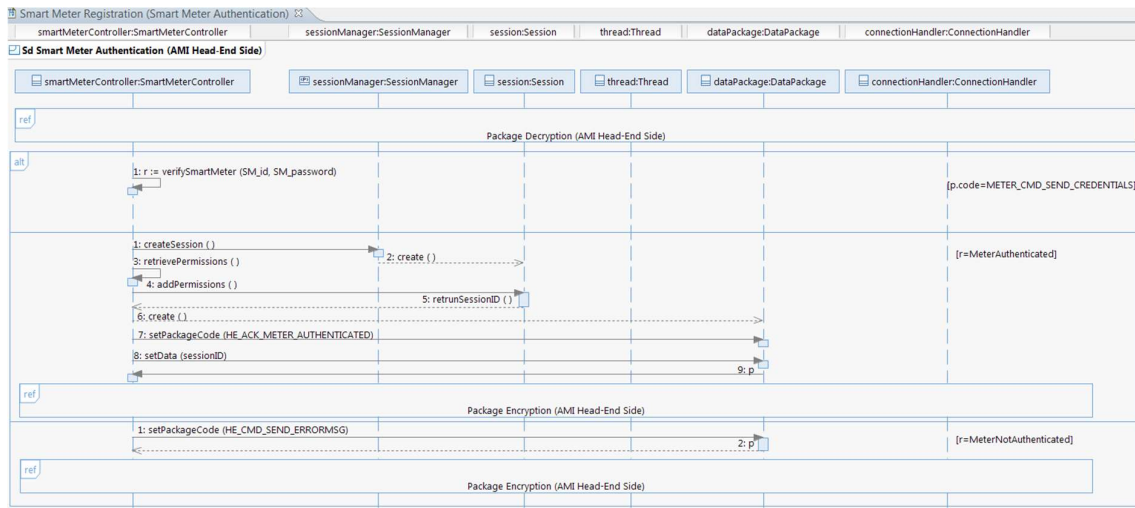


Figure 45 Smart Meter registration (Smart Meter Authentication) Sequence Diagram (AMI Head-End Side)

6.2.5 Authorization Sequence Diagram

Authorization Sequence Diagram (AMI Head-End Side)

In Authorization process, when AMI head-end receives a package from smart meter, it should check if the smart meter has sent the package that it was allowed to send, not malicious packages. In this sequence diagram, after AMI head-end receives a package from smart meter, if the smart meter has already been authenticated by AMI head-end, Smart meter controller checks the session id of the received package. Smart meter controller calls the session manager to look up for the session. Session manager calls the session to get the session id of the session, which AMI head-end has already created and sent to smart meter in authentication process. Session return the created session's session id. Then, smart meter controller matches the received package's session id with the created session's session id. If they match, it means this smart meter, which has sent the package to AMI head-end has already been authenticated by AMI head-end. Therefore, smart meter controller sends the authorization request to session. Session checks the access rights of smart meter to see if the received package code is in the list of access rights of smart meter. Then session returns the authorization decision to smart meter controller. Session acts as PDP and smart meter controller acts as PEP. If the authorization decision equals to package authorized, smart meter controller will process the received package by process request method. If the package is not authorized, session will send error message to smart meter controller. Smart meter controller shows the error message. On the other hand, if the received package's session id does not match with the created session's session id, session sends error message to smart meter controller. Smart meter controller will show the error message.

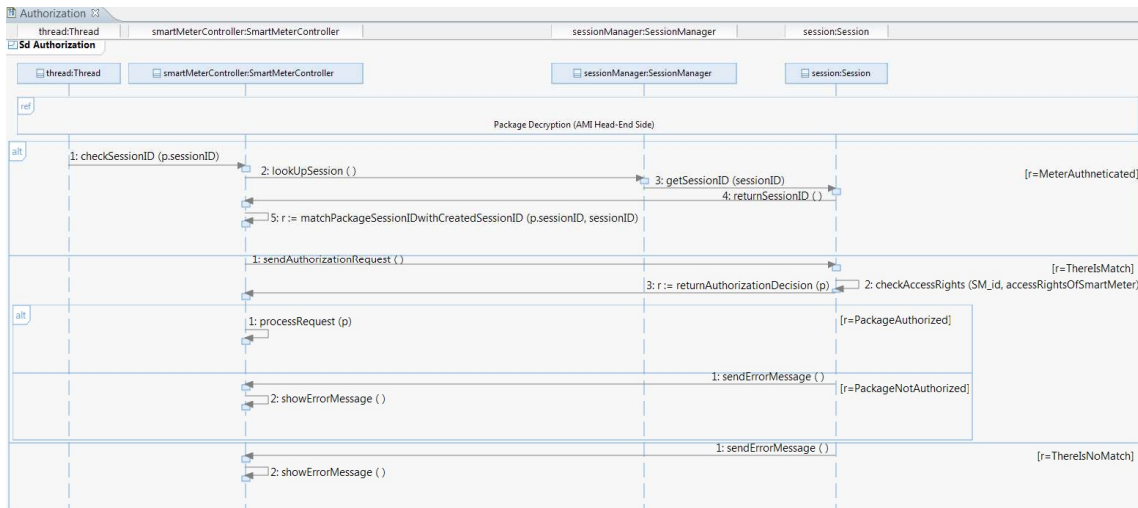


Figure 46 Authorization Sequence Diagram (AMI Head-End Side)

6.2.6 Periodic Meter Reading Sequence Diagram

Periodic Meter Reading Sequence Diagram (Smart Meter Side)

In this diagram, meter metrology board is responsible for recording meter electrical usage data and collecting meter read data in formatted table. For recording process, meter metrology board first creates a record service object. Record service performs the recording. Then record service creates a record object, which will perform serialize method. These actions are performed in the first loop structure. Then in the second loop structure after recording meter electrical usage data, meter metrology board collects meter read data in formatted table. It creates data package object. Data package creates table data object. At the end, NIC packages meter read data and it will send the meter read data to AMI head-end by “Package Encryption (Smart Meter Side)” sequence diagram.

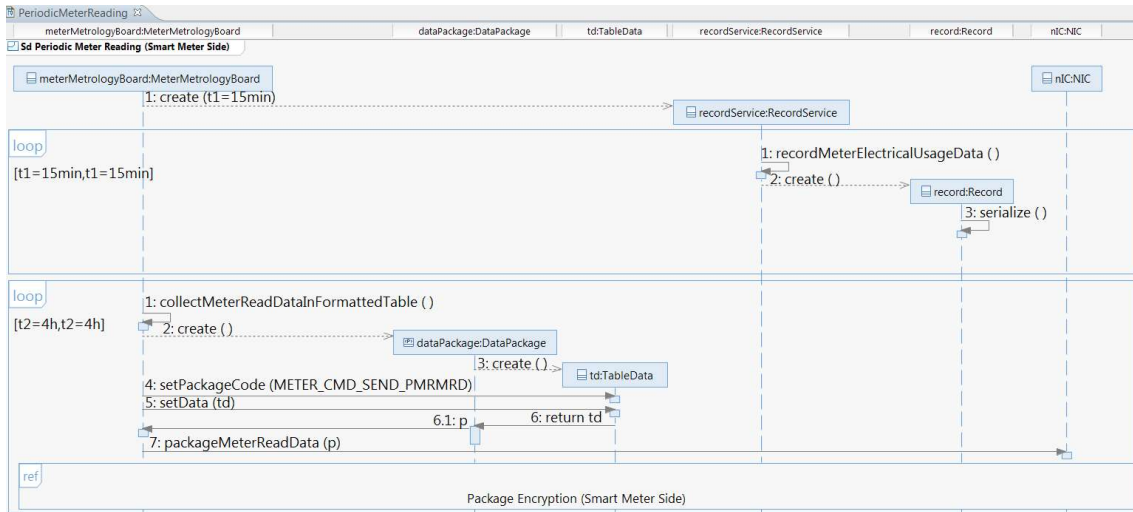


Figure 47 Periodic Meter Reading Sequence Diagram (Smart Meter Side)

Periodic Meter Reading Sequence Diagram (AMI Head-End Side)

In this sequence diagram, there are some methods related to security design for authentication. In order to send or receive package from smart meter or AMI head-end, smart meter should be authenticated at the AMI head-end. The authentication can be identified by the session status. If the session is active, it means smart meter has already been authenticated by AMI head-end. If the session is timed out, authentication process should be done again. There are some methods performed to identify the session status. Smart meter controller calls the look up session at the session manager. Session manager returns the session. Session manager calls the check time out method at session. The result of check time out will be saved in a Boolean value called “r”. “R” can be either session active or session timed out. If the session is still active, there is no need to do authentication process again. AMI head-end will receive meter read data from smart meter by re-using “Package Decryption (AMI Head-End Side)” sequence diagram using ref structure. Then, the package, which is meter read data will be authorized by re-using “Authorization (AMI Head-End Side)” sequence diagram.

On the other hand, if the session is timed out, re-authentication process should be performed. In this diagram, we refer the re-authentication process to “Smart Meter Authentication (AMI Head-End Side)” sequence diagram.

