CHALMERS

Vulnerability Assessment of Secured Message and Identity Management Services in ETSI ITS C2C Communications

Master of Science Thesis (Computer Systems and Networks)

Nasser Nowdehi

Chalmers University of Technology

Department of Computer Science and Engineering

Göteborg, Sweden, August 2013

Vulnerability Assessment of Secured Message and Identity Management Services in ETSI ITS C2C Communications

Examiner: Tomas Olovsson

Chalmers University of Technology

Department of Computer Science and Engineering

SE-412 96 Göteborg

Sweden

Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering

Göteborg, Sweden, October 2013

Vulnerability Assessment of Secured Message and Identity Management Services in ETSI ITS C2C Communications

Abstract

The Cooperative Intelligent Transport Systems (C-ITS) is a set of applications that aim at improving road safety and traffic efficiency as well as providing environmental benefits by enabling vehicles and roadside infrastructures to communicate with each other. This type of communication is mainly based on exchanging messages containing information such as speed, location and direction sent over an ad hoc local area network.

However, the privacy of the users could be impaired by an adversary intercepting the information (e.g. location and identity of the driver) used in the messages exchanged between the vehicles and other ITS stations in an ad hoc vehicular network. Further, it is necessary to fulfill security requirements such as authentication and authorization to avoid unauthorized vehicles to get access to particular applications, services or privileges that should be only accessible by authorized vehicles (e.g. claim priority rights for emergency vehicles). As an effort to validate and authorize the ITS stations in a Vehicular Ad hoc Networks (VANET), the European Telecommunication Standards Institute (ETSI) has introduced a security architecture that brings the pseudonymity, confidentiality, authenticity and integrity into the VANET communications by using Certificate Authorities (CAs) and identity management procedures.

This master thesis aims at conducting a vulnerability assessment on the ETSI ITS Secured Message and Identity Management Services in ETSI ITS C2C Communications by integrating sign/verification services into an existing implementation of the ETSI ITS communication system. We also propose countermeasures to eliminate the identified vulnerabilities.

The vulnerability assessments performed in this thesis identify one major flaw in the design of the ETSI ITS security protocol concerning the location of the signature in a Secured Message. Furthermore, the assessments also identify 6 software vulnerabilities in the implementation of the ETSI ITS Secured Message which can be exploited for different types of attacks such as Denial of Service and buffer overflow.

Keywords

C-ITS, Privacy, Authentication, Authorization, VANET, ETSI, Security architecture, Pseudonymity, Integrity, Certificate authority, Vulnerability assessment, Secured message, C2C, Countermeasures, Signature, Attack.

Acknowledgment

I owe my deepest gratitude to my supervisor at Volvo Cars Company, Henrik Broberg for trusting and believing in me and giving me the chance to start this thesis! He helped me throughout the work and without his help it was not possible to conduct this thesis.

I would like to thank Volvo Cars Company for providing such a convenient, delightful and motivational environment!

I thank my supervisor at CTH, Tomas Olovsson, for his support, patience and insightful comments.

I thank my parents for always being supportive of my education. Although they are so far away there has never been a time when I have felt alone, whether in the time of conducting this thesis or in the general attainment of my education.

My earnest thanks to Dag Helstad from Actia Nordic Company for his hospitality and unsparing help during the implementation and testing phase of the thesis.

Finally, I would like to thank Armin Van Buuren for his amazing trance podcasts which were my companion during the time I was writing the report.

Table of Contents

Acronyms

AA	Authorization Authority
AL	Access Layer
BTP	Basic Transport Protocol
C-ITS	Cooperative Intelligent Transport system
C2C-CC	Car to Car Communication Consortium
CA	Certificate Authority
CAM	Cooperative Awareness Message
DENM	Decentralized Environmental Notification Message
DLL	Data Link Layer
DoS	Denial of Service
EA	Enrolment Authority
ECDSA	Elliptic Curve Digital Signature Algorithm
ETSI	European Telecommunication Standards Institute
GCC	GNU Compiler Collection
GPS	Global Positioning System
HMI	Human Machine Interface
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Standards Organization
ITS	Intelligent Transport System
ITSC	Intelligent Transport System Communications
ITS-S	Intelligent Transport System Station
KAF	Keep-Alive Forwarding
LDM	Local Dynamic Map
LLC	Link Layer Control
LTC	Long-Term Certificate
LTE	Long Term Evolution
MAC	Medium Access Control
NISTP	National Institute for Systems Test and Productivity
OSI	Open System Interconnection
PHY	Physical Layer
PKI	Public Key Infrastructure
PRESERVE	Preparing Secure Vehicle-to-X Communication Systems
RHW	Road Hazard Warning

SHA	Secure Hash Algorithm
TVRA	Threat Vulnerability and Risk Analysis
UDP	User Datagram protocol
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to Infrastructure/Vehicle
VANET	Vehicular Ad hoc Network
WAVE	Wireless Access in Vehicular Environments

List of Figures

List of Tables

Terminology

2G: Second-Generation of mobile telecommunications technology.

3G: Third Generation of mobile telecommunications technology.

Basic Transport Protocol (BTP): The BTP provides an end-to-end, connection-less transport layer service for VANET. It provides multiplexing-demultiplexing of messages such as CAM and DENM for different processes at the ITS Facilities layer to be transmitted using the GeoNetworking protocol. BTP is an unreliable transport protocol which means that the packet could be lost or duplicated [1].

Beacon: A single-hop GeoNetworking packet which advertises the position of the GeoAdhoc router to the other neighbors. It does not carry any payload.

Black-box testing: "A method of software testing that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings (see white-box testing)" [2]. The decision table testing, all-pairs testing and state transition tables are examples of black-box testing methods.

GeoNetworking protocol: A network layer protocol that provides non-IP packet routing for VANET. "It makes use of geographical positions for packet transport. GeoNetworking supports the communication among individual ITS stations as well as the distribution of packets in geographical areas. GeoNetworking can be executed over different ITS access technologies for short-range wireless technologies, such as ITS-G5 and infrared" [3].

IPv6 over GeoNetworking: Transporting IPv6 packets using GeoNetworking protocol without introducing modifications to existing IPv6 protocol implementations [4].

Keep-Alive Forwarding (KAF): The Facilities layer forwarding functionality. **"**The KAF functionality is optional for the DEN basic service. The main objective of KAF is to store a received DENM in the DEN basic service and to forward the DENM to other ITS-Ss when necessary." [5]. "For example, if the originator ITS-S is a break-down vehicle, it may stop transmitting the DENM unexpectedly due to the failed operation of the vehicle ITS-S. In this case, the KAF function of an ITS-S may be used to continue the transmission of the DENM that it has received before." [5].

Local Dynamic Map (LDM): "A cooperative system for road safety critical applications benefits from using digital maps. Such maps used in ITS may include lane-specific information including curbs, pedestrian walking, bicycle paths and road furniture such as traffic signs and traffic lights. Furthermore, all dynamic objects that are directly sensed or indicated by other road users by means of cooperative awareness messages may be referenced in such a digital map, referred to as local dynamic map (LDM)." [6]

Packet Centric Forwarding: The ITS networking & transport layer forwarding functionality that forwards DENM from the originator ITS-S to the destination area [5].

Proof of concept: In security terms, a proof of concept "refers to a demonstration that in principle shows how a system may be protected or compromised, without the necessity of building a complete working vehicle for that purpose." [7]

Vulnerability assessment: "The process of identifying, quantifying, and prioritizing (or ranking) the vulnerabilities in a system" [8].

White-box testing: A method of testing the source code of applications to create an error free environment. It uses design techniques that exercise every visible path of the source code to minimize errors and create an error-free environment. "The whole point of white-box testing is the ability to know which line of the code is being executed and being able to identify what the correct output should be." [9]. The control flow testing, data flow testing, branch testing, decision coverage, path testing and statement coverage are some examples of white-box testing methods [9].

