Networking for Smart Meters

CHAITANYA DANDUGULA

KTH Information and Communication Technology

Degree project in
Communication Systems
Second level, 30.0 HEC
Stockholm, Sweden

Networking for Smart Meters

Chaitanya Dandugula
dandu@kth.se

June 19, 2012

Master of Science Thesis
Communication Systems

Examiner: Professor Gerald Q. Maguire Jr.
Kungliga Tekniska Högskolan
Stockholm, Sweden

Supervisor: Peter Schoo
Fraunhofer Research Institution for Applied and Integrated Security
Munich, Germany

Abstract

"Smart grid" generally refers to a class of technology bringing electricity delivery systems into the 21st century, using computer-based remote control and automation. With the growing energy demand, efficient usage of the available energy resources is increasingly becoming a major issue around the world. Smart grid is a step in that direction. Research in the European Union and the United States are currently underway to modernize the existing and aging transmission grid and to streamline the usage of electricity.

A typical electricity grid consists of two major entities - the utility company and the distribution control system (DCS). Electricity is generated at the utility company and the DCS is responsible for the distribution of electricity to individual homes/consumers. A smart meter (SM) is an electronic device that measures the electricity consumed at the consumer's premises and provides added information to the utility company. The data concentration unit (DCU) is a device acting as a communication hub collecting and encoding data from multiple smart meters in a neighborhood and forwarding the data to the utility company. The aim of this project is to design a network for securing the communication between the SM and the DCU in a smart metering network environment.

The meter data communicated from the SM to the DCU is very sensitive and in the hands of an attacker, can reveal significant personal information about an individual. Hence it is of at most importance to protect the meter data transmitted from the SM. On the other hand the control signals transmitted from the DCU to the SM, need protection in order to thwart off unauthorized signals (i.e., an intruder can impersonate the DC and send out control signals to the SMs). Hence the SM and the DCU should be authenticated by each other and authorized and the data and/or control signals exchanged between them should be encrypted.

Sammanfattning

"Smart grid" avser i allmänhet en klass av teknik föra system elleverans till 21: a århundradet, med hjälp av datorbaserade fjärrkontroll och automation. Med den ökande efterfrågan på energi, är effektiv användning av de tillgängliga energiresurser blir alltmer en viktig fråga över hela världen. Smart grid är ett steg i den riktningen. Forskning i Europeiska unionen och USA för närvarande pågår för att modernisera befintliga och åldrande transmissionsnätet och effektivisera användningen av el.

En typisk elnätet består av två större enheter - de allmännyttiga företaget och "distribution control system"(DCS). El genereras vid verktyget företaget och DCS ansvarar för distributionen av el till enskilda hem / konsumenter. En smart meter (SM) är en elektronisk apparat som mäter elförbrukning på konsumentens lokaler och ger ökad information till elbolaget. "Data concentration unit"(DCU) är en enhet fungerar som ett kommunikationsnav insamling och kodning av data från flera smarta mätare i ett område och vidarebefordra data till elbolaget. Syftet med detta projekt är att utforma ett nätverk för att säkra kommunikationen mellan SM och DCU i ett smart mätning nätverksmiljö.

Mätaren uppgifter som lämnas från SM till DCU är mycket känslig och i händerna på en angripare, kan avslöja viktig personlig information om en individ. Följaktligen är det av som mest betydelse för att skydda de mätdata som sänds från SM: en. Å andra sidan styrsignaler överförs från DCU till SM och behöver skydd för att hindra av obehöriga signaler (dvs en inkräktare kan personifiera DC och skicka ut styrsignaler till SM). Därför SM och DCU ska bestyrkas av varandra och godkänts och data och / eller styrsignaler utväxlas mellan dem ska vara krypterad.

Acknowledgements

I would like to thank my supervisor and examiner at KTH, Professor Gerald Q. Maguire Jr. I am glad he accepted to be my academic examiner since he considered this project very interesting. Prof. Maguire's guidance has also been essential in some steps of this thesis, such as the the analysis and the over all format of the report.

I would like to sincerely thank my supervisor Peter Schoo at Fraunhofer AISEC. Mr. Schoo introduced me to an interesting concept of modernizing the electricity grid using the latest computer networking technologies. We have had interesting discussions and he has provided feedback during the project.

There are more people who also deserve great thanks. They are my colleagues at Fraunhofer AISEC. Finally, I would like to thank my friends and my family, all of whom have been encouraging me during my stay in Stockholm.

Thank you all.

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

6LoWPAN IPv6 Low power Wireless Personal Area Network

AMI Automated Metering Infrastructure

CBC Cipher Block Chaining

DCS Distribution Control System

DCU Data Concentration Unit

DNS Domain Name System

DTLS Datagram Transport Layer Security

HAN Home Area Network

HI Host Identifiers

HIP Host Identity Protocol

HIT Host Identity Tag

IKEv2 Internet Key Exchange version 2

LDAP Lightweight Directory Access Protocol

LLN Low-power and Lossy Networks

MAC Media Access Control

MAN Metrological Area Network

PKI Public Key Interface

PLC Power Line Communication

PMTU Path Maximum Transmission Unit

PP Protection Profile

RAM Random Access Memory

RF Radio Frequency

RSA Rivest, Shamir and Adleman

SCTP Stream Control Transmission Protocol

SM Smart Meter

SMS Smart Metering System

SPI Security Parameter Index

SSL Secure Sockets Layer

TLS Transport Layer Security

WAN Wide Area Network

Chapter 1

Introduction

Section 1.1 presents an overview of the thesis. Section 1.2 describes the problem statement, section 1.3 lists the shortcomings of the thesis work, while section 1.4 presents the outline of the rest of the thesis report.

1.1 Overview

Electricity is fundamental to modern society and the economy. However, most of the world relies on electricity production and distribution systems built 50 years ago. Historically, the electricity grid has been a distribution grid, where a few central power generators provide all the electricity production in a country or a region and then 'distribute' this electricity to the consumers via a large network of cables and transformers. Such electricity grids are inefficient and cannot respond to today's urgent global challenges (such as, global warming, utilization of more renewable energy source, etc.). There is an estimated $13 trillion investment required in energy infrastructure over the next 20 years. The issues faced by the legacy electricity grid poses an imminent need and opportunity to shift towards a low carbon, efficient, and clean energy system.

A smart metering system is composed of many smart meters (SMs). Each SM is a device that measures the consumption or production of commodities such as electrical energy, gas, or water in a physical metrological measurement metering unit and transforms the measured values into digital information. Network attached SMs enable automated meter reading. Furthermore, a smart metering system can incorporate multiple data concentration units (DCUs), a distribution control system (DCS), and utility companies. The DCU is a device that is responsible for collecting meter data from several smart meters in a neighborhood and then transmitting this data to the DCS

and/or the utility company. The DCS is responsible for collecting the meter data from the DCUs, combining usage and price data to create the billing information for each consumer and providing additional data to the utility company. The utility company should utilize this data and design schemes for efficient energy generation and distribution. Note that, this thesis considers only an electricity smart metering system, thus extending this work for other utilities should be undertaken as a future work.

Using SMs for electricity, helps to conserve energy and reduce carbon dioxide emissions in two ways. Firstly, dynamic pricing, dependent on the current supply situation, encourages consumers to shift electricity consumptions to times when energy costs are low. This way, fewer peak-load electricity generation plants, which tend to be inefficient and often fueled by fossil fuels, are required. Secondly, SMs produce data that can be used to inform consumers their electricity consumption in an illustrative manner, which motivates these consumers to save energy.

Although real time monitoring is advantageous there are several security threats associated with collecting this data. For example, it is possible to extract usage patterns from the SM data that can help to identify the various electrical appliances used in a home. Armed with such data an attacker can obtain personal information of a particular individual thus compromising her/his privacy [29].

The communication between the SMs and the DCU is unmanaged, i.e., the SMs require self-configuration support, must ensure confidentiality of the communicated information, and this communication may need to take place in a hostile environment. This thesis project is based on the assumption that the communication between the DCU and the DCS is managed, i.e., the communication between the DCU an the DCS is suitably protected, thus it is assumed to secure.

1.2 Problem statement

This thesis project aims to define a network architecture for the secure communication of information (both metering & control information) between the SM and the DCU.

The project is divided into the following three phases:

1. Survey of different communication protocols suitable for the communication between the SM and DCU.

2. Survey of different security measures suitable for securing the communication between the SM and DCU.

3. Proposing a solution for the secure communication between the SM and DCU.

4. Defining communication use cases between the SM and DCU.

5. Analysis of the proposed solution.

1.3 Limitations of the thesis project

This thesis project does not

- Implement and analyze the proposed solution.
- Propose a solution for the communication between the DCU and / or the DCS and the utility company.
- Propose a solution for the communication between the DCS and the utility company.

1.4 Outline of the thesis

After the introduction to the problem statement in this first chapter, chapter 2 presents a survey of communication protocols suitable for the communication between the SM and DCU. Chapter 3 presents the security mechanism required for securing the communication between the SM and DCU. Communication use cases between the SM and the DCU are described in chapter 4. Chapter 5 analyses the proposed solution while chapter 6 presents the conclusion and proposes any future work based on this thesis project.

Chapter 2

Background

This chapter presents background information required to understand the work done in this thesis project. The chapter begins by reviewing relavent communication protocols, specifically PLC in section 2.1, 6LoWPAN in section 2.2, HIP in section 2.3, and DLMS/COSEM in section 2.4. Section 2.5 introduces the concept of a Smart Grid and the various components of a Smart Grid. Finally, section 2.6 highlights the details of an European Union (FP7) project (OPEN Meter Project) on standardizing SM communication throughout Europe.

2.1 Power line communication systems

Power line communication (PLC) is a technology for data communication over a conductor, used primarily for carrying electric power. PLC systems operate by impressing a modulated carrier signal on the power wiring system. The meter data can be collected at the consumer's premises and this data can be transmitted over the low voltage power lines to the nearest DCU and from there to the energy supplier and the DCS. A significant advantage of PLC systems is that the deployment costs are comparable to the wireless alternatives, as no additional infrastructure is required for communication and the power lines can be used for both power transmission and the communication of meter data [17].

Although PLC based Advanced Metering Infrastructure (AMI) has a proven track record, it lacks standardization . This lack of standardization could be a major factor regarding large scale commercial adaptation of PLC systems for AMI [18]. Standardization of the technology is necessary in order to promote competition between different utility companies, thus empowering the consumer to choose the best service suitable for her/ him. Apart

from the technical aspects of PLC systems, there are various governmental regulations and business requirements that may affect the mass acceptance of this technology.

An alternative to PLC is to use the IEEE 802.15.4 standard [20] for the physical and media sub-layers in order to transport IPv6 packets. Although the IEEE 802.15.4 standard is for a wireless medium at the physical layer, successful adaptation of the standard to a power line medium at the physical layer is presented by by Chauvenet, et al. [11]. IPv6 extends the IP address space from 32 to 128 bits and solves some very important issues, by incorporating auto configuration, a mandatory IPsec security implementation, and multicasting. IPv6 is increasingly necessary in order to grow the "Internet of things".

A proof-of-concept implementation of IPv6 over PLC was presented by Chauvenet, et al. [11]. This implementation uses PLC nodes that are architecturally similar to classic RF based IEEE 802.15.4 nodes. These nodes are powered by micro-controllers and the communication is handled by a PLC transceiver which emulates a radio transceiver. The micro-controller processes frames in the IEEE 802.15.4 frame format and the upper layers of the communication stack support IPv6, while the transceiver provides a modem with a throughput of 10 kbps. However, there are some adaptations within the MAC part of the protocol. These adaptations enable communication over a power line using the IEEE 802.15.4 frame format.