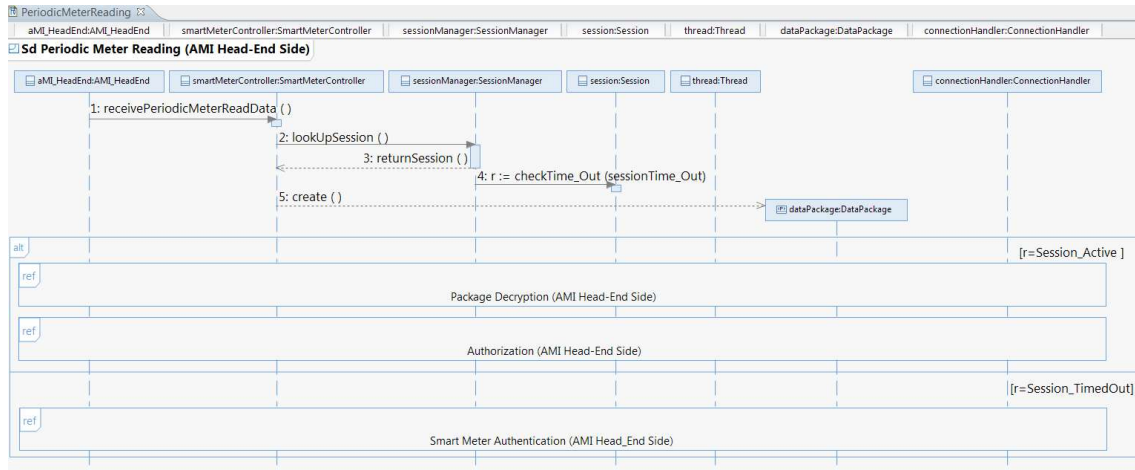


Figure 48 Periodic Meter Reading Sequence Diagram (AMI Head-End Side)

6.2.7 Remote Meter Connect Sequence Diagram

Remote Meter Connect Sequence Diagram (AMI Head-End Side)

In the process of remote meter connect, AMI head-end should send remote meter connect message to smart meter. Since this is a sensitive message, smart meter needs to be authenticated by AMI head-end before AMI head-end sends remote meter connect message to smart meter. The session needs to be active. There are two conditions, which are shown in an alternative flow. These conditions are if the session is active or if the session is timed out. To check the status of session, there are some methods performed before alternative flow. First, CIS sends remote meter connect request message to AMI head-end. Since CIS is in different system, we do not show CIS as an object. It is shown as a found message. Therefore, the sender is not identified. In this case, we use circle instead of the sender of the message. The trigger method for this diagram is the method call from AMI head-end to smart meter controller. AMI head-end calls the method remote meter connect of smart meter controller. Then methods such as look up session and check time out will be performed. If the session is still active, then there is no need to re-authentication process. AMI head-end can send the remote meter connect message to smart meter by re-using “Package Encryption (AMI Head-End Side)” sequence diagram. Then AMI head-end will receive a package from smart meter including internal meter switch verification message. The receive package is done by re-using “Package Decryption (AMI Head-End Side)” sequence diagram. After that ref structure, there is a ref structure of “Authorization (AMI Head-End Side)” sequence diagram. Then AMI head-end will send the internal meter switch verification message to CIS.

On the other hand, if the session is timed out, the re-authentication process will be performed using ref structure.

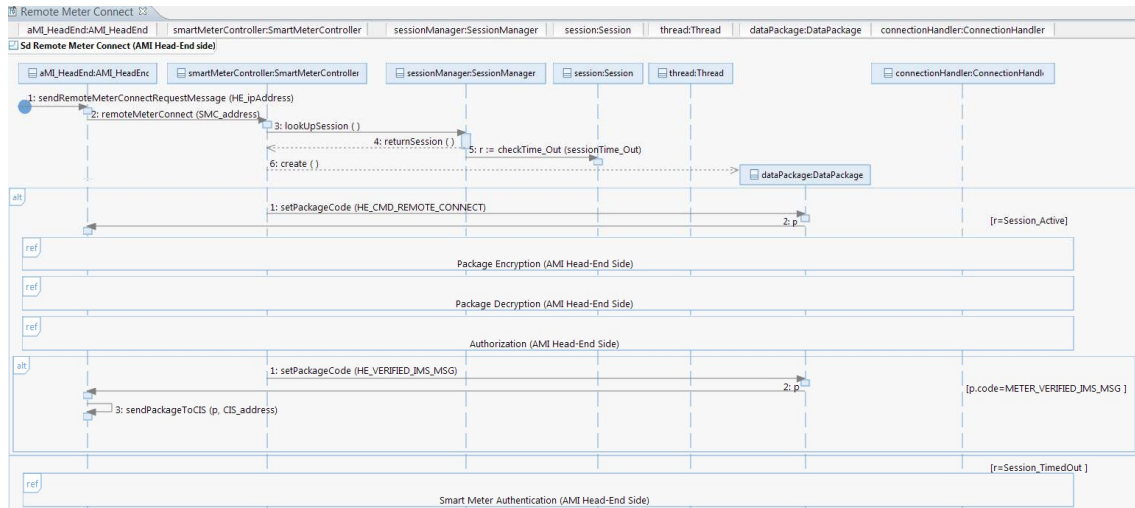


Figure 49 Remote Meter Connect Sequence Diagram (AMI Head-End Side)

Remote Meter Connect Sequence Diagram (Smart Meter Side)

In the figure below, in the first steps there is a ref structure. This part is referenced to the “Smart Meter Authentication (Smart Meter Side)” sequence diagram. The reason for using the ref structure is in the smart meter side, we want to keep the session id sent by head-end for security reasons. After that, there is another ref structure for receiving the package from AMI head-end. We reuse “Package Decryption (Smart Meter Side)” sequence diagram. The received package code is “HE_CMD_REMOTE_CONNECT”. It means the received package is the remote meter connect message sent by AMI head-end. After receiving this package, Meter metrology board will control internal meter switch. Internal meter switch will execute RCD command. RCD command means remote meter connect disconnect command. Since this diagram shows remote meter connect, therefore the internal meter switch will close the meter switch in order to connect the meter. Meter metrology board will create closed internal meter switch verification message. The package code for the verification message will be set to “METER_VERIFIED_IMS_MSG”. At the end, meter metrology board will send this package to AMI head-end through NIC by using ref structure to “Package Encryption (Smart Meter Side)” sequence diagram.

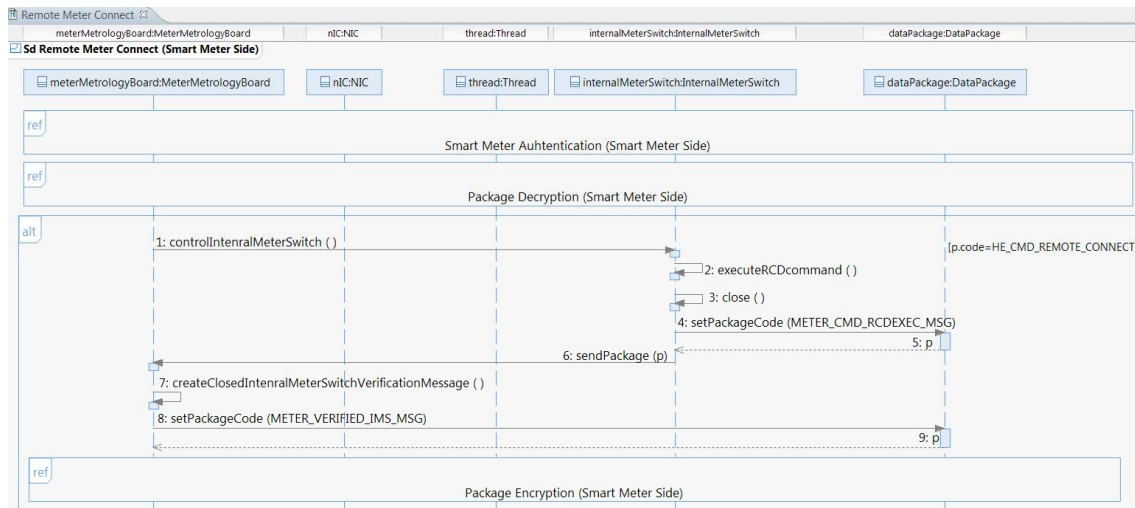


Figure 50 Remote Meter Connect Sequence Diagram (Smart Meter Side)

6.2.8 Remote Meter Disconnect Sequence Diagram

Remote Meter Disconnect Sequence Diagram (AMI Head-End Side)

The remote meter disconnect process is very similar to remote meter connect. The difference is in the package code. In remote meter disconnect, instead of sending remote meter connect message to smart meter by AMI head-end, it sends the remote meter disconnect message to disconnect the smart meter. Additionally, the verification message will be opened internal meter switch verification message instead of closed verification message. The steps are the same. In this diagram, we consider if the session is active or it is timed-out, as well. The trigger is calling remote meter disconnect method from AMI head-end to smart meter controller. Before this method, CIS sends the remote meter disconnect request message to AMI head-end. There are the same methods such as look up session and check time out to consider the status of the session. If the session is active, head-end will send the remote meter disconnect message to smart meter. If the session is timed out, there is a smart meter re-authentication process using ref structure. At the end, AMI head-end will send the internal meter switch verification message to CIS.