1 Introduction

1.1 Background

Cooperative Intelligent Transport System (C-ITS) enables vehicles and roadside infrastructures to communicate by exchanging messages containing sensitive data such as speed, direction, geographical position, driver identity, car identity and occurrence of an event or hazard. However, the exchange of sensitive data in ITS messages could lead to violation in privacy of the ITS stations. For example the position of the vehicle could be tracked or the identity of the vehicle or the driver could be revealed to anyone who listens to the exchanged ITS messages.

Furthermore, carrying sensitive data could also lead to misuse of the features that are only available to validated and authorized users. For instance an adversary's vehicle could broadcast "Emergency vehicle approaching" messages to other neighboring vehicles to get ahead in a traffic jam. This type of misuse and similar cases in which an adversary is able to abuse the ITS features have been analyzed by the European Telecommunications Standards Institute (ETSI) and described in Threat, Vulnerability and Risk Analysis document (TVRA) [10].

As an on-going process, the ETSI in collaboration with other standardization organizations such as ISO[1] and IEEE[2] is currently developing the European standard for security services in Vehicular Ad-hoc Networks (VANET). The security architecture introduced by ETSI ITS, is in compliance with the IEEE 1609.2 security standard with some amendments and aims at validation and authorization of the ITS stations via Certificate Authorities (CAs) issuing enrolment credentials and authorization tickets.

1.2 Motivation and purpose

As for any other communication security architecture in the test development cycle, partially implementing and testing of the ETSI ITS security protocol enables the researcher to gain empirical knowledge about it which is based on the experiences and observations. By using that empirical knowledge it would be feasible to identify the complexities and weaknesses of the protocol design which have the potentiality to be interpreted wrongly and lead to vulnerabilities caused by erroneous implementation.

The secured message, certificate formats and the identity management services are the basic building blocks of the ETSI ITS security architecture. These basic blocks along with certificate authorities make it feasible to validate and authorize ITS stations whilst satisfying the pseudonymity of the ITS stations as a privacy requirement.

Since the specifications of the ETSI ITS security architecture are published and the vulnerabilities described in ETSI TVRA document have been identified based on theoretical and not empirical methods, performing an in-depth vulnerability assessment on an implementation of ETSI ITS security services has been the motivation of conducting this Master thesis. Therefore, the purpose of this thesis is to implement the ETSI ITS secured message, certificate format and the identity

[1]International Organization for Standardization
[2]Institute of Electrical and Electronics Engineers

management services and perform vulnerability assessments on those implementations.

1.3 Aim

The main aim is to assess vulnerabilities of ETSI ITS identity management services by answering the following questions:

- Are there complexities/ambiguities in the ETSI ITS security protocol that might be misinterpreted and lead to software vulnerabilities?
- Is there any flaw in the design of the ETSI ITSC security protocol that needs to be addressed? What is the proposed solution to fix the flaw?
- What countermeasures should be taken into consideration during implementation phase to counter any identified vulnerability?

In order to answer the above question the following steps should be taken:

- Literature study of ETSI ITS security headers and certificate formats specifications, related articles and project deliverables.
- Implementing a proof of concept of the main ETSI ITS security structures and identity management services in order to gain an in-depth knowledge about the ETSI ITS communication (ITSC) security architecture.
- Integrate sign/verify services as two major functionalities of the identity management services into an existing implementation of the ETSI ITSC system without security services.
- Test the functionally of the implementation done in previous step.
- Perform vulnerability assessment (qualitative) on the implementation done in previous step together with the knowledge gained from studying and implementing the ETSI ITS security headers and certificate formats specifications.

1.4 Limitation and scope

The evaluations performed in this Master thesis only cover the VANET security and do not cover the in-vehicle security. Nevertheless, it has been assumed that the input data from the in-vehicle sensors could be invalid due to hardware faults or an adversary's attempt trying to tamper the in-vehicle hardware. Furthermore, this thesis mainly concentrates on the vulnerability assessment of the security headers, certificate formats and identity management services used in V2X communications and leaves the vulnerability assessments of the communications between the ITS stations and the certificate authorities for future work.

There have been two major limitations in the process of conducting this Master thesis:

- **Time**: From complexity and multiplicity point of view it is necessary for vulnerability assessments to be comprehensive but unfortunately the time restrictions affected the number/complexity of the tests performed in this Master thesis and lead to less/simpler tests.

- **Resource**: There were a few similar and publicly available projects within ITSC security at the time of conducting this Master thesis. However, the lack of informative documents made the research phase more time consuming than what was planned at the beginning and left less time for assessment phase and lead to less/simpler tests at the end.

1.5 Contribution

This Master thesis was funded by Volvo Cars Corporation and has been conducted in cooperation with Chalmers University of Technology. All the workload of this thesis has been done in the R&D department of the Volvo Car Corporation, Göteborg, Sweden.

1.6 Report structure

This thesis report is structured as follows:

- Chapter 2 discusses the research approaches used in the conduction of this thesis. It describes both the theoretical and empirical methods used in this research (literature review).
- Chapter 3 illustrates the VANET, its communication architecture and important message types as well as its basic set of applications. Furthermore, it describes some important identity management services used in VANET communications.
- Chapter 4 is dedicated to the implementation specifications. It first describes the implementation of the essential security structures and services (e.g. security header and certificates) and then it illustrates the integration process and how it changes the communication architecture.
- Chapter 5 presents the results of the tests and assessments performed on the implementation which has been described in chapter 4. It identifies certain points of the security structures or procedures that have the potentiality to become software vulnerability due to erroneous implementation and also the identified flaws in the design of the ETSI ITS security protocol. Finally it proposes countermeasures to eliminate the identified vulnerabilities.
- Chapter 6 contains conclusions and gives answers to the research questions of this thesis based on the assessment results and observations made throughout the work. It also outlines suggestions for further research.

2 Methodology

This chapter describes different research methods that have been applied within the process of this thesis work including data collection, implementation and evaluation. According to the thesis description, this research task is divided into 3 main phases. The goal of the first research phase is to understand the general concepts of cooperative intelligent transport systems as well as the safety and security concerns within this area. This is achieved by studying relevant research papers and reports published by other projects within ITS security area and investigate their achievements and constraints.

Combining the knowledge gained from the previous phase with the latest security standard drafts written by ETSI, the second phase aims at integrating two Identity management services (sign/verify) into the ETTE ITSC platform used in Volvo Cars Corporation for test and evaluation purposes. Finally, the third phase of the research aims at testing the functionality of the identity management services implemented in the previous phase and assessing the software vulnerabilities caused by erroneous implementation (as a result of misinterpreting the standard), identifying flaws in the design of the protocol and proposing solutions and countermeasures to fix them.

2.1 Literature review

By reviewing the latest information in a particular area, a literature review aims at studying the accomplishments, constrains and scientific methods found in the other related works. It allows the researcher to formulate the problems and direct the research path and also helps the reader to understand where the current research is situated within a specific area [11].

In this thesis, the materials used for literature review are among publicly available standard drafts from organizations such as ETSI and IEEE and publicly available publications of industrial projects such as PRESERVE and DRIVEC2X. The literature review of this thesis includes an in depth study of:

- The ITS station communication architecture
    - Functionalities of different layers and interfaces between them
- Transport/network protocols such as BTP[3] and Geo-Networking
- The structure of Secured Messages specially the header and trailer
- The role of different certificate authorities and the structure of certificates
- The certificate validation process
- The specification of the sign/verify services as well as the signature structure
- The process of encoding /decoding secured messages

2.2 Empirical Research

Empirical research is acquiring knowledge using experiments and observations. The questions and hypotheses that have been derived in the early stages of the research could be answered or investigated by quantitative or qualitative (even the combination) analysis of data collected from these tests.