2.2 IPv6 over Low power Wireless Personal Area Networks

The term Low-power and Lossy Networks (LLNs) refers to networks that are composed of highly constrained nodes (limited power, memory, and CPU) connected by "lossy" links (low power radio links or PLC that have a higher probability than traditional wired communication links for errors). A LoWPAN is a particular type of LLN, formed by devices/nodes complying with the IEEE 802.15.4 standard that form a low-power wireless personal area network (LoWPAN). The 6LoWPAN standard provides for header compression and encapsulation mechanism to transport IPv6 packets over IEEE 802.15.4 based networks [27].

Typical characteristics of LoWPAN nodes are [24, 27]:

- Short range: The operating range of the nodes is about 10 meters.
- Low power: The transmission power of the nodes is set at around 0 to 3dBm.

- Limited memory: The nodes typically have only 512 KB of Flash memory and a very limited amount of random access memory (RAM).
- Limited processing power: Although some devices have 16-bit and 32-bit cores, the most common nodes have only 8-bit processors with clock rates of around 10 MHz.
- Low bit rate: A maximum over-the-air data rate of 250 kbps is typical of most types of the nodes.

2.3 Host identity protocol

HIP [30] proposes a new name space consisting of Host Identities and Host Identifiers (HI). There is a subtle, but important difference between the two. A Host Identifier is cryptographic in nature; that is it is the public key of an asymmetric key-pair. A Host Identity refers to an abstract name for a 'computing platform'. Each host is uniquely identified by a Host Identity and a corresponding Host Identifier (see Figure 2.1). Note that a single host can have more than one Host Identity. The HIP architecture is shown in Figure 2.1.

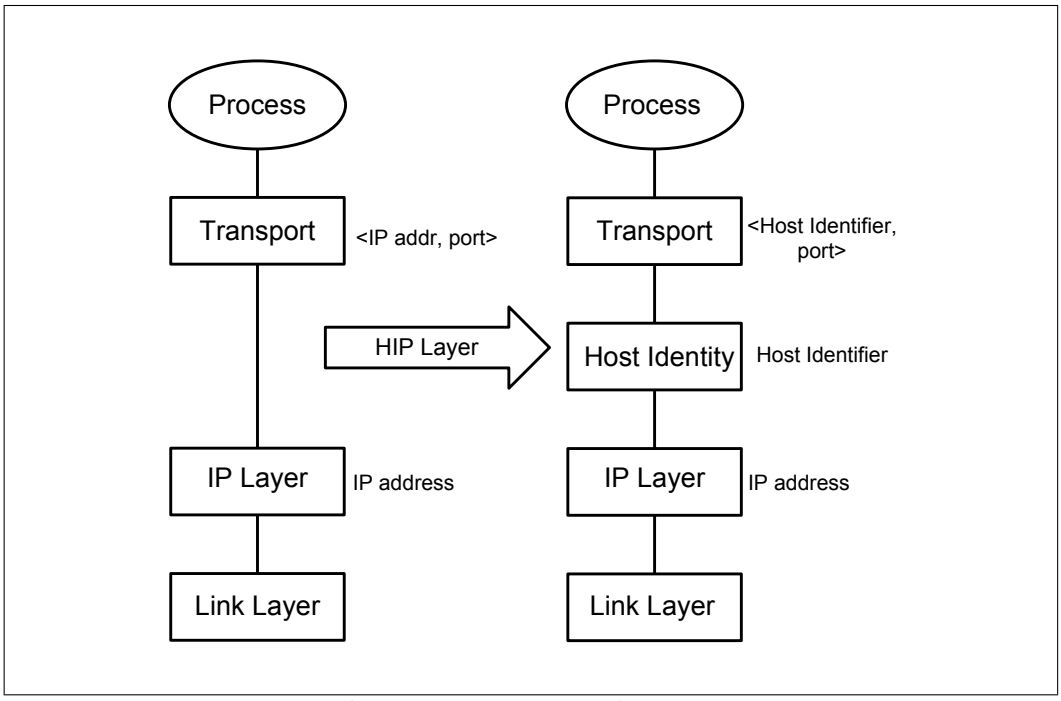

Figure 2.1: HIP architecture

In the current IP architecture, an IP address acts as both a locator and an end point identifier. That is, the IP addresses identifies the host's interface's topological location in the Internet and acts as the name of the physical network interface at the point-of-attachment. With HIP, end-point names and locators are two distinct entities. While IP addresses continue to act as locators, the Host Identifiers correspond to end-point names as was shown in Figure 2.1.

The main objectives of HIP are to enhance mobility, provide for limited forms of trust between systems, to enable dynamic IP renumbering, and to support multi-homing. The Host Identifiers can be used in many authentication systems, such as the Internet Key Exchange (IKEv2) [19] protocol, thus the payload traffic between the HIP host is typically, but not necessarily protected with IPsec. This causes these IP packets to be no different than the standard IPsec protected IP packets. In other words, HIP can be seen as a special case of using IPsec [22], thus it builds on top of the existing IPsec infrastructure.

2.3.1 Host Identity name space

A Host Identifier is a name in the Host Identity name space. Host Identifiers represents a statistically globally unique name for naming any system with an IP stack. The intent of a statistically globally unique name is to enable distributed systems to uniquely identify a host without the requirement for central coordination. Although, any name that claims to be 'statistically globally unique' may serve as a Host Identifier, a public key of a public-private key pair is recommended by Moskowitz and Nikander in [30] as the best choice for a Host Identifier.

HIP provides optional cryptographic features. The protocol (with its cryptographic features) provides the complete set of functionality described in RFC 4423 [30]. Using the public key as the Host Identifier avoids the need for an additional name. Host Identifiers can be public or private, i.e., they can be published or unpublished. The public Host Identifiers can be stored in a DNS or in LDAP [21] directories. Alternatively these identifiers can be stored in various kinds of Public Key Infrastructures (PKIs), hence extending the scope of a Host Identifier beyond simply providing host identification.

2.3.2 HIP overview

HIP uses a Host Identity Tag (HIT). A HIT is a 128 bit representation of a Host Identity and its value is computed as a cryptographic hash of the corresponding Host Identifier. The benefits of hashing the Host Identifier,

rather than directly using the Host Identifier are two fold. Firstly, this provides a consistent representation of the Host Identity irrespective of the cryptographic algorithms used. Secondly, hashing the Host Identifier provides a simpler protocol encoding because of its fixed length. Two HITs are used to (statistically) identify the sender and recipient of a packet.

Using the mathematics of 'birthday paradox', we can generalize that for a random hash space of 'n' bits, a collision is expected after approximately $1.2 \times \sqrt{2^n}$ hashes. For 64 bits, this number is roughly 4 billion. In other words, for 100 bits (or more) of hash size, we would not expect a collision until approximately 2^{50} (1 quadrillion) hashes were generated. For these reason we can assume that the value of each HIT is 'practically unique' (or statistically unique) in the whole IP universe and in the rare case of a collision, i.e., a single HIT mapping to more than one Host Identity, then the Host Identifiers (public keys) will make the final difference in identifying a unique host.

The purpose of the HIP base exchange (see Figure 2.2) is to ensure that the peers indeed possess private keys corresponding to their host identifiers (i.e., their corresponding public keys). As a result, the base exchange creates a pair of IPsec Encapsulated Security Payload (ESP) Security Associations (SAs), one in each direction.

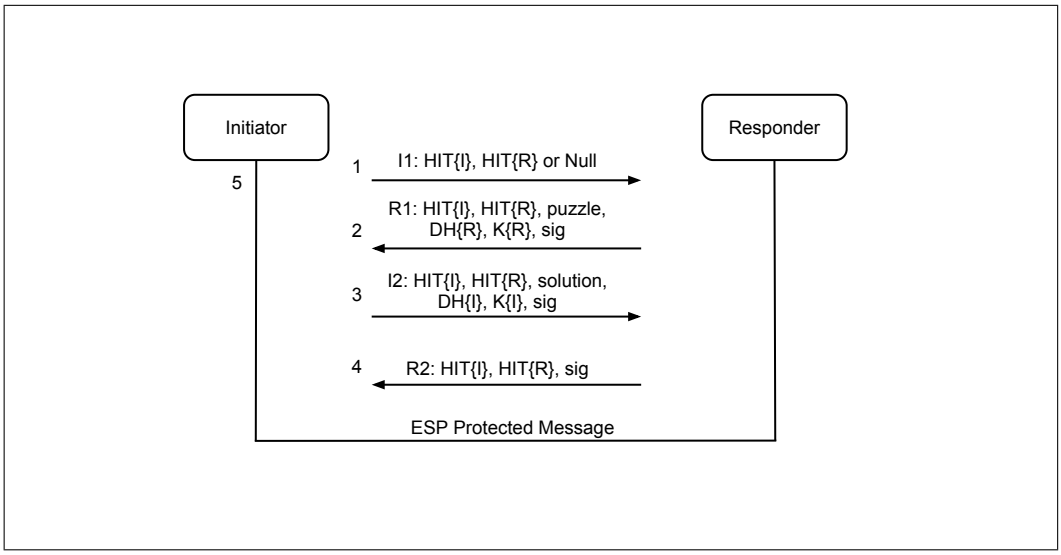

Figure 2.2: HIP base exchange

Figure 2.2 showed the base exchange process. First the initiator looks up the Host Identifier or HIT of the responder using DNS or a Rendezvous Server (RVS). Figure 2.3 depicts the HIP procedure with DNS. On the client side, the application sends a DNS query to a DNS server. The DNS server

replies with the Host Identifier by translating the fully qualified domain name (FQDN) to a host idenditifier (HI) instead of an IP address. In a second step, another lookup is made within the Host Identity layer by the HIP daemon. This time, the Host Identities are translated into IP addresses (i.e., a HI lookup yields an IP address) in order to enable a network layer delivery of a datagram.

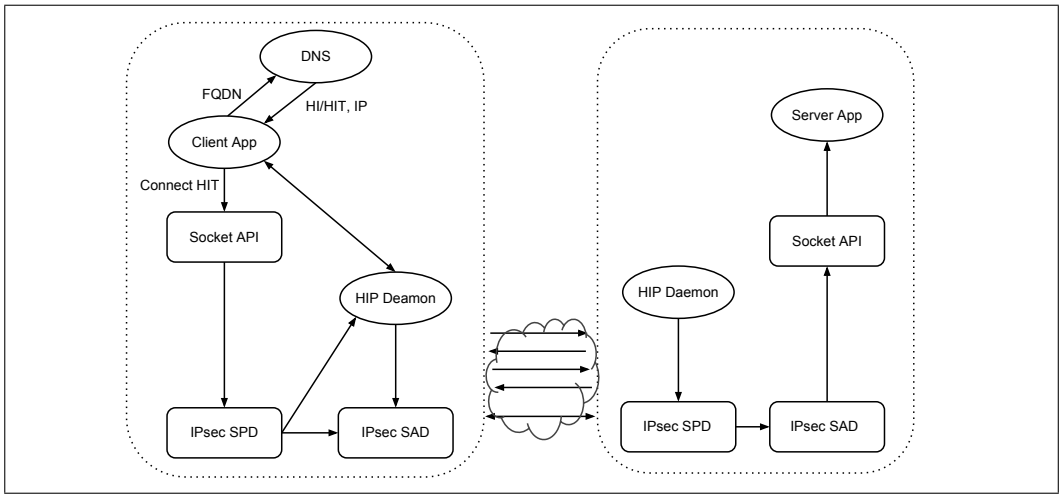

Figure 2.3: HIP with DNS

The transport protocol sends a packet containing the server's Host Identifier. The Host Identity layer replaces the Host Identifier with corresponding IP address of the server. The network layer transmits this packet with an IP header. Accordingly, the 5-tuple socket becomes <protocol, source HI, source port, destination HI, destination port> from the conventional <protocol, source IP, source port, destination IP, destination port>.