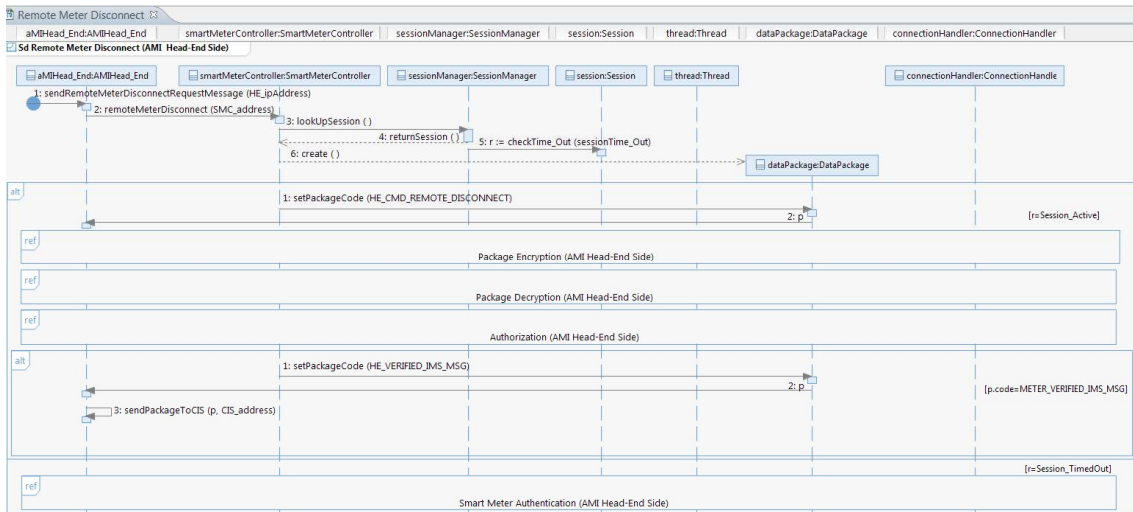


Figure 51 Remote Meter Disconnect Sequence Diagram (AMI Head-End Side)

Remote Meter Disconnect Sequence Diagram (Smart Meter Side)

This sequence diagram is similar to the remote meter connect of smart meter side. The differences are in the package code of “HE_CMD_REMOTE_DISCONNECT”. Instead of connect it is disconnect. First, similar to remote meter connect (smart meter side) sequence diagram, there is a ref structure. This is used for keeping the session id received from AMI head-end for security purposes. Then there is receiving package process using ref structure. The received package code is “HE_CMD_REMOTE_DISCONNECT”. When smart meter receives this package, meter metrology board controls internal meter switch. Internal meter switch executes the RCD command and opens the meter switch in order to disconnect the meter. Then meter metrology board creates opened internal meter switch verification message. At the end, meter metrology board will send the internal meter switch verification message to AMI head-end through NIC by using ref structure.

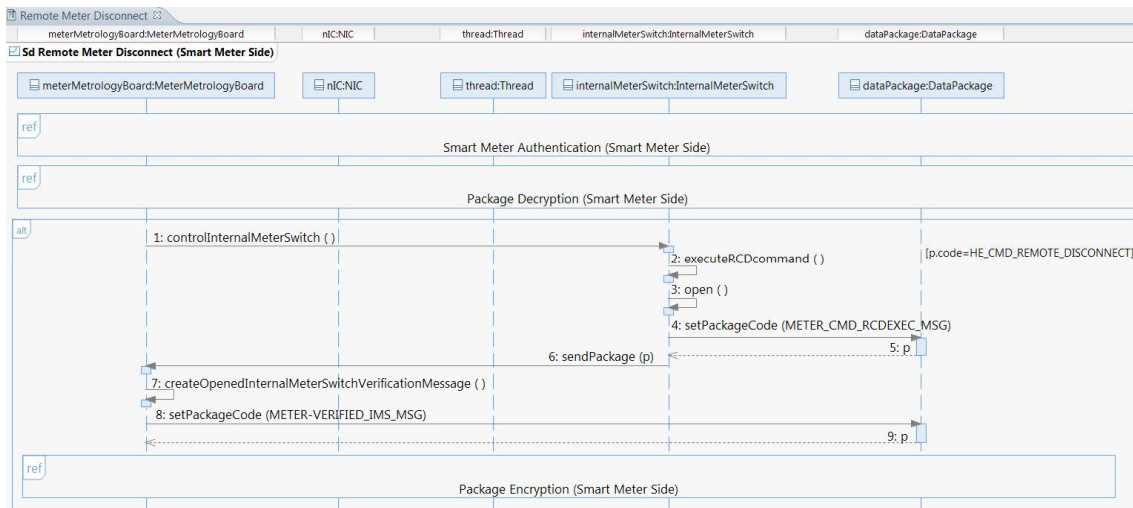


Figure 52 Remote Meter Disconnect Sequence Diagram (Smart Meter Side)

6.2.9 On-Demand Meter Reading Sequence Diagram

On-Demand Meter Reading Sequence Diagram (AMI Head-End Side)

In the process of on-demand meter reading, AMI head-end sends on-demand meter read request message to smart meter. Since this message is sensitive, there is a need to authentication process. Therefore, the session status should be checked. To check the session status, similar to pervious sequence diagrams, methods such as look up session and check time out are performed. Before performing these methods, there should be a trigger. First, CIS sends on-demand meter read request to AMI head-end. Then AMI head-end calls on-demand meter reading method of smart meter controller. After check time out method, there is an alternative flow to show if the session is active or if it is timed out. If the session is active, there is no need to re-authentication process. In this case, AMI head-end will send the on-demand meter read request message to smart meter by using ref structure. Then for receiving on-demand meter read data, there are two ref structures for decrypting the package and authorizing the package. On the other hand, if the session is timed out, there is a ref structure to show the re-authentication process. This part is referenced to “Smart Meter Authentication (AMI Head-End Side)” sequence diagram. At the end, after receiving on-demand meter read data from smart meter, AMI head-end will send this package to CIS.

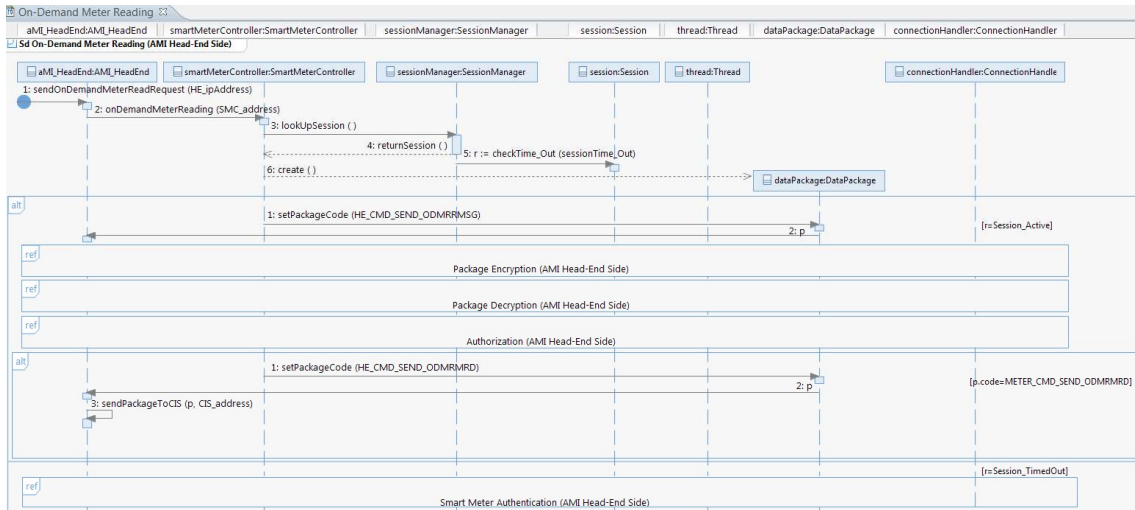


Figure 53 On-Demand Meter Reading Sequence Diagram (AMI Head-End Side)

On-Demand Meter Reading Sequence Diagram (Smart Meter Side)

In the process of on-demand meter reading in the smart meter side, first there is a ref structure. It is used for showing that the session id, which is received from AMI head-end is stored. Then there is receiving package process from AMI head-end using ref structure. The package code of received package is “HE_CMD_SEND_ODMRRMSG”. It means AMI head-end sends the on-demand meter read request message to smart meter. After receiving the package, in the alternative flow, MMB retrieves meter read data in formatted table. At the end, after setting corresponding package code and setting data, MMB sends the on-demand meter read data to AMI head-end through NIC by using ref structure.