[3] Basic Transport Protocol

Since in the time of conducting this thesis the latest ETSI ITS standard document for security headers and certificate formats [12] was a draft published for evaluation purposes, the first question arose by this thesis was: To what extent the implementations based on the current draft of the ETSI ITS standard for security headers and certificate formats are open to software vulnerabilities caused by misinterpretation of the standard? The second question arose by this thesis was: Is there any flaw in the design of the security protocol that needs to be addressed? Therefore, the following steps have been taken to answer these questions using empirical research method:

1. Integration of sign/verify services as two major identity management services described in ETSI ITS [12], into an existing implementation of ETSI ITSC system called ETTE. The source code of the ETTE communication system is confidential and has been provided by the ACTIA Nordic AB Company and Volvo Car Corporation as a government funded project.
2. Using qualitative research method for analyzing the results of the vulnerability assessments performed on the implementation done in step 1.

3 VANET and security services

The aim of this chapter is to shortly introduce the Vehicular Ad-hoc Network (VANET), highlight some of VANET applications, characterize different layers of the VANET communication architecture (especially the role of security layer), shortly describe different message types exchanged between the ITS stations and finally elaborate the role of the security services and certificate authorities in VANET communications. It is worth mentioning that the Secured Message structure as an important element in securing VANET communications has been described in details in chapter 4 of this thesis report.

3.1 Vehicular ad-hoc network

Vehicular Ad-hoc Network is a rapidly growing class of wireless networks used in the transportation industry to provide both inter-vehicle and roadside-to-vehicle communications. VANET enables the vehicles and the concerned authorities to distribute traffic information and consequently increases the road traffic efficiency by keeping the drivers informed via routing advisements. Also, VANET allows different vehicles to exchange driving information such as speed, direction and location. Therefore it enables vehicles to show safety warning messages to drivers and improves the safety of passengers as a result [13]. Figure 1 illustrates an example of VANET and the messages exchanged between ITS stations.

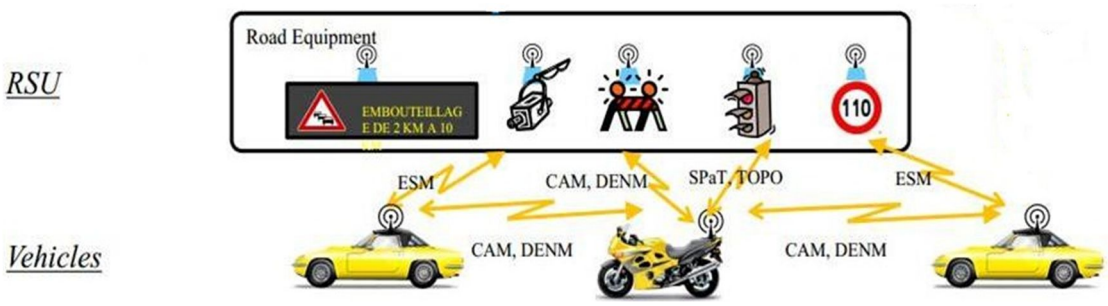

Figure 1.A simple example of VANET [14]

3.1.1 VANET communication architecture

The architecture of communication stack in VANET generally follows the same concepts as the OSI [4]model but it also adds some extensions to it [6]. As shown in Figure 2, while different layers in OSI model are isolated from each other and each layer only serves the above layer and is served by the layer below [15], some layers in VANET communications stack provide further functionality called "cross-layer functionality" and interact with every layer in the architecture [6].

Nevertheless, some layers in VANET communication stack (Access, Networking & Transport and Facilities) correspond to more than one layer in the OSI model. Further, the architecture is extended by adding two new layers: Management and Security layers.

[4]Open Systems Interconnection

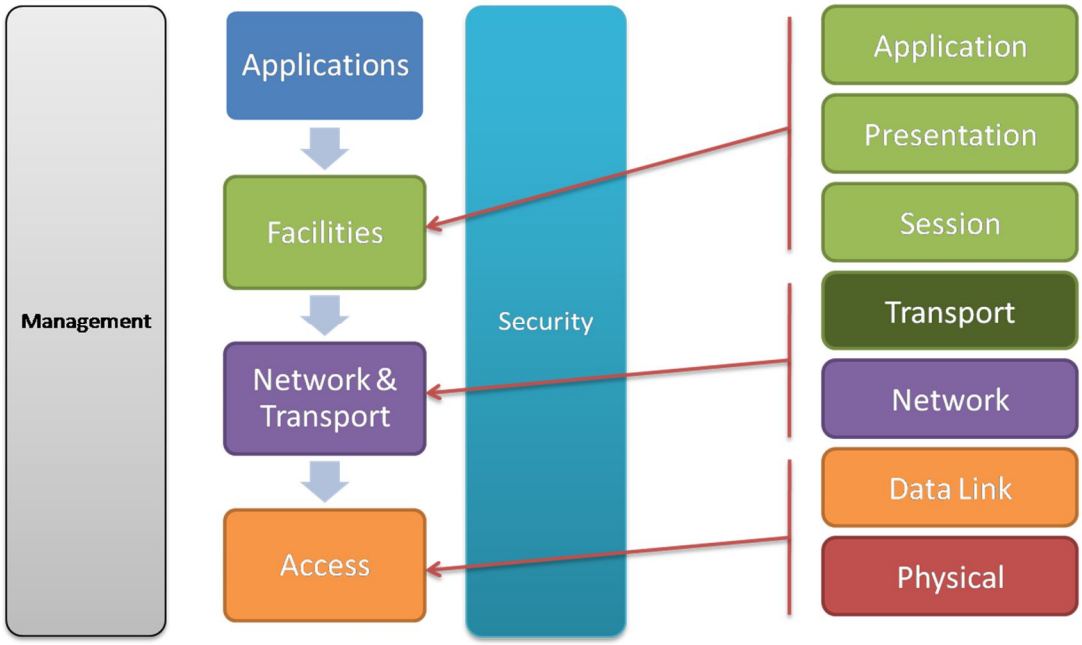

Figure 2. The mapping of ETSI ITS layered architecture to OSI model.

Each of these layers is shortly described below.

ACCESS

The Access Layer (AL) represents the Physical and Data Link layers in the OSI model. The lowest sub-layer in the AL is the physical layer (PHY) which is physically connected to the communication medium and transmits the signals. It supports different access technologies such as Wi-Fi, 2G, 3G, Bluetooth, etc. The second sub-layer of the AL is the Data Link Layer (DLL) which consists of two sub-layers called MAC [5]and LLC[6] as shown in Figure 3. Both AL and PHY are managed by an entity inside the access layer called Layer Management [6].

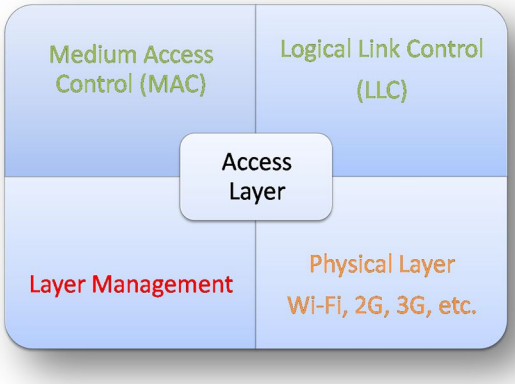

Figure 3.The Access layer: Data link and Physical sub-layers.

The Access layer has three interfaces for communication with other layers:

- **IN**: Provides data link layer services to the Networking & Transport layer [16]

[5] Medium Access Control
[6] Logical Link Control

- **MI**: The Interface between the Management entity and the AL providing management services [17]
- **SI**: The interface between the Security entity and the AL providing security services [18]

NETWORKING & TRANSPORT

As shown in Figure 4, the Networking & Transport layer in VANET corresponds to the third and fourth layers in the OSI model with new functionalities dedicated to the ITS. This layer comprises one or several networking and transport protocols and a layer management entity which manages the network and transport sub-layers [6]. It supports multiple networking and transport protocols such as:

- **Network protocols**
  - GeoNetworking [19]
  - IPv6 [20]
  - IPv6 over GeoNetworking [4]
  - CALM FAST [21]
- **Transport protocols:**
  - UDP/TCP.
  - ITSC dedicated transport protocols such as Basic Transport Protocol (BTP).