HIP uses a special IPsec ESP mode called Bound End-to-end Tunnel (BEET). The new mode provides limited tunnel mode semantics without the regular tunnel mode overhead.

2.3.3 Mobility

Since the SAs are not bound to IP addresses, the host is able to receive packets that are protected using a HIP-created ESP SA from any IP source address. Thus, a host can change its IP address and continue to send packets to its peers. Figure 2.4 depicts the mobility process. Initially, the mobile host is at address 1 and it later moves to the address 2. Due to this change in point of network attachment , the mobile host is disconnected from the peer host

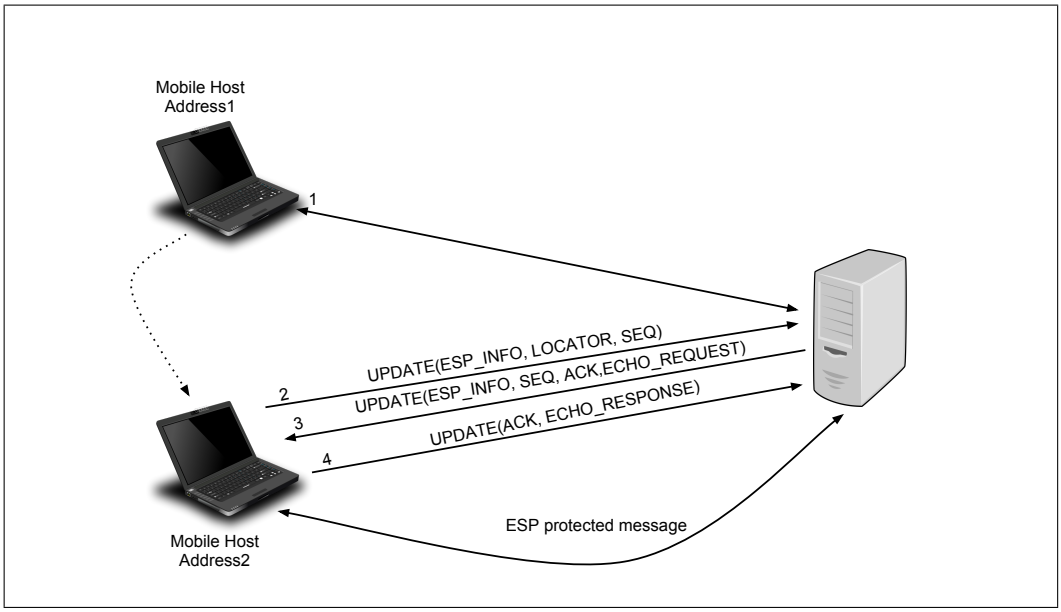

Figure 2.4: Mobility with HIP

for a brief period of time while it switches from address 1 to address 2. Upon obtaining a new IP address, the mobile host sends a `LOCATOR` parameter to the peer host in an `UPDATE` message. The `LOCATOR` indicates the new IP address, the IPsec - Security Parameters Index (SPI) associated with new IP address, the address lifetime, and whether the new address is a preferred address. The peer host performs an address check and solicits a response from the mobile host. Depending on whether the mobile host has initiated rekeying, and on whether the peer host itself wants to rekey in order to verify the mobile host's new address, the process can be categorized into three cases:

- Readdress without rekeying, but with an address check, as in Figure 2.4;
- Readdress with a mobile-initiated rekey; and
- Readdress with a peer-initiated rekey.

2.3.4 Multihoming

A host can sometimes have more than one interface. The host may notify the peer host of the additional interfaces by using the `LOCATOR` parameter. In Figure 2.5 the multihoming host is assumed to have two IP addresses, *addr1* and *addr2*. Further, *addr1* is assumed to be the preferred address. The multihoming host sends an `UPDATE` packet including *addr1* and *addr2*

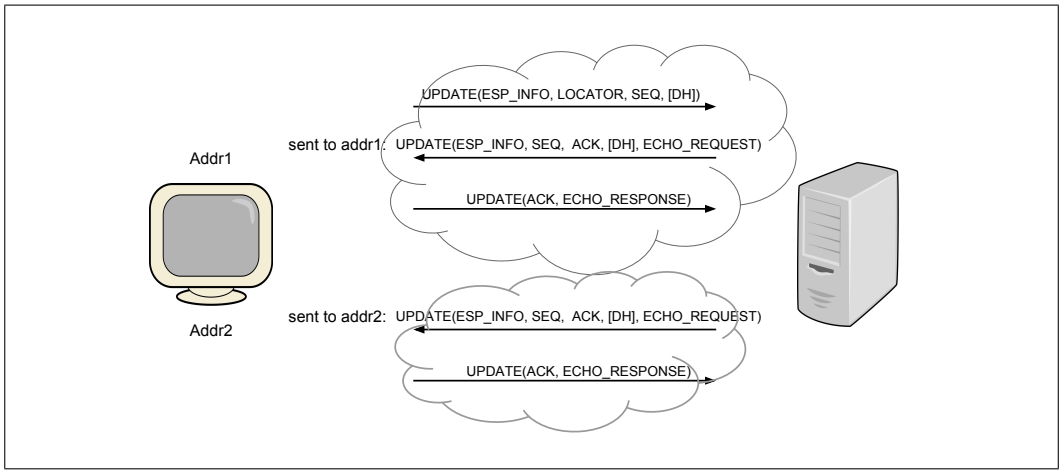

Figure 2.5: Multihoming with HIP

to its peer host. The peer host sends `UPDATE` packets to each address and updates corresponding SPIs.

2.4 DLMS / COSEM

Device Language Message Specification (DLMS) is an application layer specification. Companion Specification for Energy Metering (COSEM) presents an object oriented model for meters, providing a view of their functionality through communication interfaces. Every logical device has a world-wide unique identifier and holds certain information, which is modeled by interface objects. The information is organized in attributes and can be accessed through methods, depending on the access rights [23].

DLMS/COSEM neither supports the transmission of digital signatures with measurement data nor a firmware download. DLMS/COSEM includes authentication and confidentiality services based on symmetric encryption. The protocol does not allow the use of TLS/SSL which could realize these services with asymmetric keys [15].

2.5 Smart grid

Smart Grids are made possible by two-way communication technology and computer processing. Such integrated systems with feedback have been used for decades in other industries. These techniques are beginning to be used in electricity networks, starting with power plants (including wind farms) all the way to the consumers of electricity in homes and businesses.

Smart grids have the potential to offer large improvements in energy efficiency both within the electricity grid and in the premises of energy users', i.e., homes, offices, and factories. As shown in Figure 2.6 the basic topology of a smart grid consists of several SMs in a neighborhood or an apartment complex connected to DCU. Several such DCUs are connected to the DCS. The DCS and the utility company receives the data collected by the DCS.

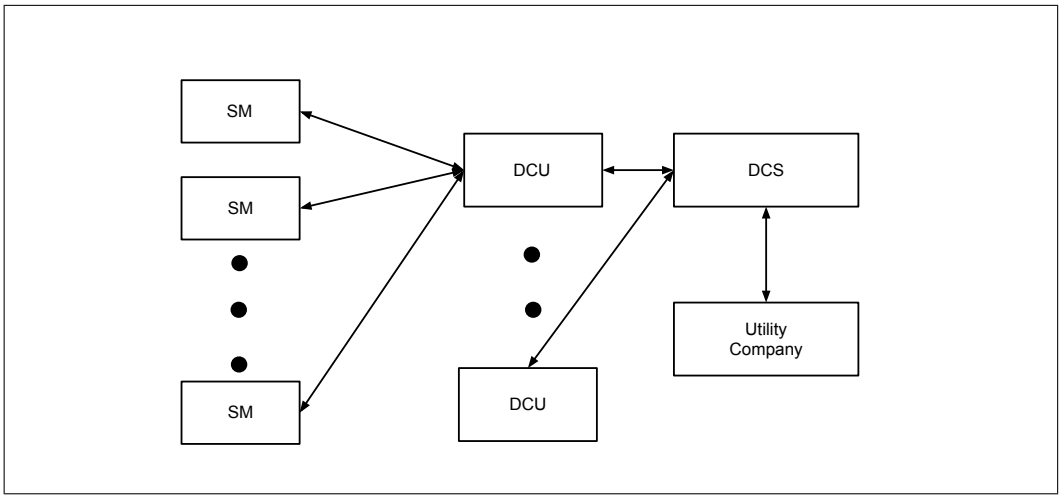

Figure 2.6: Smart Grid Topology

Traditionally utility companies (which combined generation and distribution of electricity) had to send meter readers out to gather the data needed to bill consumer for the electricity that they used. In addition to reading meters, employees of the electrical utility had to look for broken equipment and measure the voltage being delivered. Most of the devices utilities use to deliver electricity have yet to be automated and computerized. Today, many options and products are becoming available to the electricity industry enabling it to modernize.

Much in the way that these days a "smart" phone means a phone with a computer in it, smart grid means "computerizing" the electric utility grid. The most important change has been to add two-way digital communication technology to devices associated with the grid. Each device in the electrical distribution network can be given sensors to gather data (power meters, voltage sensors, fault detectors, etc.). The two-way communication enables digital communication between the device in the field and the utility's network operations center. A key feature of the smart grid is automation of the management and control of each individual device (potentially millions of devices) from a central location.

The number of applications that can be used on the smart grid, once suitable data communications technology is deployed, are growing rapidly

as companies create and produce these new applications. Benefits of smart grids include handling different sources of electricity, such as wind and solar power, and even integrating electric vehicles into the grid. The companies making smart grid technology or offering such services include technology giants, established communication firms, and even brand new technology firms.

A smart meter system employs several control devices, various sensors to identify and quantify parameters, and devices to transfer the data and command signals. In future electricity distribution grids, smart meters are expected to play an important role in monitoring the performance and the energy usage characteristics of the load on the grid. Collection of energy consumption data from all customers on a regular basis allows the utility companies to more efficiently manage electricity generation and also to advise their customers about the cost efficient ways to use their appliances. In light of this, smart meters can be used to control light, heat, air conditioning, and other appliances [10]. Smart meters can be programmed to maintain a schedule for operation of the home appliances and control operation of other devices accordingly. In addition, integration of smart meters helps utility companies in detecting unauthorized consumption and electricity (i.e., theft), while simultaneously improving their distribution efficiency and power quality [4].

2.5.1 Smart meter

A smart meter is an advanced energy meter that measures energy consumption and provides additional information to the utility company as compared to a regular energy meter. More specifically, in addition to providing real-time energy consumption, as SM can also provide voltage, phase angle, and frequency measurements. The SM needs to securely communicate all of this data to the utility company and the DCS. A smart meter allows the DCS to track how much electricity is used and, more importantly, what time of day that the energy was used. One goal of time of the day pricing is to raise awareness about the cost of power and another goal is to encourage consumers to reduce electricity use during times when the price is high. Furthermore, the ability of smart meters to communicate with the billing system enables the DCS to collect information regarding electricity supplied back into the power grid (e.g. produced by the roof top solar panels) from customer premises and to retrieve billing information automatically, thus eliminating the need for a human meter reader.