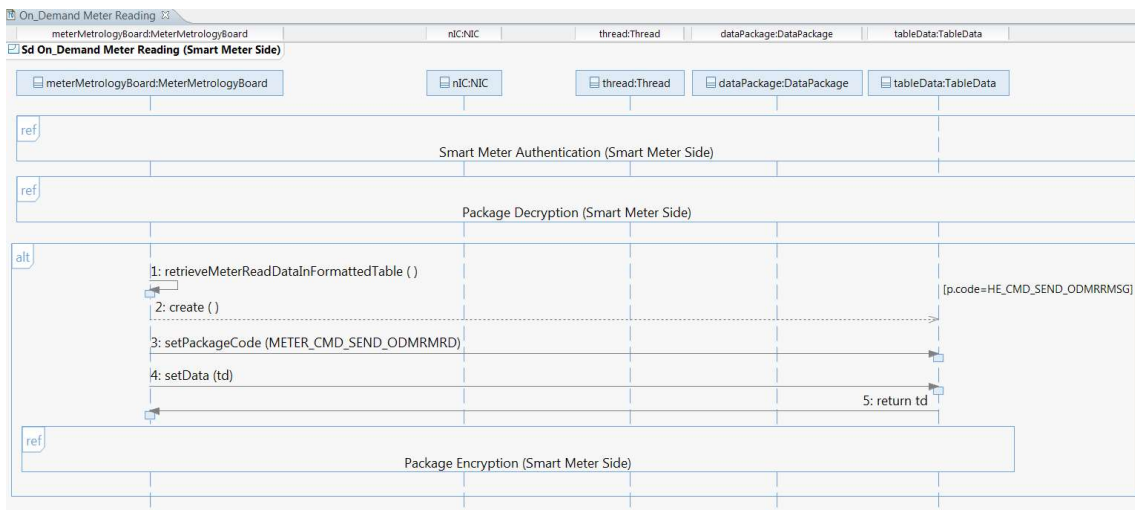


Figure 54 On-Demand Meter Reading Sequence Diagram (Smart Meter Side)

6.2.10 Misuse Model Sequence Diagram for City Blackout Uncertainty

Misuse Model Sequence Diagram for City Blackout Uncertainty (AMI Head-End Side)

This sequence diagram shows the misuse model for designing example of uncertainty in city black outs. In this example, attacker can introduce uncertainty in the smart grid by forwarding malicious shut down commands to smart meter. It results in city blackouts, which leads to uncertainty in the smart grid. We designed the steps that attacker can do in the AMI system to cause black outs in the sequence diagram. By modeling, we specifically show how attackers can introduce uncertainty by executing the malicious shut down commands. This sequence diagram is kind of misuse model. Misuse model means the composition of primary or functional model with attack model. Since shut down commands are related to remote meter connect/disconnect case, therefore, we use remote meter connect sequence diagram as a primary model. Then we add attacker as an object in this sequence diagram to make a misuse model.

The description of this sequence diagram is like this: First, the similar steps, which were in the primary model of remote meter connect are performed. For example, CIS sends remote meter connect request message to AMI head-end. AMI head-end sends remote meter connect to smart meter controller. Session manager looks up for the session to see the status of the session. If the session is active, it means smart meter has already been authenticated by AMI head-end. Therefore, AMI head-end can send the message to smart meter. We assume that the session is active. In the session active status, smart meter controller sets the package code to send the remote meter connect message to smart meter. However, there is an attacker in the AMI system. When smart meter controller tries to send a remote meter connect message to smart meter, the attacker captures package by eavesdropping to the network connections. Then the scenario is like this: After attacker captures the remote meter connect message sent by smart meter controller, the attacker changes the message to remote meter disconnect message by performing modify attack action. Then the attacker sends the remote meter disconnect message to the smart meter and performs terminate action to terminate the smart grid. This remote meter disconnect message is kind of malicious shut down commands sent by attacker. Therefore, the smart meter will be turned off. It leads to city black outs. If the session is timed out, smart meter will be re-authenticated by AMI head-end.

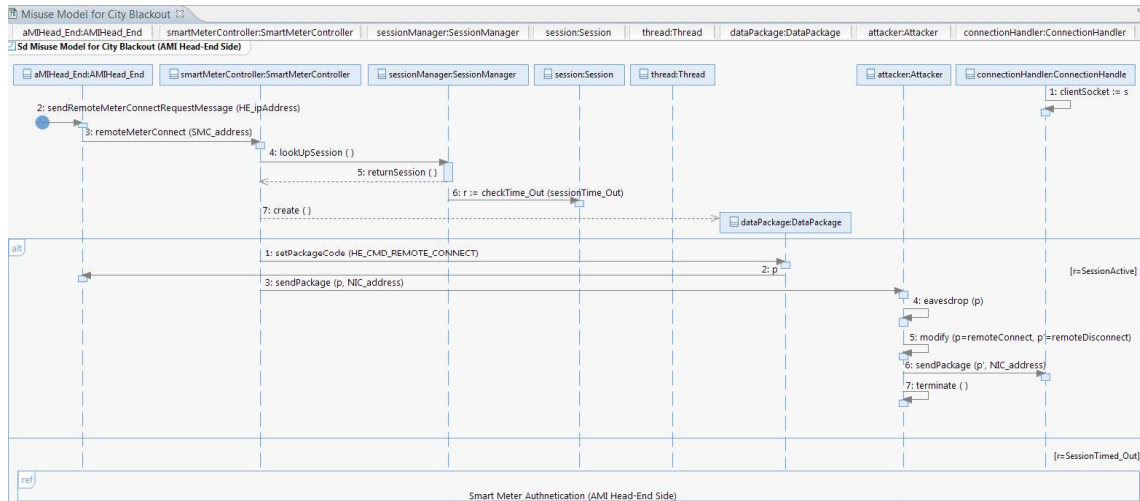


Figure 55 Misuse Model Sequence Diagram for City Blackout Uncertainty (AMI Head-End Side)

Misuse Model Sequence Diagram for City Blackout Uncertainty (Smart Meter Side)

This sequence diagram is a misuse model for city blackouts from the smart meter side. In this sequence diagram, smart meter instead of receiving remote meter connect message from AMI head-end, receives the remote meter disconnect message and acts based on the remote meter disconnect message.

The description of the design in this sequence diagram is like this: First, thread receives a package by listening to the network. This package is remote meter connect message sent from AMI head-end. However, there is an attacker in the way. The attacker eavesdrops and catches the remote meter connect message. Then the attacker modifies the content of package and changes the package to remote meter disconnect message and sends it to MMB. MMB processes the package. However, since the attacker changed the package to remote meter disconnect message, MMB assumes the package as remote meter disconnect message and behaves based on this package. Therefore, in the alternative flow the package code is HE_CMD_REMOTE_DISCONNECT instead of HE_CMD_REMOTE_CONNECT. MMB controls internal meter switch. The attacker catches this package and tries to control internal meter switch. Since the package code is remote meter disconnect, internal meter switch opens the meter switch instead of closing it, which means it will disconnect the smart meter. MMB will create the opened internal meter switch verification message instead of closed verification message and sends it to AMI head-end.

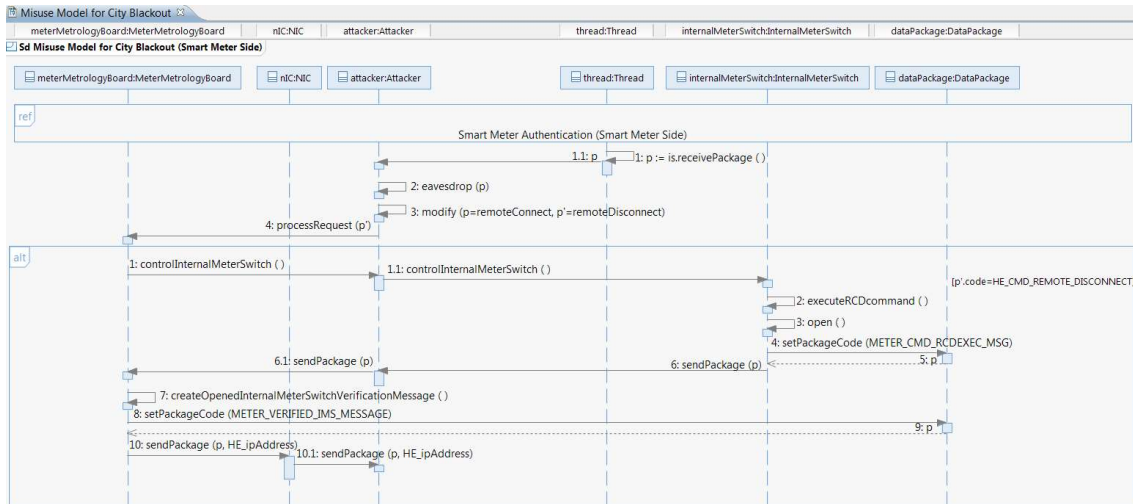


Figure 56 Misuse Model Sequence Diagram for City Blackout Uncertainty (Smart Meter Side)

6.2.11 Misuse Model Sequence Diagram for Signing Package Uncertainty

Misuse Model Sequence Diagram for Signing Package Uncertainty (AMI Head-End Side)

This sequence diagram shows the misuse model for signing package uncertainty. We draw this sequence diagram by composition of primary sequence diagram of package encryption (AMI Head-End Side) and attack model. When AMI head-end sends a package to smart meter, AMI head-end first encrypts the package before sending for the security purposes. Then the package will be signed with digital signature. However, the attackers can introduce uncertainties in package signing process by getting inside the AMI system.

The description of this sequence diagram is like this: an attacker can get into the AMI system. The attacker then creates two packages with equal MD5 sums in MD5 hash function. One of the packages is original package and the other one is the package containing malicious stuff called flawed package. Then attacker decides to encrypt the original package (p) and sign the original package by sending them to smart meter controller to encrypt and sign the original package. Now, the original package is signed successfully. However, since both of the packages have the same MD5 sum values, the attacker can modify the package and replace the original package, which is signed with flawed package. The attacker now sends the flawed package (p') to smart meter through connection handler. Since the MD5 sums are equal, it takes a long time to detect the attack. This is some kind of uncertainty in the AMI system.