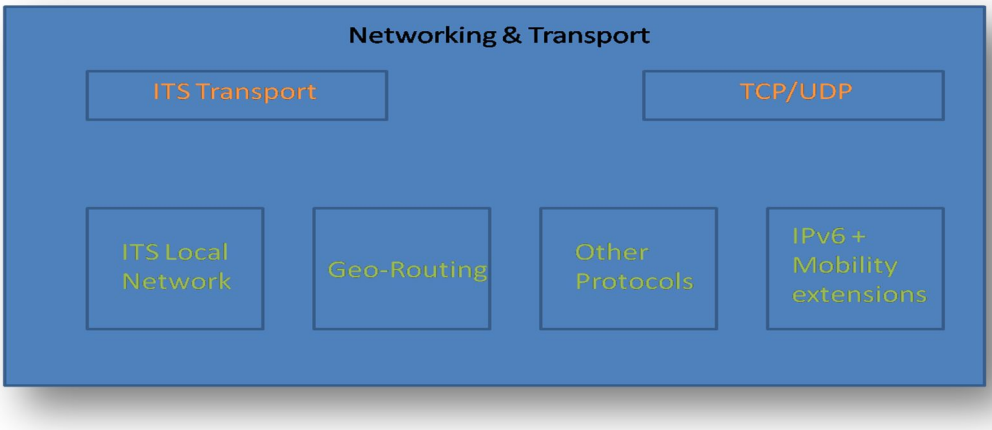

Figure 4. The Networking and Transport layer and its components.

The Networking & Transport layer has 4 interfaces for communication with other layers as described below:

- **NF**: Provides communication services to the Facilities layer [22]
- **NM**: The interface to the Management entity providing management services [23]
- **SN**: The interface to the Security entity providing security services [24]
- **IN**: The interface to the AL providing communication services [16]

Facilities

The Facilities layer provides the functionality of the fifth, sixth and seventh layers of the OSI model with respect to the ITS requirements. As shown in Figure 5, the Facilities layer functionality is divided into: Application support, information support, communication/session support and a management entity to handle aforementioned functionalities [6].

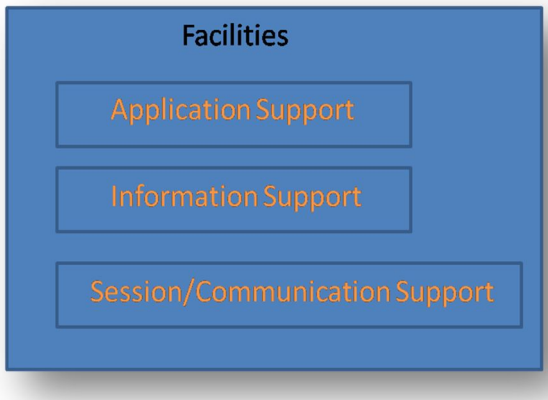

Figure 5.The Facilities layer and its main functionalities.

According to the ETSI standard for ITSC architecture [6] the Facilities layer supports different services for the ITS dedicated applications, such as:

- Generic HMI support
- Data presentation
- Addressing
- Location referencing and time stamping of data
- Local Dynamic Map (LDM) support [25]
- Relevance checking
- Station data provision
- Support for DENM [7] [5], CAM [8] [26], etc.
- Repetitive transmission of messages
- Channel selection

The Facilities layer has 4 interfaces as described below [6]:

- **FA**: Provides services to the Application layer
- **MF**: The interface for communications with the Management entity [27]
- **SF**: For access to the security services provided by the Security entity [28]
- **NF**: For access to the communication services provided by the Networking & Transport layer [22]

Management

As illustrated in Figure 6 and described by ETSI [6], the Management entity is responsible for different services such as:

[7]Decentralized Environmental Notification Messages
[8]Co-operative Awareness Messages

- Regulatory management: A common framework for regulatory information management [6]
- Cross-layer management
- Application management: Manages the installation and configuration of ITS-S applications [6].

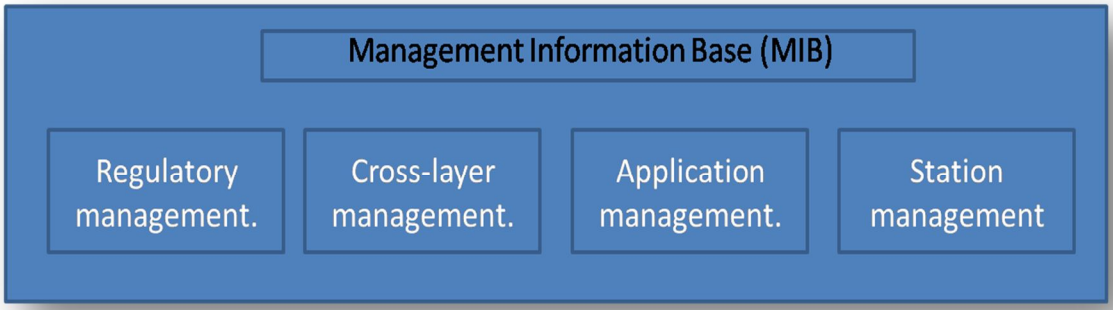

Figure 6. The Management entity and its major functionalities.

Each of the above mentioned services represents a new set of services grouped based on their functionality. As described in the ETSI standard for ITSC architecture [6], some of these services are listed below:

- General congestion control
- Management of service advertisement
- Network and communication management (e.g. configuration and update management, Communications system fault monitoring, etc.)
- Application mapping
- Local node map: Maintains information of neighboring nodes by combining communication parameters such as MAC address and networking address together with their kinematic information such as position, speed and heading [6]
- Cross-interface management

The Management entity has 5 interfaces as described below:

- **MI**: The interface to the Access layer [17]
- **MN**: The interface to the Networking & Transport layer [23]
- **MF**: The interface to the Facilities layer [27]
- **AM**: The interface to the Application layer
- **MS**: The interface to the Security entity [29]

Security

According to ETSI [6], the Security layer provides different security services including certificate management, authentication/authorization, cryptographic functionalities, etc. to other layers in the ETSI ITSC architecture. As shown in Figure **7,** abovementioned services can be accessed via the following interfaces:

- **SI**: Provides security services to the Access layer [18]
- **SN:** Provides security services to the Networking & Transport layer [24]

- **SF**: Provides security services to the Facilities layer [28]
- **SA**: Provides security services to the Application layer
- **MS:** Provides security services to the Management entity [29]