While not endless, the possibilities are many. Besides allowing the consumer to figure out the most economical time to run her dishwasher,

a smart meter could allow her to sell surplus electricity back to the grid. For example, if the home uses solar panels to help meet some of its electricity needs, then a long sunny spell might result in extra power being available during parts of the day, and she could sell that surplus power and reduce her electricity bill.

Smart meters can communicate and execute control commands remotely as well as locally. In other words, a smart meter may allow the DCS to disconnect or reconnect the home to the grid remotely. Such a disconnection could occur when the customer defaults on his or her electricity bill.

Smart meters bring the notion of comparison shopping to electricity as consumers could buy power from multiple suppliers. Additionally, there is a financial incentive to shift power usage away from periods of peak demand. When demand rises, power has to be imported from other regions/countries or generated using more expensive peak load power generators, usually at a much higher price per kilowatt than consumers are charged.

2.5.2 Data Concentration Unit

Several smart meters in a neighborhood or an apartment are connected to a data concentrator. The main goal of this thesis is to design and evaluate a secure and reliable way to communicate (both meter data and control signals) between individual SMs and the DCU. The DCU is then responsible for securely and reliably communicating the meter readings to the DCS and the utility. The communication of data between the DCU, DCS, and the utility can be considered as a future work based on this thesis project's results.

2.5.3 Distribution Control System

The distribution control system (DCS) is responsible for managing the distribution grid, collecting the meter data from all of its DCUs, billing the consumer, and the installation and maintenance of the SMs and DCUs. The DCS also provides data to the utility company. The DCS is assumed to be an authorized and secured entity and the public is assured of the safety of the data.

2.5.4 Utility company

The utility company is responsible for analyzing the consumption data obtain from the DCS in order to use this data for the efficient production of electricity. Note that the utility company's activities also includes estimating

future consumption, planning for new generating capacity, contracting to buy and sell power to other utilities, etc. Similar to the DCS, the utility company is also assumed to be authorized and secured entity and the protection of the consumers consumption data is guaranteed to be safe.

2.6 OPEN Meter Project

The Open Public Extended Network (OPEN) meter project is a EU project with the aim of specifying a comprehensive set of open and public standards for an Automated Metering Infrastructure (AMI) supporting electricity, gas, water, and heat metering, while taking into account the real conditions of the utility networks (so as to allow for a full implementation) [1].

The project is divided into the following six broad tasks:

1. Investigation of the functional requirements and regulatory issues concerning AMI in the various European countries.

2. Review of the state-of-the art of the various technologies available, including protocols for wired and wireless communication media.

3. Research and development activities to ensure that the requirements of AMI will be met in a cost effective manner.

4. Development of test approaches and procedures for laboratory, compliance, and field tests of the newly developed system elements.

5. Specification and proposal of a standard for AMI.

6. Dissemination of the project results to all the stakeholders, utilities, manufacturers, energy market participants, and end users.

The following subsections summarize the results of this project that are relevant to this thesis project.

2.6.1 Regulations for Germany

The OPEN meter project outlines the existing regulatory requirements on smart metering throughout the European countries [32]. This section highlights the regulatory requirements of Germany.

Metering Actors

Four companies dominate the electricity market. RWE, E.ON, Vattenfall, and EnBW control 90% of the generation and almost all of the transmission market. However, there are around 870 local distribution network operators. These big four represent over 50% of the retail electricity market.

Metering services are the responsibility of the DCS. The operational model places the responsibility for metering generally with the distribution businesses, although some of the larger utilities may use in-house operations for installation and reading meters.

Regulatory Framework

The regulatory authority in Germany is the Federal Network Agency for Electricity, Gas, Telecommunications, Posts, and Railway (BnetzA). For smaller utilities regulation is carried out by local (state) regulators.

Both gas and electricity markets were fully opened to competition in 1998. However, in both markets, there has not been a great deal of activity due mainly to the high level of vertical and horizontal integration in the energy markets, and the emergence/existence of a number of dominant participants. Residential switching (i.e., customers changing from one supplier to another) runs at about 5% for electricity, with gas switching numbers negligible.

Functional and technical requirements

Apart from the lack of legal direct technical requirements for smart metering, there are two main initiatives working on stating requirements: Open Metering and the Multi Utility Communications initiative.

Open Metering is a community of manufacturers of metering and related equipment and is supported by the associations FIGAWA [2], ZVEI [7], and KNX [4]. The goal of Open Metering is the promotion of open, cross-vendor devices and interface standards and their application. Open Metering has developed specifications for product compliance, i.e., a defined degree of functionality and interoperability. In this context interoperable communications interfaces of consumption meters are considered. The result of this work is the Open Metering System (OMS). In particular, a cross-media standardization is sought (that would support multi-utility metering).

A German initiative driven by utilities themselves is the Multi Utility Communication (MUC) initiative [6]. This activity started in the spring of 2007 and is being undertaken by companies in the utilities sector under the banner of a trade association - the BDEW [3].

2.6.2 Technological alternatives

The OPEN meter project studies the concepts, architectures, and state-of-the art wired and wireless communications technologies, protocols, and data models applicable to Automatic Meter Reading as part of an Advanced Metering Infrastructure [10].

Table 2.1 lists the various technologies evaluated by the OPEN meter project.

Table 2.1: Different technological alternatives

AMR Techniques	**Wireless Technologies**	**Wire-line Technologies**
Walk-by, Drive-by, Fixed networks & Hybrid networks	Wavenis, Plextex, Everblu, Bluetooth, WPAN (Zigbee, 6LoWPAN, PHY and MAC), WLAN, WiMax, GSM/GPRS/EDGE, UMTS, LTE, PMR, 2-way radio paging, European Radio Ripple Control, Satellite Systems.	PSTN, xDSL, FTTB, FTTH, M-Bus, PLC.

2.6.3 OPEN meter System Architecture

The overall system architecture was designed after the selection of suitable technologies required for AMI. Figure 2.7 illustrates the different system components and interfaces that define the system architecture of the OPEN meter project [9].

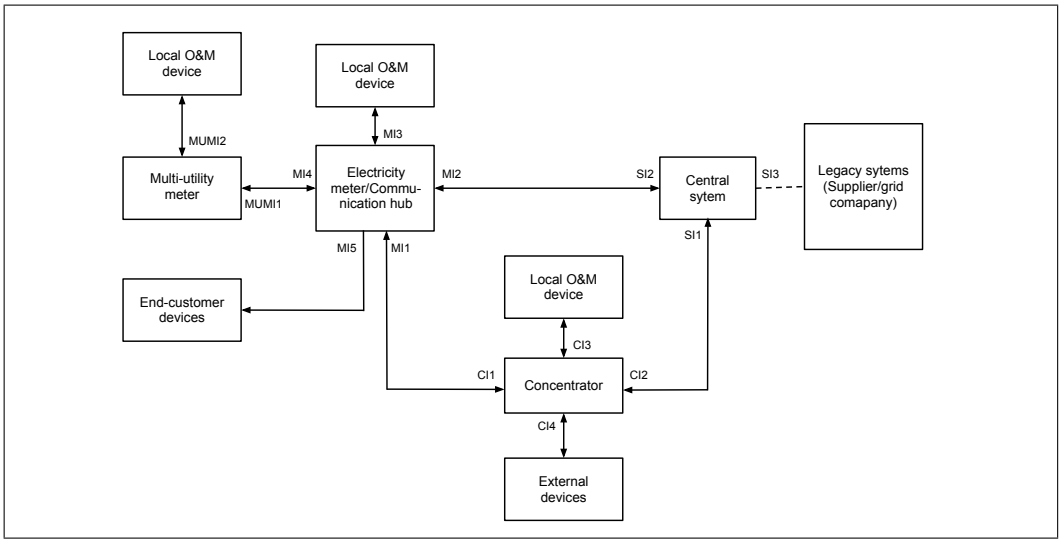

Figure 2.7: OPEN meter system architecture [9].

The electricity meter can act as a communication hub for other meters in the house, hence other meters could delegate certain power-intensive operations to the electricity meter such as cryptographic functions. The concentrator is need when the local network uses power line communications as a media. Local operation and maintenance (O&M) devices are used by the utility company's personnel to locally configure, operate, and maintain the meters and other electronic devices of the architecture. Table 2.2 lists the various technologies chosen for the different interfaces in the system architecture.

Table 2.2: Technologies for the interfaces

Interface	Selected Technology Type	Lower Layer protocols
MI1-CI1	PLC	Prime, IEC 61334-5-1
CI2-SI1	Wireless	UMTS, GPRS
MI2-SI2	Wireless	UMTS, GPRS
MI3, CI3 and MUMI2	Wireless	IEE802.15.4, IEE802.11-2007
MUMI1-MI4	Wireless	IEE802.15.4, IEE802.11-2007, Wireless M-Bus
CI4	Wireless	Zigbee, WiFi
MI5	Wireless	Bluetooth, Zigbee

Security is taken care by the DLMS/COSEM protocols. Device Language Message Specification (DLMS), is the suite of standards developed and maintained by the DLMS User Association and has been co-opted by the IEC TC13 WG14 into the IEC 62056 series of standards. Companion Specification for Energy Metering (COSEM), includes a set of specifications that defines the transport and application layers of the DLMS protocol. Details of DLMS and COSEM are given in section 2.4.

Chapter 3

Security

Security threats can be broadly classified into three main classes, depending on whether the system property being threatened is confidentiality, integrity, or availability. The protection schemes to counter these security threats involve a three step process: identification (the user says who she is), authentication (the system verifies the validity of this claim), and authorization (she is granted specific access rights).

3.1 BSI Protection Profile for the DCU

The Bundesamt für Sicherheit in der Infromationstechnik (BSI) (Federal Office for Information Security - Germany) has proposed a Protection Profile (PP) for the DCU of a Smart Metering System (SMS) [26]. An implicit SMS architecture is defined in order to provide an overall technical perspective of the DCU. The PP first, defines a problem statement listing the plausible security threats to the SMS, followed by the security objectives that mitigate these security threats. Furthermore, the PP defines the security requirements to be fulfilled by the DCU in order to achieve the desired security objectives. The security functionality of the DCU includes protection of confidentiality, authenticity, integrity of data, and information flow control.

As shown in Figure 3.1, the SMS is comprised of different functional units:

- Home Area Network (HAN): In-house data communication network which interconnects domestic equipment and can be used for energy management purposes.

- Metrological Area Network (MAN): In-house data communication network which interconnects metrological equipment and can be used for energy metering and management purposes.