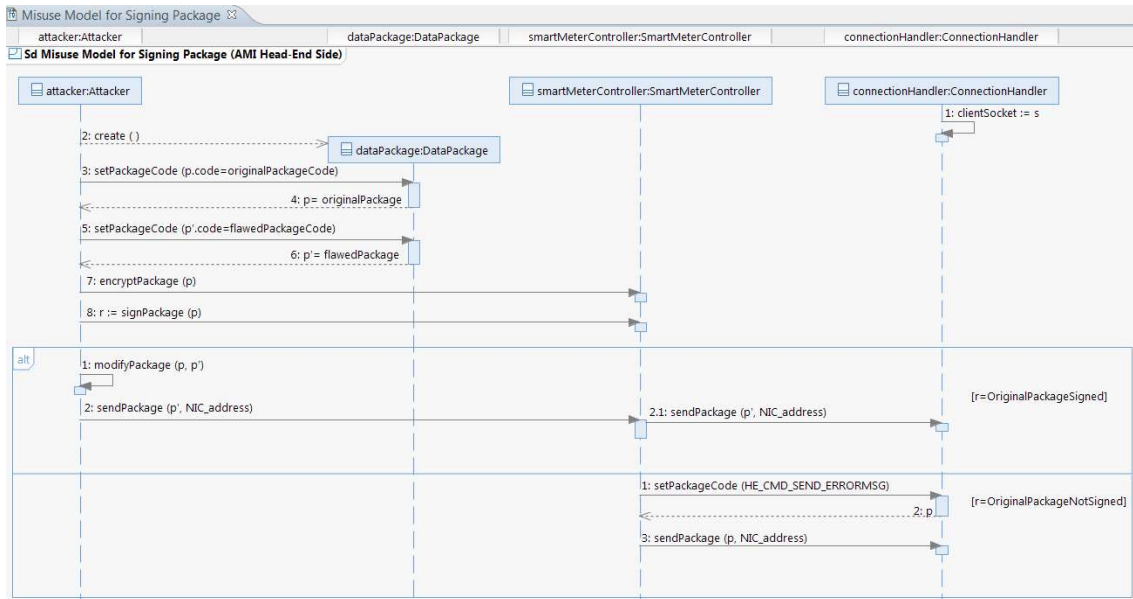


Figure 57 Misuse Model Sequence Diagram for Signing Package Uncertainty (AMI Head-End Side)

Misuse Model Sequence Diagram for Signing Package Uncertainty (Smart Meter Side)

This sequence diagram is similar to the sequence diagram of signing package uncertainty from AMI head-end side. The difference is in this sequence diagram, we have smart meter’s components. This misuse model is composition of encryption sequence diagram (Smart Meter Side) and attack model.

The description of this sequence diagram is similar to the previous sequence diagram from the AMI head-end side. The attacker creates two packages; one original package and one flawed package. The attacker sends original package to meter metrology board for encrypting and signing. Now, the original package (p) is signed successfully. The attacker decides to change the package by replacing the original package with flawed package, which contains malicious stuff since both of the packages have equal MD5 sums. The attacker then will send the flawed package (p’) to AMI head-end. This is some kind of uncertainty in the AMI system.

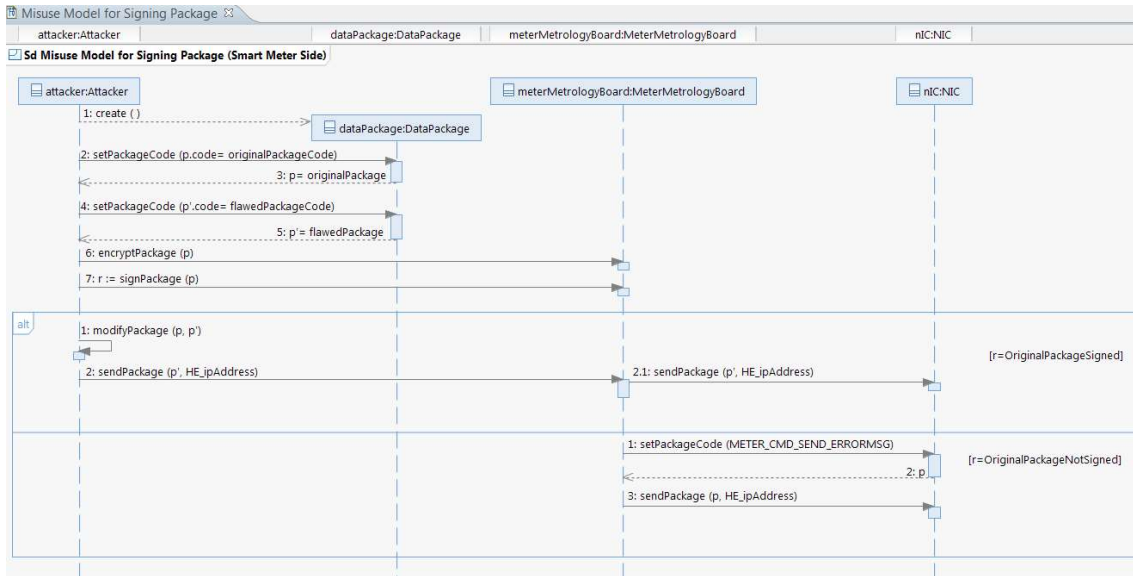


Figure 58 Misuse Model Sequence Diagram for Signing Package Uncertainty (Smart Meter Side)

6.3 State Chart Diagrams

6.3.1 Smart Meter Registration State Chart Diagram

Summary Smart Meter Registration State Chart Diagram (AMI Head-End Side)

This state chart diagram is composed of two composite states: “Establish Connection” and “Authentication”. Establish connection is the first step in smart meter registration process. Authentication is the second step in smart meter registration process. There is an initial state before establish connection state starts. This is the initial state for the whole of this state chart diagram. There is also final state at the end of authenticate composite state. This is the final state of whole registration state diagram.

In establish connection composite state, there are some states, which indicate the state of AMI head-end when AMI head-end wants to establish connection with smart meter. First, AMI head-end is disconnected. Then AMI head-end waits for connection. If AMI head-end accepts the connection, AMI head-end will be in connected state. Else if there is a connection error, AMI head-end is still in disconnected state. The transitions are the method calls in class diagrams.

In Authentication composite state, first, AMI head-end receives smart meter’s credentials. Then it will verify the digital signature of the received package. If the package is verified, it will be in state of decrypting package. If not, it will be in state of sending error message to smart meter. After package

decryption, smart meter’s credentials will be verified. Then state of AMI head-end will be changed to Authenticating smart meter. After authentication, AMI head-end should also authorize smart meter. After that, AMI head-end will send the acknowledgment to smart meter to show that smart meter is authenticated. However, before that the acknowledgment package should be encrypted and signed. Therefore, these are two states between sending authenticated acknowledgment message to smart meter state. At the end, there will be final state and the authentication composite state will end there. If verifying smart meter’ s credentials and signing package states are not successful, AMI head-end will change its state to sending error message to smart meter. Additionally, if authorizing smart meter is not successful, AMI head-end will change its state to sending error message to smart meter controller.

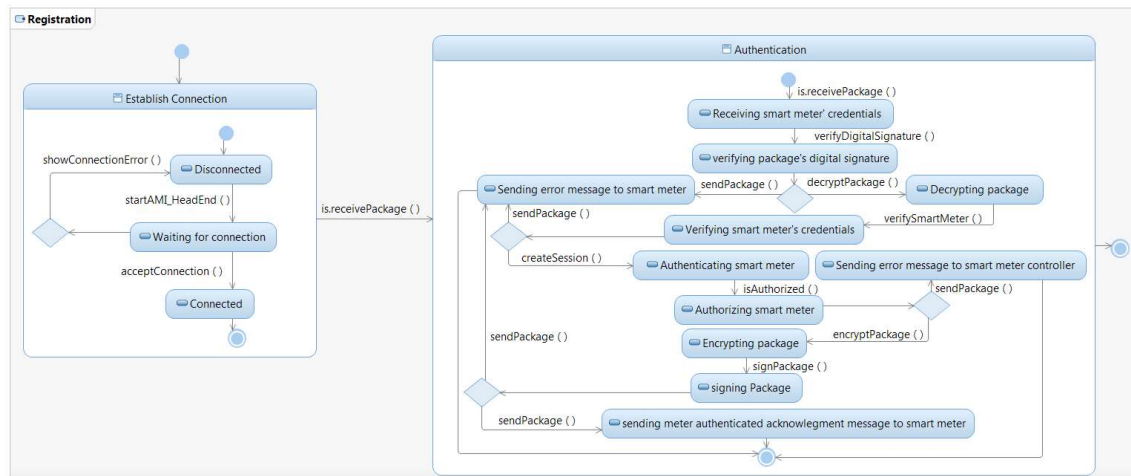


Figure 59 Registration State chart diagram (AMI Head-End Side)

Smart Meter Registration State Chart Diagram (Smart Meter Side)

This is the state chart diagram of smart meter registration from smart meter side. This diagram is also composed of two composite states: “Establish Connection” and “Authentication” composite states.

Establish connection composite state from smart meter side is similar to establish connection composite state from AMI head-end side. First, smart meter is in disconnected state. Because, there is no connection established between smart meter and AMI head-end. Then smart meter waits for connection. If the connection is ok, smart meter changes its state to connected state. If there is connection error, smart meter will be in disconnected state.