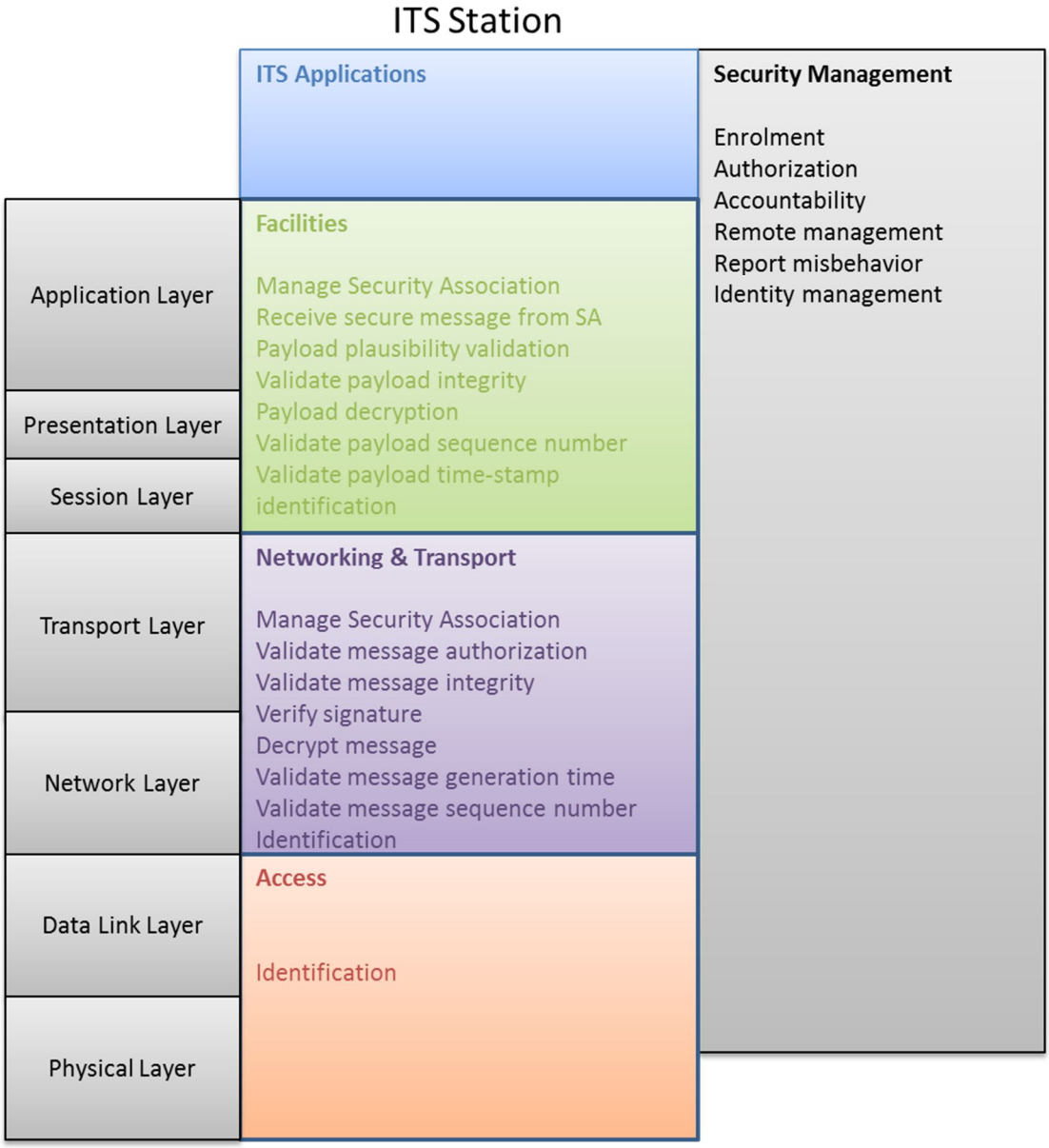

Figure 7. The ITS security layer services

3.1.2 WAVE [9] message types

According to the ETSI ITS TVRA [10], there are two fundamental message types exchanged in the V2V[10] and V2I [11] communications as described below:

Cooperative Awareness Messages (CAM): This type of message is generated by the Facilities layer containing detailed information regarding the vehicle's position,

[9] wireless access in vehicular environments
[10] Vehicle to Vehicle
[11] Vehicle to Infrastructure

speed, direction, etc. broadcasted in high frequency to the neighboring nodes. Hence, CAM helps the vehicles to be aware of the presence, status and movement of the other vehicles located in a single hop distance [26].

Decentralized Environmental Notification Message (DENM): Contrary to the CAM that is broadcasted in high frequency and in a single hop distance, the DENMs are generated and exchanged upon the occurrence and detection of some specific events (event-driven, e.g. broken car event) and could be optionally forwarded between the ITS stations as long as certain conditions described in the DENM specification [5] are met. This type of message plays an important role in the Cooperative Road Hazard Warning (RHW) applications in which a vehicle generates DENMs after detection of an event and alerts the other vehicles concerned by the event [5].

After receiving a relevant DENM, a warning will be shown to the driver. Further, based on the geographical area and the expiration time specified in DENM, each vehicle decides if the message should be retransmitted. The transmission of DENM stops either by sending a specific type of DENM or by using a timer which stops the transmission if the event is finished [5].

3.1.3 VANET basic set of applications

According to the ETSI ITS basic set of applications document [30], there is a compulsory set of services that each ITS station should support. Further, the CAM and DENM specification documents [26] [5] also provide some examples of the CAM and DENM applications. The following use cases are examples of the VANET basic set of applications:

1. Approaching emergency vehicle
2. Stationary vehicle warning (accident/broken vehicle)
3. Collision risk warning
4. Signal violation warning
5. Road-work warning
6. Wrong way driving warning
7. Emergency electronic brake light
8. Slow vehicle warning
9. Road adhesion (Slippery road)
10. Low visibility/Strong wind

3.2 Security services in VANET communications

The ITSC messages (e.g. CAM) exchanged between the ITS stations might contain sensitive data such as driver's name, vehicle license plate and location. Therefore “it is necessary to ensure that the data cannot be linked to any individual so that no personally identifying information is leaked by the CAM service” [31]. In other words, “It should not be possible for an unauthorized party to deduce the location or identity of an ITS station by analyzing communications traffic which flows to and from the ITS user's vehicle” [10]. On the other hand, the ITS users should be trusted before using specific ITS services and applications that are only available to authorized users. For example, authorization to use chargeable services like

personalized route guidance and authorization to claim priority rights for emergency and police vehicles are two examples of the CAM authorization.

According to both ETSI ITS Trust and Privacy Management [32] and ETSI ITSC Security Architecture and Security Management [31] documents:

- The privacy of the ITS stations should be protected by using pseudonym identifiers that can be frequently changed.
- The ITS stations should be trusted before using the ITS system (including services and applications) by provision of certificates proving their identity and permissions.

3.3 ETSI ITS certificate authority hierarchy

As shown in Figure 8, ETSI ITS has introduced a hierarchy of Certificate Authorities (CAs) to provide security services such as authentication and authorization that enable ITS stations to prove their identity and permissions and yet stay anonymous. Each CA entity issues digital certificates for ITS entities and is also able to revoke the issued digital certificates. The following ITS authorities are listed and described according to the ETSI ITS Security Trust and Privacy Management document [32]:

Enrolment Authority (EA)

The EA issues a Long Term Digital Certificate (LTC) used for authentication of the ITS stations. To be a part of the ITS and also gain authorization to use further services, it is necessary to have an LTC. To receive an LTC, first the ITS station uses its initialization credentials to request the enrolment credentials from the EA. The process of gaining initialization credentials is performed in conjunction with the manufacturer of the vehicle or the ITS device [31]. According to the ETSI ITS [31] the initialization credentials are

- A canonical identity for the ITS station
- A public private key pair for the ITS station
- A generic profile of the properties of the ITS station
- A cryptographic certificate linking the canonical identity with its public key and generic profile

Next, the EA authenticates the sender of the request and issues a long term certificate for it. The issued LTC is then used by the Authorization Authority to authorize the ITS station [32].

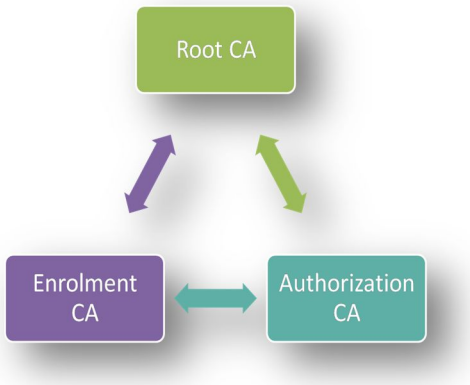

Figure 8. The ETSI ITS certificate authority hierarchy

Authorization Authority (AA)

The AA is responsible for granting specific permissions to the ITS stations. In order to authorize an ITS station, first it will be asked to provide its long term certificate to prove that it has been authenticated by an EA. Next, the AA checks the validity of the received LTC and finally issues authorization tickets (Pseudonym certificates). The pseudonym certificates shall be changed periodically according to the predefined time periods defined by the standard [32].

Root CA

The Root CA is the root of trust for all CAs and each ITS station should have access to at least one root certificate. A root certificate address might have been either installed by the manufacturer or broadcasted over the air [32].

4 Implementation specification

The main goal of this chapter is to describe the ETTE ITSC architecture and also elaborate on the method of integrating sign/verify services into the communication architecture. Furthermore, it describes the process of sending and receiving ITS messages, the specification of the sign/verify processes and the structure of the certificates and Secured Message formats.