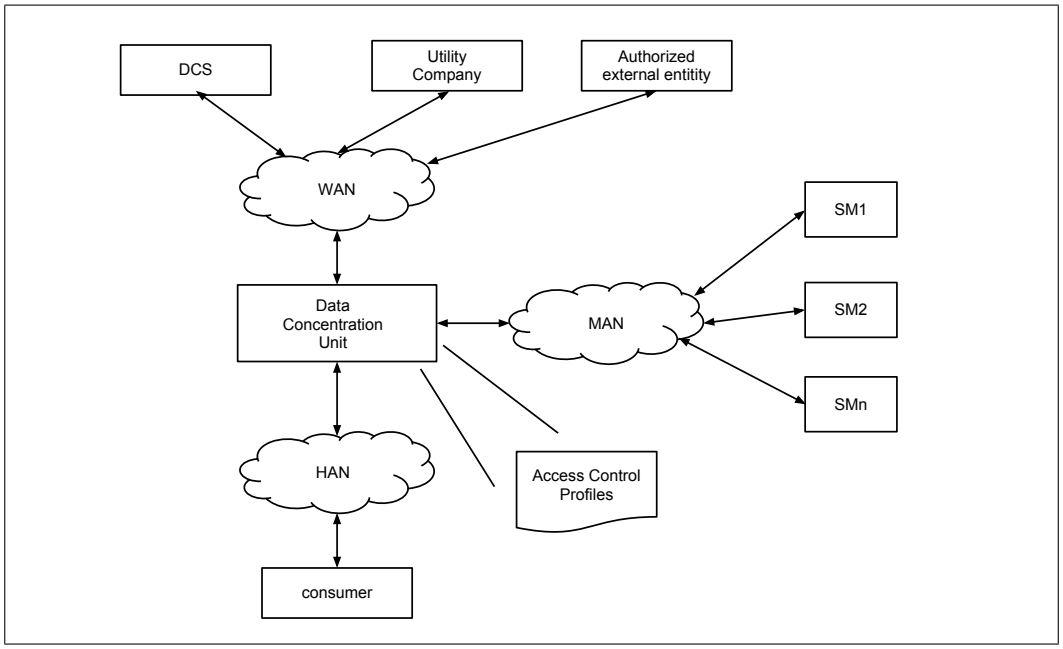

Figure 3.1: DCU as a part of the Smart Metering System [26]

- DCU: Device or unit responsible for collecting SM Data, processing this data, providing cryptographic primitives (with the help of a Security Module). The DCU utilizes access control profiles to determine which data shall be sent to which external entity.
- Access Control Profile: An access control profile defines:
  - how the SM data must be processed,
  - which processed SM data must be sent in which intervals,
  - to which external entity,
  - signed using which key material,
  - encrypted using which key material,
  - whether processed SM data be pseudonymised or not, and
  - which pseudonym shall be used to send the data.
- Security Module: The Security Module is a part of the DCU and provides cryptographic services and a secure storage for confidential assets.
- Smart Meter (SM_n): Device responsible for collecting consumption or production data of a commodity and transmitting this data to the

DCU. The SM has to be able to encrypt and sign the data it sends to the DCU, as the DCU and SM are physically separated.

- Wide Area Network (WAN): Extended data communication network connecting a large number of communication devices over a large geographical area.

In order to define the possible security threats associated with the SMS and the DCU in particular, the PP lists assumptions about the environment of the components in the threat model.

- The processing of any kind of private or billing related data by external entities (eg. DCS, utility company, etc.) is assumed to be trustworthy.
- The DCU admin is assumed to be trustworthy.
- The DCU is installed in a private premises of the consumer's house and thus is assumed to have a basic level of physical protection.
- The access control profiles are guaranteed to provide correct privileges to the external entities while they (the external entities) are assumed to properly handle the data.
- The software updates for the DCU are assumed to be well tested and certified by an authorized third party.
- The WAN network connection is assumed to be adequately reliable and provides sufficient bandwidth. SMs in the MAN communicate only with the DCU. In case of disjoint connections between the parties connected to the HAN and WAN, the connection is assumed to be suitably protected.

The threat model describes the threats for the communication between the SM and the DCU. The threat model takes into consideration two types of attackers: local attackers having physical access to SM, DCU, or a connection between these components and external attackers located in the WAN trying to compromise the confidentiality and/or integrity of the metering data and or configuration data transmitted via the WAN.

- A local attacker may try to alter, insert, replay, or redirect the metering data being transmitted between the SM and the DCU.

- An external attacker may try to modify metering data, modify the DCU configuration data, modify the SM configuration data, or tamper with a software update when it is being transmitted between the DCU and an external entity in the WAN.

- A local attacker or WAN attacker may try to alter the DCU's time.

- An external attacker may try to obtain control over DCU and/or SMs which enables the attacker to cause damage to devices attached to the consumers HAN or MAN (or devices controlled by the SMs), to external entities, or to grids used for distribution of the commodity (i.e., the electricity, gas, water, etc.).

- By physical and/or logical means a local attacker or an external attacker may try to extract historical data from the DCU, even though this data is no longer needed by the DCU.

- An external attacker or local attacker may try to access information to which they do not have permission to access or access it in a way that they do not have permission to do, while the information is stored in the DCU.

According to the PP, the following features must be implemented by the DCU in order to counter the threats defined above. Each threat can be mitigated using a combination of these features.

- Firewall: The DCU shall provide firewall functionality in order to protect the devices or units connected to the MAN and HAN against threats from the WAN side. The firewall shall:

  - allow only connections established from the internal network to the external network (i.e., from the devices in the HAN to the external entities in the WAN or from the DCU to the external entities in the WAN);
  - provide a wake-up service on the WAN side interface;
  - not allow any other services to be offered on the WAN side interface; and
  - enforce communication flows by allowing traffic from devices in the HAN to the WAN only if the three following aspects of security are achieved: confidentiality, integrity, and authentication.

- Separate Interface: The DCU shall have physically separated ports for the MAN, the HAN, and the WAN. The DCU shall automatically detect during its self test whether link connections, if any, are incorrectly connected.

- Concealing: To protect the privacy of the consumers, the DCU shall conceal the details of its communication with the external entities in the WAN in order to ensure that no privacy-relevant information may be obtained by analyzing the frequency, load, size, or the absence of external communication.

- Cryptography: The DCU shall provide the relevant cryptographic functionalities for secure handling of the data between the SM and the external entities in the WAN. The following functions will be supported by this cryptography:

  - authentication, integrity-protection, and encryption of all the communication between the DCU and all the entities in the WAN, MAN, and HAN.
  - replay detection for all communication with external entities.
  - encryption of the persistently stored user data stored within the DCU.

- Time stamp: The DCU shall provide reliable time stamps and update its internal clock at regular intervals by retrieving reliable time information from a dedicated reliable time source located in the WAN.

- Protection of security functionality: The DCU shall implement functionality to protect its security functions against malfunctions and tampering.

- Management: The DCU shall only provide authorized administrators with access to manage its security features. Furthermore, a secure method for software upgrade must be implemented by the DCU.

- Logging: The DCU shall maintain a set of log files (system log, consumer log, and billing log) and access to the information in these logs will be restricted by access control profiles.

- Access: The DCU shall control the access of users to information and functions via all of its external interfaces.

3.2 Key exchange

The key exchange problem concerns how to exchange whatever keys or other information are needed in such a manner that no one else learn this keying information. Identification of an entity is a non-trivial problem. In the case of asymmetric key cryptography it is possible to spoof another node's identity in several ways.

Three possible solutions to the problem of key exchange are:

- Diffie-Hellman key exchange (as used by IKE, HIP,),
- Public key infrastructure, or
- Web of trust.

Key management is an important part of any security system. The purpose of key management is to provide secure procedures for handling cryptographic keying material to be used in symmetric or asymmetric cryptographic mechanisms. The problem is establishing keying material whose origin, integrity, and - in the case of secret keys - confidentiality must be guaranteed.

For almost all systems, it is necessary to devise a means to distribute keys over the same communication channels that will be used for actual data.

In a in-network collaborative communication scheme: an AMI system can provide trust services, data privacy, and integrity based upon mutual authentication - whenever a smart meter joins the smart grid AMI network. This approach employs mutual authentications together with a remote authentication server and a neighboring smart meter as the authenticator in order to get the proper cryptography keys for subsequent secure data communications. The major weakness in this scheme is that the SMs are arranged in a tree structure, thus if a higher level SM loses connectivity then all the children SMs loose connectivity [36].

3.3 TLS

Transport Layer Security (TLS) is the next generation of the Secure Sockets Layer (SSL) [16]. TLS consists of two protocols: - the TLS record protocol and the TLS handshake protocol. The TLS record protocol is layered on top of a reliable transport protocol such as TCP, while the TLS handshake protocol is layered on top of the TLS Record protocol. The TLS handshake protocol provides connection security that authenticates the party or parties using asymmetric cryptography, negotiates a secret key, and

provides a reliable negotiation. The TLS record protocol is responsible for providing private reliable connections. The overall goals of the TLS protocol include cryptographic security, interoperability, and extensibility [12]. TLS is widely used together with HTTP to realize HTTPS.

3.3.1 TLS Record Protocol

A Message Authentication Code (MAC) is an authentication tag (also called a checksum) derived by applying an authentication scheme, together with a secret key, to a message [5]. The TLS Record Protocol takes messages to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, and transmits the result together with the authentication tag. The received data is decrypted, verified (by comparing the authentication tag with a locally computed MAC), decompressed, reassembled, and then delivered to a higher-level application [12].

To study the transformations of the data by the TLS record protocol, lets understand an example case. For this example lets assume that the communication end points have agreed on a block cipher for data encryption purposes. When an upper layer protocol passes data down to the record layer, a plain text record is created. TLS calls such records `TLSPlaintext`. The structure of a `TLSPlaintext` record is as shown:

```
struct {
   ContentType type;
   ProtocolVersion version;
   uint16 length;
   opaque fragment[TLSPlaintext.length];
} TLSPlaintext;
```

The `type` field classifies the data contained in `fragment` as either user data or data related to the TLS handshake protocol. The `version` variable specifies the version of the TLS. This discussion is based on TLS version 1.2 [12]. Multiple versions of TLS have incompatible record formats. The `length` field specifies the size in bytes of the payload contained in `fragment`. Maximum size imposed on `fragment` is 2^{14} bytes. Note that `TLSPlaintext` records are sent over only during the initial handshake procedure before the security parameters are negotiated by the communication end points.

TLS uses data compression to reduce the size of the record size. The compression algorithm to be used for this purpose is negotiated during the handshake phase. Thus `TLSPlaintext` records are subsequently transformed

into TLSCompressed records by compressing the fragment field. The length of the data fragment might decrease or increase depending on the data and the performance of the compression algorithm. The case that a compression algorithm increases the length is unlikely but still possible. The structure of a TLSCompressed record is as shown:

```
struct {
   ContentType type; /* same as TLSPlaintext.type */
   ProtocolVersion version;/* same as TLSPlaintext.version */
   uint16 length;
   opaque fragment[TLSCompressed.length];
} TLSCompressed;
```

Finally, this compressed data is protected using cryptographic algorithms. This cryptographically encoded record type is called TLSCiphertext and is as shown:

```
struct {
   ContentType type;
   ProtocolVersion version;
   uint16 length;
   select (SecurityParameters.cipher_type) {
      case stream:  GenericStreamCipher;
      case block:  GenericBlockCipher;
      case aead:  GenericAEADCipher;
   } fragment;
} TLSCiphertext;
```

Although the exact encoding differs depending on the type of cryptographic algorithm selected. However for the sake of this example let us assume that a CBC block cipher algorithm has been selected:

```
   struct {
   opaque IV[SecurityParameters.record_iv_length];
   block-ciphered struct {
      opaque content[TLSCompressed.length];
      opaque MAC[SecurityParameters.mac_length];
      uint8 padding[GenericBlockCipher.padding_length];
      uint8 padding_length;
   };
} GenericBlockCipher
```

The data that is actually encrypted is the concatenation of the four fields inside the `struct` that is marked with `block-ciphered`. The padding of the data is necessary as CBC can only encrypt chunks of data with a length that is an integral multiple of block length. A Message Authentication Code (MAC) is also appended to the content. According to RFC 5246 [13] the MAC is calculate using `type`, `version`, `length` and `fragment` field of the `TLSCompressed` record along with an *implicit sequence number*:

```
MAC(MAC_write_key, seq_num +
TLSCompressed.type +
TLSCompressed.version +
TLSCompressed.length +
TLSCompressed.fragment);
```

TLS operates on top of a reliable transport protocol (usually TCP) which ensures in-sequence delivery of the records, thus the sequence number of a record is implicit with the sequence number of the first record, after the handshake, being zero. Although the sequence number of the subsequent records can be easily determined, the sequence number is still included in the calculation of the MAC in order to mitigate the replay and re-ordering attacks.