In Authentication composite state, first smart meter is not authenticated. Then smart meter receives connected acknowledgment message from AMI head-end. Smart meter will verify the digital signature of the received package. If it can verify the digital signature, it will decrypt the package. After that, smart meter will encrypt the package, which includes its credentials, signs it, and sends the credentials to AMI head-end to be authenticated by AMI head-end. After sending credentials state, the state will

change to receiving meter authenticated acknowledgment message. At the end, the state of smart meter is smart meter is authenticated. On the other hand, if verifying package's digital signature and signing package states fail, smart meter will send error message to AMI head-end.

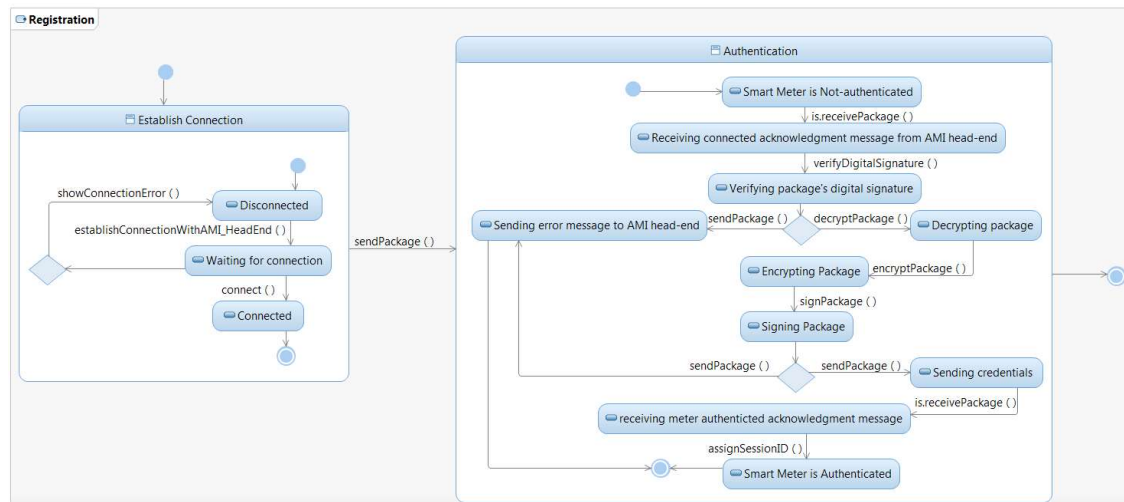


Figure 60 Registration State chart diagram (Smart Meter Side)

6.3.2 Periodic and On-Demand Meter Reading State Chart Diagram

Periodic and On-Demand Meter Reading State Chart Diagram (AMI Head-End Side)

In this state chart diagram, we use orthogonal state. We divided periodic meter reading and on-demand meter reading by using orthogonal state. The name of the orthogonal state is meter reading.

In periodic meter reading, first the session status will be checked. If the session is active, the meter read data would be received from smart meter. Then the package including meter read data would be decrypted. At the end, AMI head-end will authorize smart meter to check if the meter read data sent by smart meter is valid and secure. If the session is timed-out, smart meter will be re-authenticated. Then AMI head-end will change its state to receiving meter read data from smart meter and the next following steps will be performed until the state chart diagram reaches its final state.

On-demand meter reading is similar to periodic meter reading. First, the session status will be checked. If the session is active, AMI head-end will encrypt and send on-demand meter read request message to smart meter. After that, AMI head-end will receive on-demand meter read data from smart meter. It will decrypt the package, authorizes smart meter and sends meter read data to CIS. If the session is timed-out, the smart meter will be re-authenticated and again the steps from encrypting package until final state will be performed.

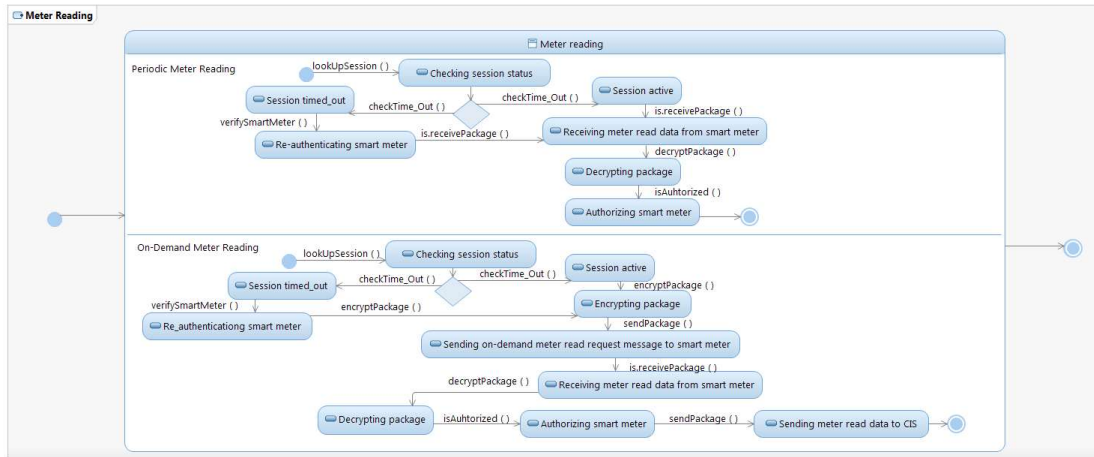


Figure 61 Periodic and on-demand Meter Reading State Chart Diagram (AMI Head-End Side)

Periodic and On-Demand Meter Reading State Chart Diagram (Smart Meter Side)

In this state chart diagram, we use orthogonal state. We divided periodic meter reading and on-demand meter reading by using orthogonal state. The name of the orthogonal state is meter reading.

In periodic meter reading, first, the state of smart meter is recording. Then it is collecting, after that encrypting package and at the end, the state is sending meter read data to AMI head-end.

In on-demand meter reading, first the state is receiving on-demand meter read request message from AMI head-end. By receiving a package, the package will be decrypted. Then the states are retrieving and encrypting package. At the end, the state is sending meter read data to AMI head-end.

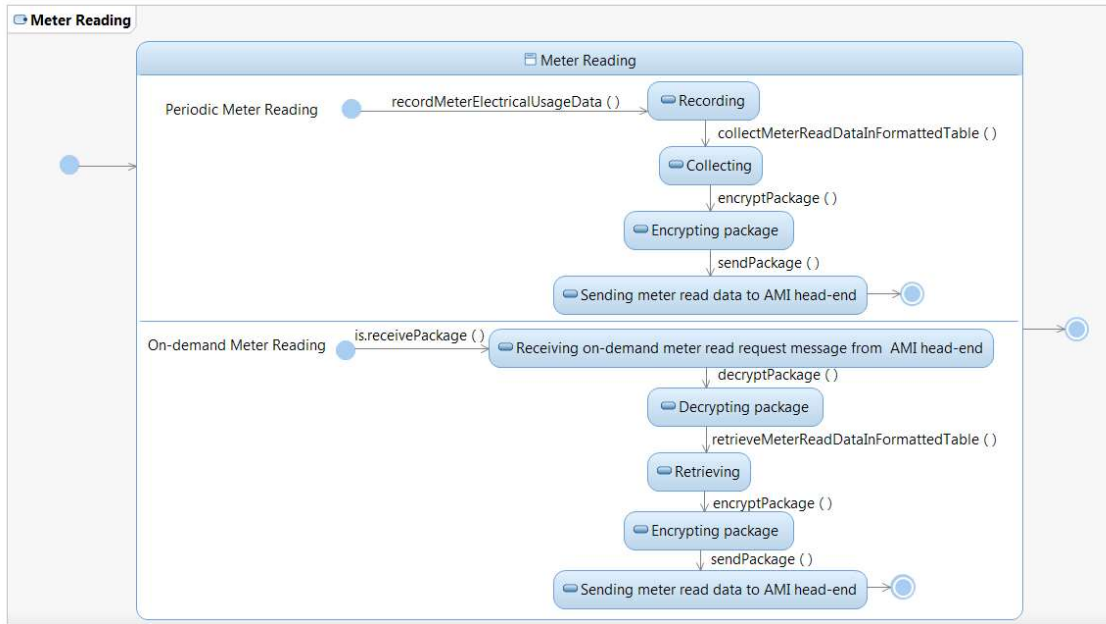


Figure 62 Periodic and on-demand Meter Reading State Chart Diagram (Smart Meter Side)

6.3.3 Remote Meter Connect/Disconnect State Chart Diagram

Remote Meter Connect/Disconnect State Chart Diagram (AMI Head-End Side)

Remote meter connect/disconnect state chart diagram is composed of two composite states. These composite states are “Remote Meter Connect” and “Remote Meter Disconnect”.

In remote meter connect, first, the session status will be checked. If the session is active, remote meter connect message will be encrypted and will be sent to smart meter. The following states will be receiving closed internal meter switch verification message from smart meter and decrypting package, and authorizing smart meter. The last state is sending closed internal meter switch verification message to CIS. If the session is timed-out, smart meter will be re-authenticated and the states will be followed from encrypting package until the final state.

Remote meter disconnect part is similar to remote meter connect. The difference is instead of sending remote meter connect message to smart meter, remote meter disconnect message will be sent to smart meter. The other difference is opened internal meter switch verification message will be sent to AMI head-end and at the end, will be sent to CIS.

The transition from remote meter connect composite state to remote meter disconnect composite state is remote disconnect. The transition from remote meter disconnect composite state to remote meter connect composite state is remote connect.