4.1 ETTE project

Following the ETSI ITS standards, the ETTE project aims at creating a cost-effective C2X platform for wirelessly connected vehicles based on the IEEE 802.11p standard and 4G/ LTE[12] by modifying the existing telematic platforms used in AB Volvo and Volvo Cars Corporation. Another goal of the project is to develop a test workbench for verification of the C2X communications [33] [34]. The communication stack for the ETTE platform has been implemented by ACTIA Nordic AB [35] (except for the application layer).

4.1.1 Overview of ETTE ITS communication architecture

As illustrated in Figure 9, the ETTE ITSC architecture follows the same layered architecture as the one introduced by ETSI ITS [6] with some amendments purposefully made to fulfill the requirements of the project. The implementation of the DENM has been done according to the specifications defined in a draft of the standard published by ETSI ITS [36]. Furthermore, the Applications and Facilities layer common data as well as the structure of the Facilities layer has been designed and implemented based on the latest draft of the standard published by ETSI ITS [37] at the time of conducting this thesis.

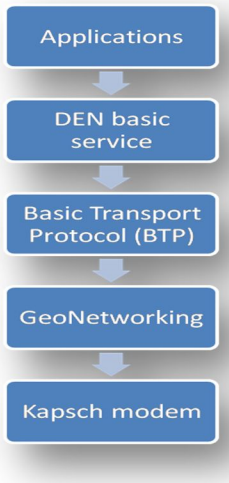

Figure 9. The ETTE ITSC architecture (stack)

In the ETTE communication architecture the functionality of the Access layer has been deployed in a separate modem device called Kapsch [38]. In other words; the Access layer has been separated from the other layers of the communication architecture.

[12]Long Term Evolution

The following DENM transmission scenario explains the functionality of different layers in the ETTE ITSC architecture run in the simulation mode[13]. In this scenario, the DENM contains a road hazard warning message for a broken car event.

Originator side

Application layer

The detection of a broken car event immediately triggers the associated RHW application to start the transmission of a broken car DENM. First, the application sends a transmission request to the DEN basic service in Facilities layer that is responsible for construction, management and processing of the DENMs [36].

Facilities layer

After receiving the request from the Application layer, the DEN Basic service creates a DENM containing the location of the broken car and some additional information about the broken car event. Next, it passes the DENM on to the Basic Transport Protocol (BTP) layer [1] implemented in the Transport layer that provides BTP services to the Facilities layer.

Network & Transport layer

The BTP layer adds the BTP header and parts of the GeoNetworking (GN) [19] header to the outgoing DENM (packet) and passes it to the GN protocol interface implemented in Network Layer. It is worth mentioning that ETTE only supports GeoBroadcast and BEACON packet headers described by ETSI ITS [3] and also implements the packet routing procedures described in the same document. Once the next hop for the GN packet (containing DENM) has been decided, the GN protocol passes the packet on to the Access layer.

Access layer

In simulation mode, the Access layer encapsulates the packet inside a UDP packet and passes it on to the Kapsch modem simulator to take the final step and transmit the signals. It is only in simulation mode that the outgoing packet is encapsulated in UDP frame and sent to the loopback interface instead of being transmitted using the IEEE 802.11.p.

Receiver side

Access layer

The Kapsch modem receives the packet and removes the UDP header off the packet and passes it on to the GN protocol interface in Network Layer.

Network & Transport layer

GN protocol updates the location table and checks for the packet duplication. Now, If the ITS station is inside of the geographical area specified in the received GeoBroadcast packet, the Network layer removes the GN header and passes the packet payload (BTP packet) on to the BTP layer to be handled. Now, the Packet Centric Forwarding [3] functionality decides if the packet should be forwarded to the destination area. Finally, the BTP passes the payload of the BTP packet on to the DEN Basic Service in the Facilities layer.

[13] The simulation environment has been described in chapter 5 of this document.

Facilities layer

The DEN Basic Service extracts DEN data, and passes the event information on to the Local Dynamic Map (LDM) [25]. Further, if certain conditions described in [5] are met, the optional Keep-Alive Forwarding (KAF) functionality decides if it has to store the received DENM in DEN basic service to be forwarded [5].

Application layer

The corresponding application gets notified of the new, updated or invalidated events by the LDM.

4.2 Secured Message format

The Secured Message format specifies how different fields should be encoded to form a secured message. For example, Figure 10 illustrates the secured message format used for a signed DENM.

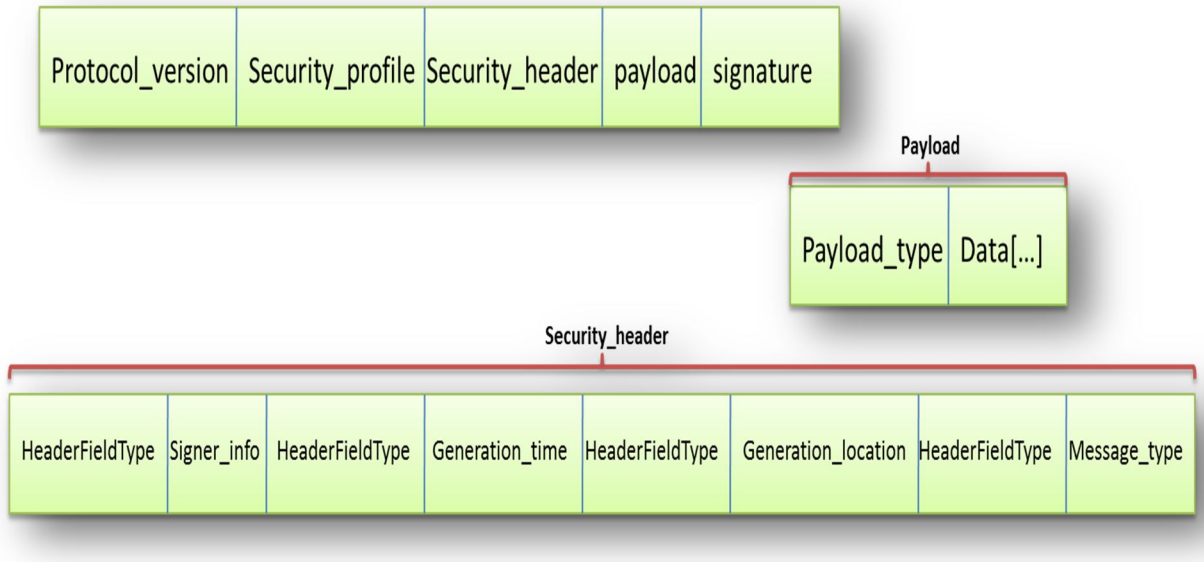

Figure 10.The Secured Message packet used for carrying a signed DENM

Further, the following section describes different parts of the Secured Message structure according to the ETSI TS document for security headers and certificate formats [12] and with more concentration on DENM:

protocol_version

8 bits of data that specify the protocol version applied to the packet. At the time of conducting this Master thesis the value for this field was 0x01.

security_profile

8 bits of data specifying the Security Profile [12] used for encoding the packet. It defines the content of the security header, payload and trailer fields. Security Profile number for DENM is 0x02.

header_fields<var>

A variable-length vector of different security header fields which should be encoded according to the order specified by the security profile for each type of message. According to the ETSI ITS [12], for all types of messages the security header fields should be encoded in ascending numerical order of the header field type unless the order defined in the security profile is different. Each *HeaderFieldType* is 8 bits of data. The following list is an example of different header field types:

- *generation_time* (For DENM it shall contain the current absolute time)
- *generation_location* (For DENM it shall contain the current location of the sender at the time of message generation)
- *expiration*
- *message_type* (For DENM it shall contain the value 0x01)
- *signer_info* (For DENM it shall contain the certificate of the signer of the message.)
- *recipient_info* (the RecipientInfo structure stored after this type of header filed contains recipient-specific information which are the identifier digest for the recipient's certificate and the public key algorithm used for generating that digest.