In the case of stream cipher, the encoding rules are as defined:

```
stream-ciphered struct {
   opaque content[TLSCompressed.length];
   opaque MAC[SecurityParameters.mac_length];
} GenericStreamCipher;
```

The MAC is calculated in a similar way as in the block cipher case. However, the encryption procedure works differently. A stream cipher has no block size and hence no padding is required. In the case of block ciphers the state is maintained across the TLS records, i.e., the records have to decrypted in the exact order in which they were encrypted. In the case of a stream cipher the sate must be explicitly saved after a record is decrypted and this state has to be restored when the next record needs to be decrypted.

3.3.2 TLS Handshake Protocol

The TLS handshake protocol allows two communicating end points to automatically negotiate security parameters in a secure fashion. The result

of this process is also referred to as Security Association (SA).

Besides establishing a shared secret, the handshaking protocols are also responsible for authenticating the peers. This is usually achieved by employing certificates that were signed by a trusted Certificate Authority (CA). This process is crucial for secure communication. The parties involved need to be sure of the fact that they are indeed talking to the entity they think they are talking to.

Figure 3.2 indicates the message flow for a full TLS handshake. The overall handshake procedure can be broken down into the following broad steps [12]:

- Exchange of hello messages to agree on algorithms, exchange of random values, and a check for session resumption.
- Exchange the necessary cryptographic parameters to allow the client and server to authenticate themselves (if mutual authentication is to be done).
- Exchange certificates and cryptographic information to allow the client and server to authenticate themselves.
- Generate a master secret from the pre-master secret and exchanged random values.
- Provide security parameters to the record layer.
- Allow the client and server to verify that their peer has calculated the same security parameters and that the handshake occurred without tampering by an attacker.

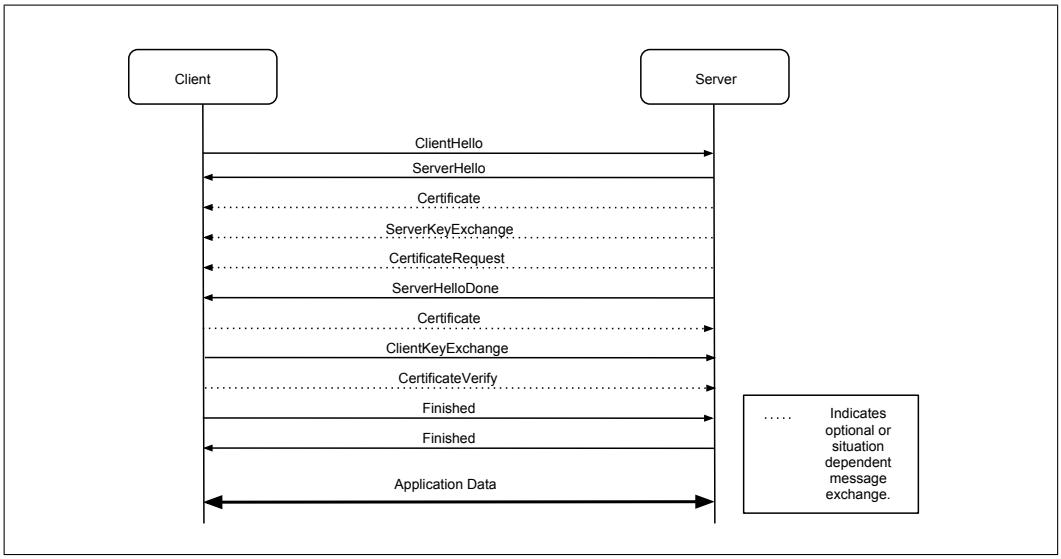

Figure 3.2: TLS Handshake

Handshakes can take place at different points in the lifetime of a TLS connection. RFC 5246 [13] states that the first handshake must be initiated right after the transport connection has been established i.e. right after TCP's three-way handshake has been completed. The reason is that the TLS record protocol must not transmit user data unprotected.

To ensure better security cipher algorithms should be periodically renewed after a certain time or after a certain amount of data has been encrypted, whichever comes first. For this reason, TLS allows both peers to initiate a re-negotiation of the cryptographic parameters. It should be pointed out that the sequence number used by the TLS record protocol is reset to zero after every (re-)handshake.

3.4 DTLS

DTLS is a modified version TLS that functions properly over datagram transport [28]. The target applications for DTLS are primarily of the client server variety. Almost all of the protocol elements of TLS are reused in the design of DTLS, however DTLS has minor but important modifications for it to work properly with datagram transport.

3.4.1 DTLS Record Protocol

All DTLS data is carried in records similar to TLS. DTLS records are required to fit within a single datagram to avoid fragmentation. This

requirement provides three advantages of DTLS over TLS. First, since the DTLS layer does not need to buffer partial records, host memory can be used more efficiently, which makes the host less susceptible to a DoS attack. Second, it is quite possible that datagrams carrying the remaining record fragments are lost, in which case the received fragments are useless and cannot be processed. Third, it is not clear how long received fragments should be buffered before being discarded.

DTLS endpoints use epoch numbers in the record format to determine which cipher state has been used to protect the record payload. Epoch numbers help resolve any ambiguity that arises when data loss occurs during a session renegotiation. For example, consider a client transmitting data records 2, 3, and 4, followed by *ChangeCipherSpec* message in record 5. Suppose the server receives records 2 and 4 (3 and 5 are lost). From the server's point of view, record 3 could have been the *ChangeCipherSpec* message, in which case record 4 is (incorrectly) assumed to be associated with the pending cipher state. Since epoch numbers are incremented upon sending a *ChangeCipherSpec* message, the server can use the epoch number to resolve the ambiguity. In this case, records 2 and 4 have the same epoch, implying that record 3 must have been a data record. The encoding rules for a DTLS record are as shown:

```
struct {
   ContentType type;
   ProtocolVersion version;
   uint16 epoch; //New field
   uint48 sequence_number; //New field
   uint16 length;
   select (CipherSpec.cipher_type) {
      case block:  GenericBlockCipher;
      case aead:  GenericAEADCipher;
   } fragment;
} DTLSCiphertext;
```

The epoch number and the sequence number are used in a hierarchy such that the sequence number is unique only in combination with an epoch number. With every rehandshake the epoch number is incremented by one and the sequence number is reset to zero. This means, effectively, that epoch numbers map to security parameters including the keys used by the bulk cipher algorithm.

The purpose of the sequence numbers is to defend against replay attacks. Records carrying sequence numbers that have been dealt with already are

discarded. Re-ordering is not prevented since it is not necessarily malicious but an effect that can occur even in the absence of an attacker. DTLS implementations do not restore the order of DTLS records if they arrive out of order as this would result in head-of-line blocking or even deadlock.

One of the requirements of DTLS is that a record should fit entirely within a single datagram. Thus DTLS records are often smaller than TLS records. The largest packet that can be transmitted between two hosts - the Path Maximum Transmission Unit (PMTU) - is typically less than the maximum of a TLS record.

3.4.2 DTLS Handshake Protocol

The DTLS handshake, shown in Figure 3.3, is nearly identical to that of TLS. There are two major changes:

1. Stateless cookie exchange to prevent denial of service.

2. Message fragmentation and re-assembly.

These changes are necessary as DTLS operates on datagram transport protocols such as UDP.

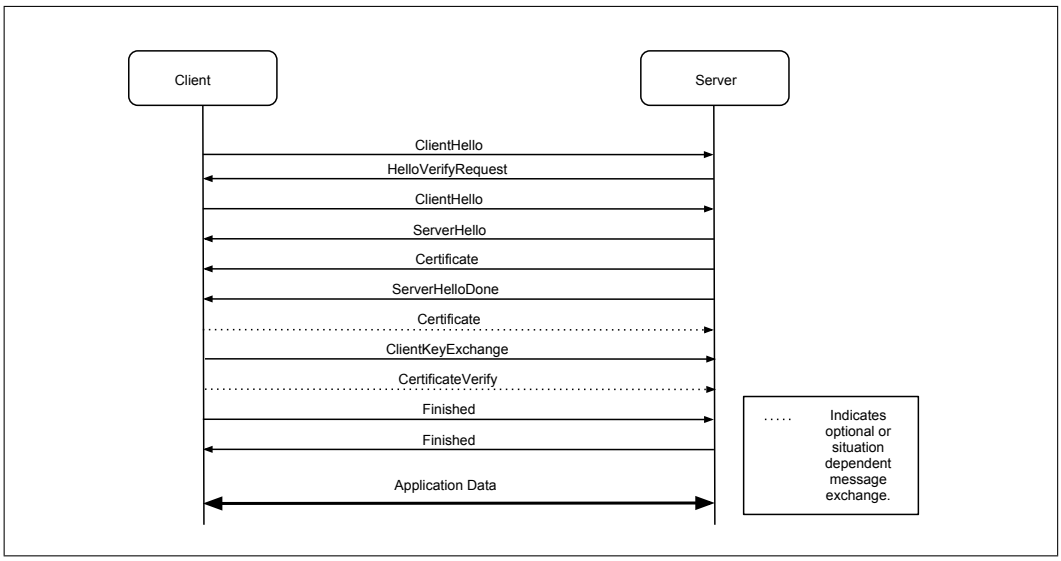

Figure 3.3: DTLS Handshake

DTLS handshake uses the *cookie exchange technique* to mitigate the standard resource consumption attack and the amplification attack. These attacks are described in [28]. In the *cookie exchange technique* the client

must replay a "cookie" provided by the server in order to demonstrate that it is capable of receiving packets at its claimed IP address.

DTLS handshake messages require a mechanism for retransmission as these messages could be lost. A single timer is used for the retransmission of the lost messages at each end-point. Each end-point keeps retransmitting its last message until a reply is received. Deciding the exact value for the time is tricky because the peer is often doing some kind of cryptographic computation, which can take a substantial fraction of the round trip time. Modadugu, et al. [28] recommends timer values between 500ms to 1000ms.

3.5 Summary

TLS is used on top of a connection oriented transport protocol such as TCP and DTLS is the adapted version of TLS for operation on top a datagram transport protocol such as UDP. UDP is chosen over TCP as the transport protocol for the communication between the SM and DCU because UDP is light weight and unlike TCP, UDP does not require the maintenance of any state at the communication end points. DCU and the SMs use DTLS to secure the communication between them. SM acts as a DTLS Server while the DCU acts as a DTLS client. This is because the SM is the source of the data (metering information) and the DCU is the receiver of this information. DTLS uses public-private key pair for negotiating a common secret and this secret is then used to encrypt the communication between the SM and the DCU. Thus DTLS uses asymmetric cryptography for negotiating a symmetric key, which is in turn used for symmetric cryptography of the communication data.