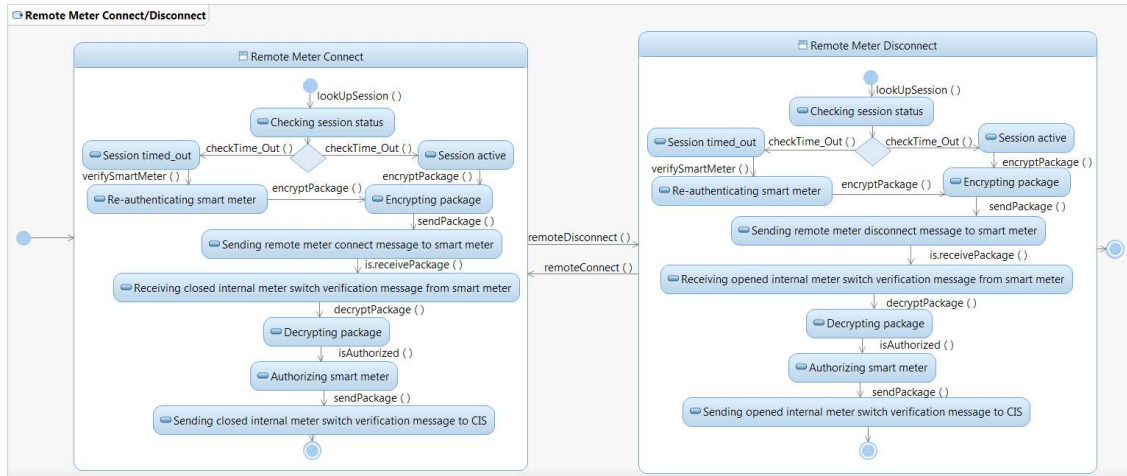


Figure 63 Remote Meter Connect/Disconnect State Chart Diagram (AMI Head-End Side)

Remote Meter Connect/Disconnect State Chart Diagram (Smart Meter Side)

This Diagram is composed of two composite states: “Remote Meter Connect” and “Remote Meter Disconnect”. In remote meter connect, first remote meter connect message is received, and then it is decrypted. The following states are controlling, executing, closing meter switch, and encrypting package. The last state is sending closed internal meter switch verification message to AMI head-end.

Remote meter disconnect composite state is similar to remote meter connect. The differences are in these steps: receiving remote meter disconnect message from AMI head-end, opening meter switch, and sending opened internal meter switch verification message to AMI head-end.

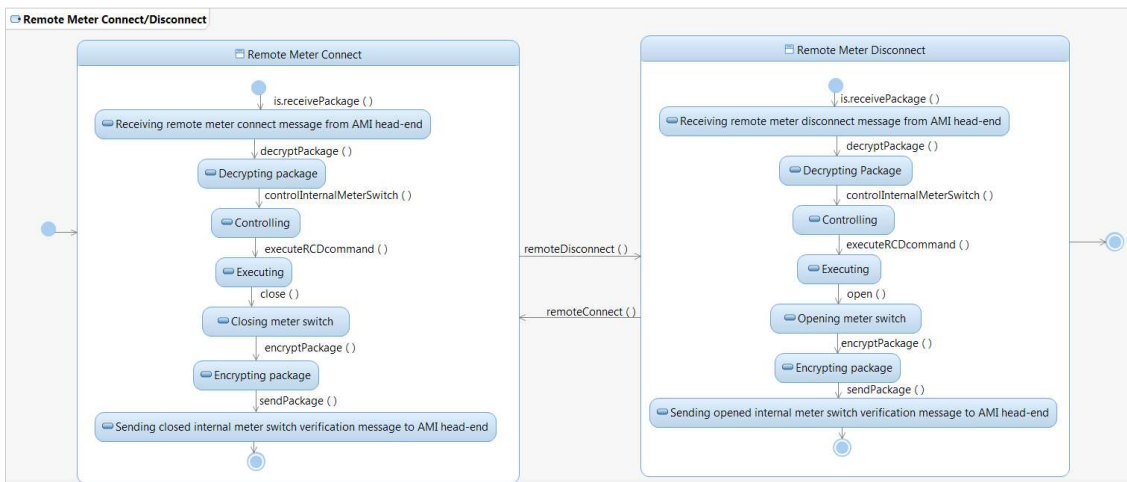


Figure 64 Remote Meter Connect/Disconnect State Chart Diagram (Smart Meter Side)

7 Conclusion

In this thesis, we worked on the specific application of CPSs called smart grid. Smart grids generate electricity power and transmit this power to different categories of customers such as industries, factories, houses, etc. We specifically focused on main parts of smart grid systems called AMI and smart meter and considered the communication between them. Our case study was about AMI system. Based on this case study, we modeled some core functionalities of AMI system including periodic meter reading, remote meter connect/disconnect and on-demand meter reading. Besides designing these core functionalities of AMI system, we also modeled the security aspects of AMI system. It was one of the main contributions of the thesis. Additionally, we modeled some security-related uncertainties of AMI system. We used UML for designing and modeling part as the main methodology. The tool for designing the models was IBM RSA.

The most important part in this thesis was to model security aspects of AMI followed by modeling some security-related uncertainties of AMI. In order to design security part, first we collected some main security requirements of AMI system. They are Confidentiality, Integrity and Availability. Then we tried to show some security mechanisms or solutions to address security requirements. Security is important subject. By ensuring security, we can prevent the attackers to violate the security requirements and harm the system. There are some security mechanisms, which we provided in the thesis to handle the security requirements of AMI system. These mechanisms are such as Authentication, Authorization, Encryption and Decryption. We showed how we could make the AMI system more secure by introducing these mechanisms to our design. We designed all of these mechanisms in all types of UML diagrams given in the thesis.

Our conclusion is that the security mechanisms can be used to address security requirements if they are designed in a proper way using better algorithms or solutions. For example, in encryption and decryption mechanism, there are different types of cryptographic algorithms to encrypt or decrypt data. However, our research result was based on the criticality of data. Therefore, we decided to choose the proper cryptographic algorithm based on the data sensitively. In remote meter connect/disconnect, we concluded that asymmetric cryptographic algorithm is better solution in providing better security. However, in periodic meter reading, hybrid approach was the better solution to provide better security to the AMI system.

In term of uncertainty, we showed some diagrams, which introduce uncertainty to the AMI system. However, we need to define some security solutions to address the security-related uncertainties of AMI system. The work that we conducted in the thesis for uncertainty part was to collect some security-related uncertainties of smart grid in general or AMI system in specific. Then, we selected

some of them for designing. We showed in the design how attackers could play role in the AMI system and introduce some uncertainties in the AMI system.

The conclusion of uncertainty part is that attackers can make severe damages to the AMI system. The city blackout is one of the major impacts that can happen in the AMI system. Attackers can be a reason for city blackout uncertainty. City blackouts can have some unpleasant effects in the people's lives and economy. Therefore, it is necessary to make some plans to prevent these kinds of uncertainties. We can address uncertainties by providing some security solutions.

8 Future Work

In future work, we want to collect more functionalities of AMI system and design them. It can give us better overview of AMI system that leads us to do more research in AMI system in detail and smart grid in general.

In case of security design, we plan to continue the research to find the best and optimal solutions regarding to each security requirement with taking into account different factors such as the cost of security solutions, the performance of each security implementation solution, etc.

We plan to provide some security solutions to address security-related uncertainties of AMI system. Furthermore, our plan is to investigate more security-related uncertainties of AMI system and map them to the design.

9 References

1. Güngör, V.C., et al., *Smart grid technologies: communication technologies and standards*. Industrial informatics, IEEE transactions on, 2011. **7**(4): p. 529-539.
2. Hahn, A. and M. Govindarasu, *Cyber attack exposure evaluation framework for the smart grid*. Smart Grid, IEEE Transactions on, 2011. **2**(4): p. 835-843.
3. SGiP, *Introduction to NISTIR 7628 guidelines for smart grid cyber security*. 2010, The Smart Grid Interoperability Panel - Cyber Security Working Group.
4. Hartmann, T., et al., *Reactive Security for Smart Grids Using Models@run.time-Based Simulation and Reasoning*, in *Smart Grid Security: Second International Workshop, SmartGridSec 2014, Munich, Germany, February 26, 2014, Revised Selected Papers*, J. Cuellar, Editor. 2014, Springer International Publishing: Cham. p. 139-153.
5. Berthier, R. and W.H. Sanders. *Specification-based intrusion detection for advanced metering infrastructures*. in *Dependable Computing (PRDC), 2011 IEEE 17th Pacific Rim International Symposium on*. 2011. IEEE.
6. Strategy, N.M.G., *Advanced metering infrastructure*. US Department of Energy Office of Electricity and Energy Reliability, 2008.
7. Cleveland, F.M. *Cyber security issues for advanced metering infrastructure (ami)*. in *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*. 2008. IEEE.
8. Wang, E.K., et al. *Security issues and challenges for cyber physical system*. in *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*. 2010. IEEE Computer Society.
9. Alfonso, B.M.C.V.P., *Formal Methods for Model-Driven Engineering*. Vol. 7320. 2012: Springer-Verlag Berlin Heidelberg.
10. Bézivin, J., *In search of a Basic Principle for Model-Driven Engineering*. Novatica -- Special Issue on UML (Unified Modeling Language), 2004. **5**(2): p. 21-24.
11. Aksit, U.A.M. and A. Rensink, *Model Driven Architecture*. 2005.
12. Ghosh, D., *DSL for the uninitiated*. Commun. ACM, 2011. **54**(7): p. 44-50.
13. Deursen, A.v., P. Klint, and J. Visser, *Domain-specific languages: an annotated bibliography*. SIGPLAN Not., 2000. **35**(6): p. 26-36.
14. Mens, T. and P. Van Gorp, *A Taxonomy of Model Transformation*. Electronic Notes in Theoretical Computer Science, 2006. **152**: p. 125-142.
15. Czarnecki, K., and S. Helsen, *Classification of Model Transformation Approaches*, in *2nd OOPSLA'03 Workshop on Generative Techniques in the Context of MDA*. 2003: Anaheim, CA, USA.
16. Eriksson, H.-E. and M. Penker, *UML toolkit*. 1998: John Wiley & Sons, Inc. 397.
17. Berardi, D., D. Calvanese, and G. De Giacomo, *Reasoning on UML class diagrams*. Artificial Intelligence, 2005. **168**(1): p. 70-118.