According to ETSI ITS [12], for a DEN message the *signer_info* field shall be placed first as an exception, unless overridden by the security profile.

payload_fields<var>

As shown in Figure 11, the payload_fields is a variable-length vector of different allowed payload types. Each *payload_type* is 8 bits of data and it could be any of the following types [12]:

- *unsecured*
- *signed*
- *encrypted*
- *signed_and_encrypted*
- *signed_external*

Figure 11.General view of Payload field.

trailer_fields<var>

A variable-length vector of different security trailer fields which follows the same encoding order rules as the header_fields. At the time of conducting this Master thesis, *signature* was the only trailer field type defined in ETSI TS security headers and certificate formats document.

4.3 ETSI TS 103 097 format certificate

The ETSI TS Certificate format duplicates some of the main elements of IEEE 1609.2 certificate format but with new amendments and extra fields to fulfill its own purposes. As shown in Figure 12, the ETSI TS 103 097 certificate structure specifies how different information should be encoded inside a certificate.

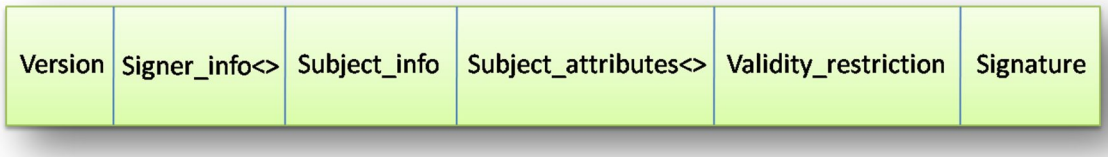

Figure 12. The certificate format for carrying different types of certificate

For more elaboration, the following section describes different fields of the ETSI TS certificate structure according to the ETSI TS document for security headers and certificate formats [12]:

Version

8 bits of data representing the version of the certificate. At the time of conducting this Master thesis the value for this field was 0x01.

signer_info<var>

A vector containing different fields filled with the information about the certificate's signer. According to the ETSI ITS [12] it could be any of the following types:

- *self* (self-signed certificate.)
- *certificate_digest_with_ecdsap256*
- *certificate*
- *certificate_chain*

subject_info

It contains different subject types and specifies the certificate's subject name. Each *SubjectType* is 8 bits of data. Some of the subject types are listed below:

- *authorization_ticket (Pseudonym certificate)*
- *aturhorization_authority*
- *enrollmnet_authority (long-term certificate)*
- *root_ca*

subject_attribute<var>

It is a variable-length vector containing additional information about the certificate subject. In fact, different subject attribute types specify what technical information shall be included in the certificate. This information shall be encoded in ascending numerical order of their type, unless specified in a different order by the security profile [12]. Each *SubjectAttributeType* is 8 bits of data. Some of the subject attribute types are listed below:

- *verification_key (A public key shall be given)*
- *encryption_key (A public key shall be given)*

- *reconstruction_value (An ECC point shall be given)*
- *assurance_level.*

validity_restriction<var>

8 bits of data that specify the validity of the certificate based on the time duration or the geographical region the certificate is issued for. The validity restriction is specified by any of the following attributes:

- *time_end*
- *time_start_and_end*
- *time_start_and_duration*
- *region*

Signature

It contains the signature of the certificate signed by the CA in charge.

4.4 Sign and Verify processes

A digital signature provides both authenticity and integrity for digital communications and helps the recipient of a message to find out if the sender of the message is a known identity (authenticity checking) and also make sure that the message has not been altered during the transmission (integrity checking). Digital signature takes advantage of the public key cryptography by signing the message using the sender's private key and verifying the signature using the sender's public key [39].

According to ETSI ITS [12], Elliptic Curve Digital Signature Algorithm (ECDSA) [40] shall be used for signing and verification of the messages transmitted in the VANET communications. Further, according to ETSI ITS [12] the symmetric algorithm used for this purpose is the *ecdsa_nistp256_with_sha256* which was also used by the pilot PKI during the time of conducting this Master thesis. The essential data structures for signing and verification of the messages (e.g. Signature structure, EcdsaSignature structure, etc.) were also implemented according to the specifications described in the same document [12].

Sign process

According to the ETSI ITS [12], the signature of a secured message shall be calculated based on the following steps:

1) Calculation of a hash digest over all the fields precedent to the *signature* structure in the trailer field (i.e. *protocol_version*, *header_fields*, *payload_fields* and any other trailer field located before the *signature* field) including the encoding of their length
   a) Payloads of type *Unsecure* or *Encrypted* shall be excluded from hash calculation except for their *payload_type* field.
   b) Payloads of type *signed_external* shall be included in the hash calculation only after the payload field
2) Signing the calculated digest (step 1) using the private key that is bounded to a public key stored in a pseudonym certificate

3) Store the pseudonym certificate and the signature inside the *signer_info* and *signature* structures respectively.

Verify process

The signature verification process starts with the certificate validation. According to the ETSI ITS [12], a certificate is valid only if

- The current time and region are within the time and region validity specified in the certificate
- The certificate has been issued for the communication type in use
- The certificate signature is valid
- The certificate for the signer of the given certificate is valid

After certificate validation, the receiver takes the following steps:

1) Calculates the hash digest of the message fields as described in the sign process
2) Extracts the public key from the pseudonym certificate stored in the message
3) Applies the gained public key on the signature of the message
4) Compares the calculated hash digest of step 1 with the received hash digest of step 3. The signature is valid only if the result of comparing the aforementioned hash digests is "equal"

4.5 Security integration

This section elaborates the method of integrating sign/verify services into the ETTE ITSC architecture from a high-level perspective. It describes the programming language used for implementation, the cryptographic library used for accessing standard cryptographic functions, the implemented functions and data structures, integration procedure and the schema of the ETTE communication architecture after the integration.

4.5.1 Programming language and cryptographic library

At the time of conducting this Master thesis, C++ was the programming language in which ETTE architecture was implemented. Accordingly, all the essential functions and structures for integrating sign/verify services are also written in C++ and according to the Google C++ coding standard [41]. The main essential compiling requirements are the GCC [14] and Gnu make. Finally, the Openssl [42] library has been used for cryptographic purposes such as storing big prime numbers and public/private keys, arithmetic calculations used in hash, sign and verify functions, etc.

4.5.2 Implemented functions and structures

The empirical phase of this master thesis starts with implementation of the Certificate structure defined by the ETSI ITS [12]. It is worth mentioning that the certificates provided by the pilot PKI follow the same format. Further, a certificate parser has been also implemented to extract the public key and other attributes of the received

[14]Gnu Compiler Collection

certificate. Finally, the Secured Message structure together with the sign and verify functions has been implemented according to the specifications defined by ETSI ITS [12].

4.5.3 Integration procedure

According to ETSI ITS [31] the enveloping of messages should be done in the Network layer. Also, as shown in Figure 13, the messages that should be signed /verified are sent to and received from the Access layer. As a result, the first step of the integration procedure starts with studying the interfaces between these layers.

As illustrated in Figure 13, in ETTE ITSC architecture the ILinkLayerObserver interface watches the Access layer for incoming packets and the ILinkLayer interface watches the Network layer for outgoing messages. Therefore, the next step was to modify the aforementioned interfaces and add the new code for controlling the direction of sent/received messages and redirect them to the security layer accordingly. Finally, the sign and verify functions with all of their pre-processing requirements are developed in the security layer to be called by the modified ILinkLayerObserver and ILinkLayer interfaces. It should be mentioned that due to lack of having remote communication to the pilot PKI in the time of conducting this master thesis, the certificates had to be locally stored and updated.