The cryptographic requirements set out by the BSI PP, on the DCU and the SM are fulfilled by DTLS. Authentication, integrity-protection and encryption of the metering data is provided by the inherent cryptographic ciphers of DTLS. These cryptographic ciphers are exchanged between the DCU and the SM during the handshake phase of the DTLS connection establishment.

The requirement for replay detection is fulfilled by the use of Record Sequence Numbers (RSN) in DTLS records. Unlike TLS, RSNs are explicitly specified in DTLS records since these records can get lost or be delivered out of order. Replay detection is performed using the replay window mechanism of RFC 2401 [22]. If datagrams always arrived in order, it would be sufficient for a DTLS end point to keep track of the most recent record seen in order to detect replays. But since datagrams may also arrive out of order, a replay window mechanism is required. This is most easily implemented as a bitmap

where the set bits represent the most recently received records. RSNs that are too old to be checked against the bitmap are discarded.

All the communication use cases described in the following chapter are assumed to be secured. The communication between the DCU and the SM is secured with the help of DTLS. Before the start of any of the communication use case it is assumed that the DTLS handshake has been successful and that the communication following this handshake is assumed to be secure. Key management is taken care by the DTLS handshake and it is a symmetric key pair that secures the communication of the data between the DCU and the SM.

Chapter 4

Communication Use Cases

This chapter lists and describes a number of use cases regarding to communication of data between the SM and DCU [35].

4.1 Obtaining meter readings

The smart metering system (SMS) measures and records consumption of units of some commodity and provides this information in the form of readings. This use case describes how the SMS provides this metering data to the data recipient. This metering data may be routinely sent to the data recipient as per a defined schedule (an example of this would be periodic readings), at specific time intervals, or according to some usage profile. Another variant of this use case would be that the eventual data recipient can request and obtain metering data on demand from the SMS, for example, where a reading is required to assist with resolving a billing enquiry. All of these alternatives can be broadly classified into two sub-cases:

1. Scheduled readings:
    - The SM locally registers metering data together with a time stamp (as a side effect it will record the timestamped information into the appropriate log file(s)).
    - The SM transfers this timestamped data to the DCU.
2. DCU requested meter reading:
    - The DCU sends a request for a meter reading to the SM.
    - The SM validates this request.

- If the request is valid, then the SM registers the latest metering data with a time stamp and transfers the timestamped data to the DCU.
- The DCU logs the status of the operation (success or failure) along with additional data such as time and the error response from the SM in the case of a failure.

4.2 Install, configure, and manage a SM

This use case describes how an SM is installed at a consumer's premises and the initial connection established between the SM and the DCU. This use case also describes a scenario where in an upgrade of the SM's firmware is to be performed as part of managing the SM.

Installation:

- A representative of the DCS arranges to visit the consumer's premise at a pre-arranged time with the relevant metering equipment.
- The DCS representative installs a SM for the first time or replaces existing metering equipment with a (possibly new) SM.
- The SM, once installed, contacts the DCS to self-register itself with the DCS.
- The SM informs the DCS of any required information at start up (e.g. initial meter readings, technical information about the SM, etc.). The SM then requests any required updates, applying each of the relevant updates.
- The installation is complete and the representative of the DCS is informed.

Configuration:

- The DCS sends configuration settings to the SM.
- The SM validates the message. The SM may also check the message against an access control list to see if it should process the requested operation for this source. If not, it should return an error message, otherwise it proceeds to the next step.
- The SM applies or updates the indicated settings.

- The SM confirms to the DCS that it has applied the settings.

Maintenance:

- If the SM identifies that a part of it has either reached the end of its operating life or if the SM detects that there is a malfunction, then the SM requires maintenance. Additionally, the DCS can indicate that it wishes to update the SM's software.

- In the case of end of life or malfunction, the SM alerts the DCS to its maintenance requirement(s). In the case of a DCS initiated software update, the DCS informs the SM that a software update is to be performed.

- When the DCS receives an alert it assesses whether maintenance requirements necessitates a software or hardware update or perhaps even requires an on-site visit. The DCS takes the appropriate action to maintain and manage the SM.

- If the SM receives an update from the DCS, it applies the update and sends a confirmation of this update to the DCS.

- If there had been an alert that leads to maintenance, then the SM re-runs the original diagnostics that identified the need for maintenance. If the maintenance requirement is resolved, then nothing more happens. If the maintenance requirements have not been resolved, then a new alert is generate to the DCS and the diagnosis and repair procedure begins again (note that the DCS may limit the number of times this procedure occurs and may place the SM into some known state and schedule a technician to repair or replace the SM).

4.3 Remotely Enable & Disable the SM

This use case describes how an SM is intentionally disconnected in case of a failing to pay or another illegal activity being detected.

- The DCS sends an instruction to the SM to enable or disable the supply of the commodity or to limit the availability of the commodity at the consumer's premise.

- The SM validates the request (again checking the access control list to make sure that this source is allowed to make this request).

- The SM carries out its instruction(s).
- The supply is enabled or disabled or the availability of the commodity is limited.
- The SM sends a message to the DCS confirming its instructions has been carried out.

4.4 Display a Message

On several occasions there may be a need to inform the consumer (or in the case of premises based generation - the local generator of the commodity), the DCS, or a utility company. This message might even come from an authorized third party. This use case describes a scenario regarding the display of such messages on the SM. (Here we have assumed that the SM meets the requirements for a DCU in a SMS as described in [26] - thus it has a display.)

- The DCS, utility company, or authorized third party sends an instruction to the SM to display a message on the SM.
- The SM validates the request and performs the access control operations to ascertain if it should perform this instruction for this source.
- The SM displays the message.
- If the SM features some means for a human to responds to this message, perhaps with a yes/no response, then the SM sends this response along with a confirmation of having displayed the message, otherwise the SM simply confirms that it has displayed the message.
- The SM sends a message to the DCS and/or the utility company confirming the consumer's response (if any).

4.5 Manage Tariff Settings in a Smart Metering System

This use case describes how the SMS can utilize tariff information, for example providing a unit price for different quantities as a function of time as per a predefined schedule or as a result of *ad hoc* notifications from the DCS. The later might occur when there is a need for a controlled brownout or other change in service outside of the normal service.

- The DCS sends an instruction to the SM to apply a new/updated tariff scheme.
- The SM validates the request and checks the access control list to see if this source is allowed to provide tariff rates.
- The SM applies the new/updated tariff as instructed when it computes new prices.
- The SM's display could display information about the new tariff scheme. The consumer could then save or deletes this message.
- The SM sends a message to the source of the new tariff confirming that the new/updated tariff scheme has been applied. In the case of a displayed message which requires a human response, the SM might also send this response. For example, this might be used to indicate that the human has read the message about the new tariff scheme or that the human wants to accept or not accept the new tariff.

Chapter 5

Analysis

The performance of DTLS in various scenarios is analyzed in this chapter. Section 5.1 analyzes the performance of DTLS over TLS, while section 5.2 analyzes the security attributes of DTLS while providing results of the security analysis of DTLS in specific scenarios.

5.1 Performance

Modadugu, et al. [28] have compared the network traffic generated by TLS and DTLS. Their implementation of DTLS is based on the `openssl` library as it was found to provide acceptable performance and is relatively easy to program. The results listed in Tables 5.1 and 5.2 are based on their tests in which the cipher negotiated was `EDH-RSA-DES-CBC3-SHA`. In the DTLS negotiation, each DTLS handshake message fragment has 25 bytes of overhead due to headers (13 for the record header and 12 for the message fragment), compared to 9 bytes for TLS. Table 5.1 shows the number of bytes transferred over a link with a PMTU of 1500 bytes and with a certificate size of 562 bytes. Table 5.2 shows the number of bytes transferred over a link with a PMTU of 1500 bytes with a certificate size of 1671 bytes. These results are only for the handshake phase. We can see that in both cases DTLS transfers more bytes that TLS (respectively 35% and 18% more bytes). With a larger certificate the relative difference is reduced.

Table 5.1: Bytes transferred with PMTU 1500, certificate size of 562 bytes [28].

	DTLS	TLS
client	446 bytes	228 bytes
server	1015 bytes	857 bytes
Total	1461 bytes	1085 bytes

Table 5.2: Bytes transferred with PMTU 1500, certificate size of 1671 bytes [28].

	DTLS	TLS
client	446 bytes	228 bytes
server	2313 bytes	2105 bytes
Total	2759 bytes	2333 bytes

With respect to latency their results show that for a handshake TLS beats DTLS by a very narrow margin. As shown in the performance results in [28] a DTLS handshake takes 42.9 ms, while the TLS handshake takes 41.5 ms. This narrow margin is due to the fact that DTLS handshake includes once extra round trip time for cookie exchange. It should be noted that these measurements do not include the time taken for TCP connection establishment.

Latency in DTLS is also caused by retransmissions of packets in the event of a packet loss. DTLS uses a retransmission timer to handle packet loss. In other words, a packet is retransmitted if a response to the packet is not received before the expiration of the retransmission timer. Thus care must be taken to choose suitable timeout values. With the current API (`openssl`), it is non-trivial to implement adaptive timers and hence implementations generally use fixed interval timers [33].

Dreibholz, et al. have described their design and implementation of SCTP aware DTLS in detail in [14]. Performance measurements have proven that an optimized SCTP-aware DTLS can be almost competitive to TLS over TCP.

The design of DTLS is closest to that of IPsec because a number of techniques are borrowed from IPsec. These techniques were incorporated into the design of DTLS to make the DTLS records safe. However, DTLS

differs from IPsec in two major respects. First, DTLS is an application layer protocol rather than a network layer protocol. Second, DTLS uses the familiar TLS programming model in which security contexts are application controlled and have a one-to-one relationship with communication channels.

5.2 Security

It should be noted that DTLS does not offer any improvements over TLS and DTLS does not reveal any additional information beyond that revealed by TLS during the handshake or bulk transfer phase. An attacker, by observing DTLS handshake records being exchanged during an established session, may identify the current epoch and sequence number. This is the only information revealed by the DTLS record layer, however this is public information to an attacker monitoring a TLS session. Thus this information does not cause any addition loopholes in the security of DTLS.

The DTLS Handshake messages reveal message number, fragment length, and fragment offset. This is again public information to an attacker monitoring a TLS session. On the other hand handshake messages exchanged due to session renegotiation are completely encrypted in both DTLS and TLS.

The processing of DTLS records and messages are identical to the processing procedure of TLS. In other words records and handshake messages are not processed until available in entirety in both DTLS and TLS.However, DTLS transmit processing leaks a small amount of timing information when compared to TLS. DTLS packets have the potential to reveal information about the plain text under certain circumstances [34]. A plain text recovery attack on DTLS is presented in [8]. TLS overcomes this problem because it uses TCP, as TCP congestion and flow control hides this information. A solution to this problem could be to use buffered writes [28].

The prototype implementation in [25] shows that a DTLS handshake with strong security parameters is feasible for key establishment even with power constrained devices. This prototype is a proof of concept implementation that supports a client and server authenticated DTLS handshake using a Trusted Platform Module (TPM) for executing the RSA key exchange algorithm. This prototype uses OpenSSL 1.0.0d with the padding for RSA signature verification changed from PKCS#1 version 1.5 to version 2 on the server side. The client only has to sign a SHA1 hash instead of the concatenation of a MD5 and SHA1 hash. These changes were made to maintain compatibility with the TPM hardware. Tables 5.3 and 5.4 shows the average time over ten measurements that were needed to establish a DTLS

connection when using 1024 and 2048 bit RSA keys for server and certificate authority X.509 certificates. The "Drop Rate" column species the chance for a packet to be lost in the link layer. Note that the measurements were take by introducing a 500ms delay between sending two DTLS handshake messages.