18. Rumbaugh, J., I. Jacobson, and G. Booch, *Unified Modeling Language Reference Manual, The*. 2004: Pearson Higher Education.
19. Cartaxo, E.G., F.G. Neto, and P.D. Machado. *Test case generation by means of UML sequence diagrams and labeled transition systems*. in *2007 IEEE International Conference on Systems, Man and Cybernetics*. 2007. IEEE.
20. Fuentes-Fernández, L. and A. Vallecillo-Moreno, *An introduction to UML profiles*. UML and Model Engineering, 2004. **2**.
21. Pandey, R.K., *Object constraint language (OCL): past, present and future*. SIGSOFT Softw. Eng. Notes, 2011. **36**(1): p. 1-4.
22. Yue, T., L.C. Briand, and Y. Labiche, *A use case modeling approach to facilitate the transition towards analysis models: Concepts and empirical evaluation*, in *Model Driven Engineering Languages and Systems*. 2009, Springer. p. 484-498.
23. Lee, E.A. *Cyber Physical Systems: Design Challenges*. in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. 2008.
24. Karsai, G. and J. Sztipanovits, *Model-Integrated Development of Cyber-Physical Systems*, in *Software Technologies for Embedded and Ubiquitous Systems: 6th IFIP WG 10.2 International Workshop, SEUS 2008, Anacapri, Capri Island, Italy, October 1-3, 2008 Proceedings*, U. Brinkschulte, T. Givargis, and S. Russo, Editors. 2008, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 46-54.
25. Shi, J., et al. *A survey of Cyber-Physical Systems*. in *2011 International Conference on Wireless Communications and Signal Processing (WCSP)*. 2011.
26. Lee, E.A., *Cyber-physical systems-are computing foundations adequate*, in *Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*. 2006: Austin, TX, USA.
27. Rajkumar, R., et al. *Cyber-physical systems: The next computing revolution*. in *Design Automation Conference*. 2010.
28. Sha, L., et al., *Cyber-physical systems: A new frontier*, in *Machine Learning in Cyber Trust*. 2009, Springer. p. 3-13.
29. Alvaro Cardenas, S.A., Bruno Sinopoli, Annarita Giani, Adrian Perrig, Shankar Sastry, *Challenges for Securing Cyber Physical Systems*, in *Workshop on Future Directions in Cyber-physical Systems Security*. 2009.
30. Agency, I.E., *Technology Roadmap: Smart Grids*. 2011, International Energy Agency
31. Nicholson, M., *The Power Makers' Challenge: And the Need for Fission Energy* 2012 Edition ed. Green Energy and Technology. 2012: Springer; 2012 edition.
32. Ali, A.B.M.S., *Smart Grids: Opportunities, Developments, and Trends*, in *Smart Grids*. 2013, Springer-Verlag London.
33. Farhangi, H., *The path of the smart grid*. IEEE Power and Energy Magazine, 2010. **8**(1): p. 18-28.
34. Nguyen, P.H., et al., *A Systematic Review of Model-Driven Security*, in *Software Engineering Conference (APSEC, 2013) 20th Asia-Pacific*. 2013. p. 432-441.

35. Nguyen, P.H., et al., *An extensive systematic review on the Model-Driven Development of secure systems*. Information and Software Technology, 2015. **68**: p. 62-81.
36. Nguyen, P.H., S. Ali, and T. Yue, *Model-based security engineering for cyber-physical systems: A systematic mapping study*. Information and Software Technology, 2017. **83**: p. 116-135.
37. Cuellar, J., *Smart Grid Security*. 2013: Springer.
38. Baumeister, T., *Literature review on smart grid cyber security*. Collaborative Software Development Laboratory at the University of Hawaii, 2010.
39. Man Zhang, B.S., Shaukat Ali, Tao Yue, Dipesh Pradhan, Stefan Nastic, Hong-Linh Truong, Martin Schneider, Max Bureck, Christian Hein, *Testing Cyber-Physical Systems under Uncertainty: Systematic, Extensible, and Configurable Model-based and Search-based Testing Methodologies*. 2015. p. 1-57.
40. Man Zhang, S.A., Tao Yue, Phu Hong Nguyen. *Uncertainty Modeling Framework for the Integration Level V.I*. 2016.
41. Jürjens, J., *Model-Based Security Engineering with UML*, in *Foundations of Security Analysis and Design III: FOSAD 2004/2005 Tutorial Lectures*, A. Aldini, R. Gorrieri, and F. Martinelli, Editors. 2005, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 42-77.
42. Hartmann, T., et al. *Generating realistic smart grid communication topologies based on real-data*. in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*. 2014. IEEE.
43. Taein, H., et al. *Design of application-level reference models for micro energy grid in IT perspective*. in *2012 8th International Conference on Computing and Networking Technology (INC, ICCIS and ICMIC)*. 2012.
44. EPRI, *Smart Grid Resource Center*. 2011, Electric Power Research Institute.
45. Mohassel, R.R., et al. *A survey on advanced metering infrastructure and its application in Smart Grids*. in *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*. 2014.
46. Fouda, M.M., et al., *A Lightweight Message Authentication Scheme for Smart Grid Communications*. IEEE Transactions on Smart Grid, 2011. **2**(4): p. 675-685.
47. Schumacher, M., et al., *Security Patterns: Integrating security and systems engineering*. 2013: John Wiley & Sons.
48. Wang, W. and Z. Lu, *Cyber security in the Smart Grid: Survey and challenges*. Computer Networks, 2013. **57**(5): p. 1344-1371.
49. Sinnhofer, A.D., et al., *Patterns to establish a secure communication channel*, in *Proceedings of the 21st European Conference on Pattern Languages of Programs*. 2016, ACM: Kaufbeuren, Germany. p. 1-21.
50. Tripathi, R. and S. Agrawal, *Comparative study of symmetric and asymmetric cryptography techniques*. International Journal of Advance Foundation and Research in Computer (IJAFRC), 2014. **1**(6): p. 68-76.
51. Zafar, N., et al., *System security requirements analysis: A smart grid case study*. Systems Engineering, 2014. **17**(1): p. 77-88.

52. Lu, R., et al., *Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications*. IEEE Transactions on Parallel and Distributed Systems, 2012. **23**(9): p. 1621-1631.
53. Brown, B., et al., *AMI system security requirements*. UCA Int. Users Group, US Dept. Energy, Washington, DC, USA, Tech. Rep. UCAIUG: AMI-SEC-ASAP, 2008.
54. Butler, K.R., et al., *A Survey of BGP Security Issues and Solutions*. Proceedings of the IEEE, 2010. **98**(1): p. 100-122.
55. Zio, E. and T. Aven, *Uncertainties in smart grids behavior and modeling: What are the risks and vulnerabilities? How to analyze them?* Energy Policy, 2011. **39**(10): p. 6308-6320.
56. Fleury, T., H. Khurana, and V. Welch, *Towards A Taxonomy Of Attacks Against Energy Control Systems*, in *Critical Infrastructure Protection II*, M. Papa and S. Sheno, Editors. 2008, Springer US: Boston, MA. p. 71-85.
57. McHenry, M.P., *Technical and governance considerations for advanced metering infrastructure/smart meters: Technology, security, uncertainty, costs, benefits, and risks*. Energy Policy, 2013. **59**: p. 834-842.
58. Misra, S., et al., *ENTRUST: Energy trading under uncertainty in smart grid systems*. Computer Networks, 2016. **110**: p. 232-242.
59. Mikle, O., *Practical Attacks on Digital Signatures Using MD5 Message Digest*. IACR Cryptology ePrint Archive, 2004. **2004**: p. 356.
60. Efthymiou, C. and G. Kalogridis. *Smart Grid Privacy via Anonymization of Smart Metering Data*. in *2010 First IEEE International Conference on Smart Grid Communications*. 2010.
61. Deep, K. and M. Thakur, *A new mutation operator for real coded genetic algorithms*. Applied Mathematics and Computation, 2007. **193**(1): p. 211-230.
62. Nguyen, P.H., M. Papadakis, and I. Rubab. *Testing Delegation Policy Enforcement via Mutation Analysis*. in *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops*. 2013.
63. Jia, Y. and M. Harman, *An Analysis and Survey of the Development of Mutation Testing*. IEEE Transactions on Software Engineering, 2011. **37**(5): p. 649-678.