Table 5.3: Connection latency over a lossy link (ms) with RSA key size of 1024 bits [25].

Drop Rate	Min.	Avg.	Max.
0%	5,789	5,851	5,938
5%	5,835	16,435	27,592
10%	11,045	51,973	171,925

Table 5.4: Connection latency over a lossy link (ms) with RSA key size of 2048 bits [25].

Drop Rate	Min.	Avg.	Max.
0%	6,861	6,949	7,065
5%	16,600	26,945	39,680
10%	21,706	37,386	52,443

Furthermore, energy calculations are done which shows that the energy consumption of a successful handshake with 2048-bit RSA keys and without packet loss is 579 mJ [25]. The computation energy is the amount of energy spent for parsing the received certificates, hashing each handshake message, and computing the HMAC for the last message as well as encrypting it.

A novel and efficient approach to provide a strong security for UDP communications in vehicular networks is presented in the implementation of Vehicular DTLS (VDTLS) [31]. By extending the DTLS, VDTLS integrates a new-breed of public key cryptography, called Identity Based Encryption (IBE). VDTLS achieves significant bandwidth savings and eliminates overheads associated with traditional certificate management while still providing proven security services to UDP communications.

TLS and DTLS are both transport layer security protocols and both the protocols are very similar to each other except that TLS is designed to operate on top of a reliable communication protocol such as TCP, whereas

DTLS is designed for operating on top of a datagram transport protocol such as UDP. Both protocols use asymmetric cryptography for key exchange and symmetric cryptography for privacy.

DTLS is a generic channel security protocol designed for use in datagram environments. DTLS is based on the well understood TLS protocol and like TLS it is designed to provide a secure channel that mimics TLS. The semantics of TLS is session based whereas DTLS is epoch based (as sessions are not possible with datagram communication protocols).

DTLS uses the pre-existing protocol infrastructure and implementations can be reused. DTLS provides a familiar interface to a generic security layer, hence it is easy to adapt protocols to use it [28]. DTLS is able to complete key negotiation and bulk data transfer over a single channel.

Chapter 6

Conclusion

In chapter 2 several communication protocols suitable for the smart grid communication were discussed. PLC is a nascent technology and standardization of this technology is currently ongoing. During early stages of the thesis project HIP was evaluated to be a suitable communication protocol between the SM and the DCU. However, HIP was later dropped because it was considered overkill to provide HIP functionality in both the DCU and SM. DLMS /COSEM provides an overall communication technology encompassing all the actors in the smart grid.

In chapter 3, the requirements set out by the BSI Protection profile for the DCU were discussed. Also the security concepts such as key exchange were discussed. Finally TLS and DTLS are discussed in this chapter and it is argued that DTLS is a better solution for providing a secure communication between the SM and the DCU. How the security features provided by DTLS relate to the security requirements set out by the BSI PP was also discussed.

In chapter 4, five simple use cases related to the communication between the SM and the DCU are discussed. These use cases are just a starting point for further broadening of the communication use cases possible between the SM and the DCU.

Finally, in chapter 5, a survey of the work done regarding performance analysis of DTLS is provided along with the results of these works. Furthermore, this chapter surveys the work done relating to the performance of DTLS with respect to the its various security features.

More communication protocols should be surveyed and the security requirements should be broadened and standardized. Implementation of a prototype of the communication between the SM and DCU based on DTLS could be done to produce an accurate analysis of DTLS particularly with regards to the communication between the SM and the DCU.

Chapter 7

Future Work

This thesis should be seen as a starting point for carrying out further research into smart grid communication and in particular to the communication between the SM and the DCU. The security requirements for smart grid communication should be broadened and the BSI PP should be taken as a starting point for this. Furthermore, communication protocols should be evaluated along with the security protocols. A holistic approach should be taken to design an efficient network topology while keep in mind the security requirements for the smart grid.

A proof of concept implementation should be carried out to evaluate the performance of DTLS in the smart grid communication environment. Also a thorough evaluation on this proof of concept should be carried out to check the security features of DTLS and how those features relate to the security requirements set out by BSI.

Bibliography

[1] OPEN meter project. http://www.openmeter.com, 2011.

[2] Bundesvereinigung der Firmen im Gas- und Wasserfach e.V. http://www.figawa.de/, 2012.

[3] German Energy and Water Association. http://www.bdew.de, 2012.

[4] KNX. http://www.knx.org/, 2012.

[5] Message Authentication Codes - RSA Labs. Available from: http://www.rsa.com/rsalabs/node.asp?id=2177, 2012.

[6] Multi Utility Communication. http://www.m-u-c.org, 2012.

[7] Zentralverband Elektrotechnik- und Elektronikindustrie e.V. http://www.zvei.org/, 2012.

[8] N.J. AlFardan and K.G. Paterson. Plaintext-Recovery Attacks Against Datagram TLS. 2012. Available from: http://www.isg.rhul.ac.uk/~kp/dtls.pdf.

[9] A. Ankou et al. OPEN Meter Project - Design of the overall system architecture. Available from: http://www.openmeter.com/files/deliverables/Open%20Meter_D3%201_Architecture_v6_.pdf, 2010.

[10] M. Bittner et al. OPEN Meter Project - General overview of state-of-the-art technological alternatives, 2009. Available from: http://www.openmeter.com/files/deliverables/OPEN-Meter%20WP2%20D2.1%20part1%20v3.0.pdf.

[11] C. Chauvenet, B. Tourancheau, D. Genon-Catalot, P.-E. Goudet, and M. Pouillot. A Communication Stack over PLC for Multi Physical Layer IPv6 Networking. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 250 –255, Oct. 2010.

[12] D. Eastlake 3rd. Transport layer security (TLS) extensions: Extension definitions. *Internet Request for Comments*, RFC 6066 (Proposed Standard), January 2011. Available from: `http://www.ietf.org/rfc/rfc6066.txt`.

[13] T. Dierks and E. Rescorla. The transport layer security (TLS) protocol version 1.2. *IETF, RFC 5246*, 2008. Available from: `http://www.ietf.org/rfc/rfc5246.txt`.

[14] T. Dreibholz, E.P. Rathgeb, I. Rüngeler, R. Seggelmann, M. Tüxen, and R.R. Stewart. Stream control transmission protocol: Past, current, and future standardization activities. *Communications Magazine, IEEE*, 49(4):82 –88, April 2011.

[15] S. Feuerhahn, M. Zillgith, C. Wittwer, and C. Wietfeld. Comparison of the communication protocols dlms/cosem, sml and iec 61850 for smart metering applications. In *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pages 410 –415, oct. 2011.

[16] A. Freier, P. Karlton, and P. Kocher. The Secure Sockets Layer (SSL) Protocol Version 3.0. *IETF, RFC 6101*, 2011. Available from: `http://www.ietf.org/rfc/rfc6101.txt`.

[17] Ingvar Fröroth. *More than power down the line*. Licentiate thesis, KTH, Teleinformatics, 1999.

[18] S. Galli, A. Scaglione, and Zhifang Wang. For the grid and through the grid: The role of power line communications in the smart grid. *Proceedings of the IEEE*, 99(6):998 –1027, Oct. 2011.

[19] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). *IETF, RFC 2409*, 1998. Available from: `http://www.ietf.org/rfc/rfc2409.txt`.

[20] IEEE. 802.15.4. *IEEE Computer Society*, 2003. Available from: `http://standards.ieee.org/getieee802/download/802.15.4d-2009.pdf`.

[21] Ed J. Sermersheim. Lightweight Directory Acess Protocol (LDAP): The Protocol. *IETF, RFC 4511*, 2006. Available from: `http://www.ietf.org/rfc/rfc4511.txt`.

[22] S. Kent and K. Seo. Security Architecture for the Internet Protocol. *IETF, RFC 4301*, 2005. Available from: `http://www.ietf.org/rfc/rfc4301.txt`.

[23] T. Khalifa, K. Naik, and A. Nayak. A survey of communication protocols for automatic meter reading applications. *Communications Surveys Tutorials, IEEE*, 13(2):168 –182, quarter 2011.

[24] E. Kim, D. Kaspar, N. Chevrollier, and J.P. Vasseur. Design and Application Spaces for 6LoWPANs. draft-ietf-6lowpan-usecases-09, January 2011.

[25] T. Kothmayr, W. Hu, C. Schmitt, M. Brünig, and G. Carle. Poster: Securing the Internet of Things with DTLS. 2011. Available from: `http://www.cse.unsw.edu.au/~wenh/Kothmayr_sensys11.pdf`.

[26] H. Kreytzmann, N. Tekampe, and A. Abromeit. Protection profile for the gateway of a smart metering system 0.9.2 draft. Technical report, Federal Office for Information Security, Germany(BSI), 2011. Available from: `https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/SmartMeter/PP-SmartMeter.pdf?__blob=publicationFile`.

[27] N. Kushalnagar, G. Montenegro, and C. Schumacher. IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANS): Overview, Assumptions, Problem Statement, and Goals. *IETF, RFC 4919*, 2007. Available from: `http://www.ietf.org/rfc/rfc4919.txt`.

[28] N. Modadugu and E. Rescorla. The Design and Implementation of Datagram TLS. 2002. Available from: `http://crypto.stanford.edu/~nagendra/papers/dtls.pdf`.

[29] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin. Private memoirs of a smart meter. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, BuildSys '10, pages 61–66, New York, NY, USA, 2010. ACM.

[30] R. Moskowitz and P. Nikander. Host Identity Protocol (HIP). *IETF, RFC 5201*, 2008. Available from: `http://www.ietf.org/rfc/rfc5201.txt`.

[31] S. Pietrowicz, H. Shim, G.D. Crescenzo, and T. Zhang. VDTLS - Providing Secure Communications in Vehicle Networks. 2008. Available from: `http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04544651`.

[32] G. Romero et al. OPEN Meter Project - Report on regulatory requirements. Available from: `http://www.openmeter.com/files/`

deliverables/Open_Meter_D1.2_Regulation_v1.1_20090717.pdf, 2009.

[33] J. Schönwälder and V. Marinov. On the impact of security protocols on the performance of (SNMP). *Network and Service Management, IEEE Transactions on*, 8(1):52 –64, March 2011.

[34] D.X. Song, D. Wagner, and X. Tian. Timing analysis of keystrokes and timing attacks on ssh. In *Proceedings of the 10th conference on USENIX Security Symposium - Volume 10*, SSYM'01, pages 25–25, Berkeley, CA, USA, 2001. USENIX Association.

[35] ETSI Technical Committee Machine to-Machine communications (M2M). Machine–to–machine communications (m2m):smart metering use cases. Technical report, ETSI, May. 2010. Available from: http://www.etsi.org/deliver/etsi_tr/102600_102699/102691/01.01.01_60/tr_102691v010101p.pdf.

[36] Y. Yan, Y. Qian, and H. Sharif. A secure and reliable in-network collaborative communication scheme for advanced metering infrastructure in smart grid. In *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pages 909 –914, march 2011.

TRITA-ICT-EX-2012:102

www.kth.se