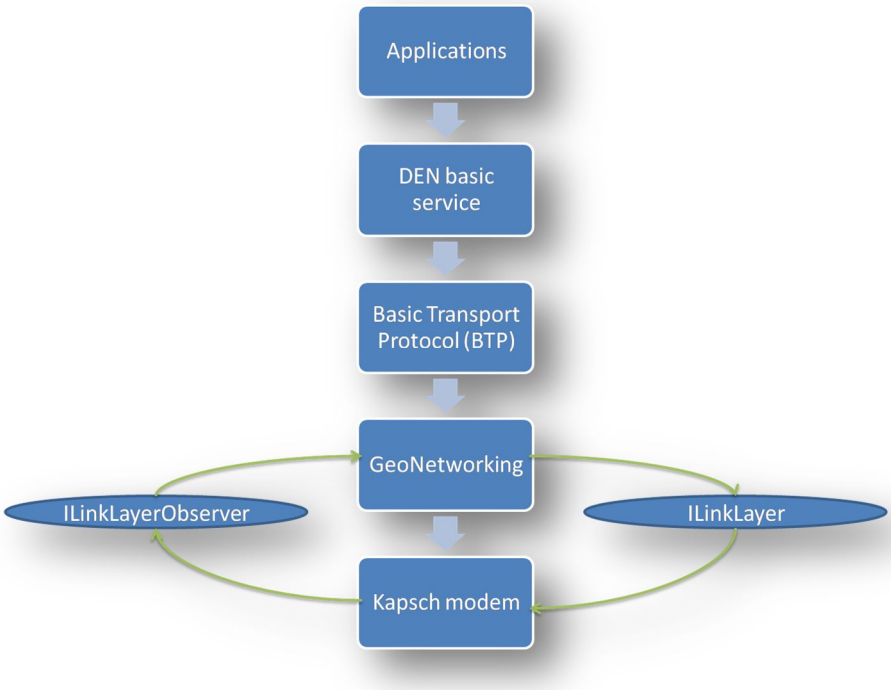

Figure 13.The ETTE ITSC stack and the interfaces between the Network layer and Link Layer

4.5.4 ETTE architecture after security integration

Figure 14 shows the new structure of an incoming/outgoing DEN message after integration of sign/verify services into the ETTE ITSC architecture. It should be mentioned again that the outermost frame (UDP) in Figure 14 is only used in

simulation mode to carry the Secured Message but in real mode the sent/received packet starts with the Secured Message frame.

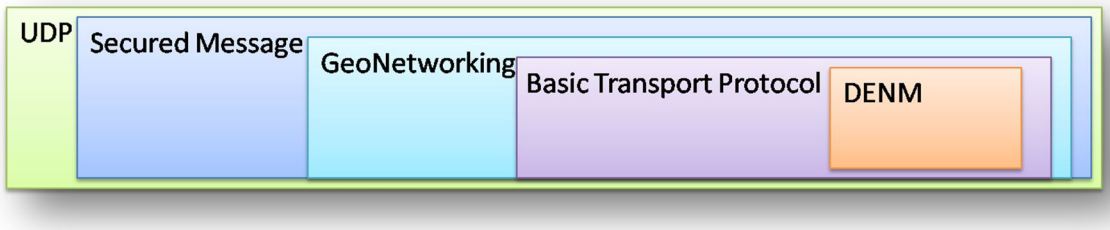

Figure 14.General view of a Secured Message used in simulation mode of the ETTE ITS

Finally, Figure 15 illustrates the new ETTE communication architecture after integration of sign/verify services.

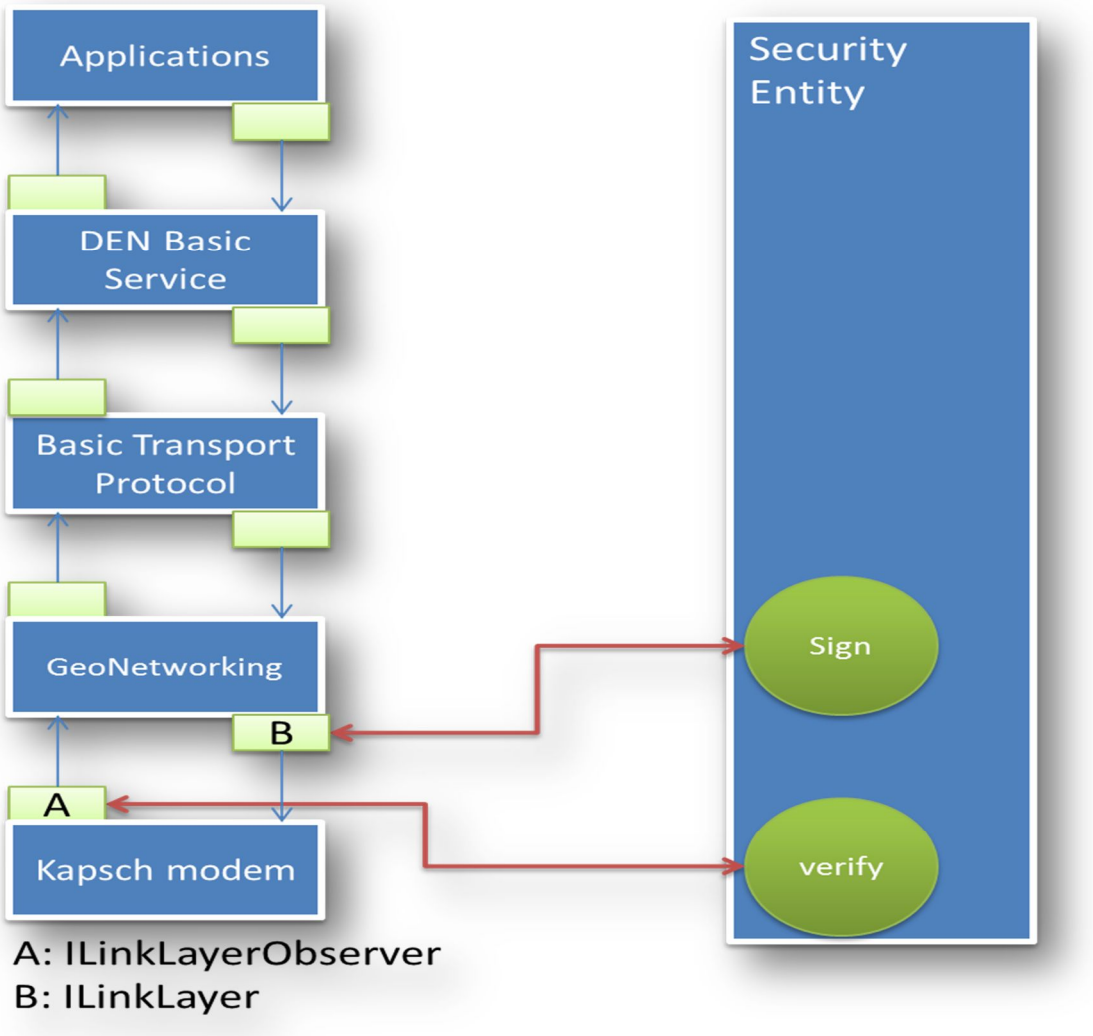

Figure 15.The ETTE communication architecture after adding sign/verify services

5 Functional testing and software vulnerability assessment

This chapter aims at presenting the test setups used for testing the functionality of the implementation described in chapter four, discussing the results, describing the test setups for software vulnerability assessment of the implementation and the possible countermeasures to eliminate the identified software vulnerabilities. Furthermore, it describes an identified flaw in the protocol design and possible solutions to fix the flaw. Since the implementation has been tested in simulation mode, this chapter also describes the simulation environment.

5.1 Simulation environment

The simulation environment and the test setups have been provided by Actia Nordic AB. The simulation environment allows the user to run multiple instances of the ETTE ITS stations communicating with each other and retrieves the runtime information and logs useful information such as the current location of each ITS station, moving status, distance from the other ITS stations, communication stack and GPS device status, etc. The simulator has a command parser written in Python [43] that allows the user to run different test setups by invoking commands with a well-defined syntax.

Running a test setup starts by specifying global options and then an arbitrary number of ITS stations followed by the options for each of them. For example it is possible to specify if the ITS station is a stationary vehicle or a vehicle moving in circle around a specific place. It also allows the user to assign a socket for sending some useful commands to the communication stack of each ITS station (e.g. turning the warning lights on and off) using a simulated command line environment. Furthermore, each ITS station has a GPS device simulator and all of the stations communicate through an air medium simulator. For demonstration purposes, the test package is also equipped with an ITS station visualizer which shows the ITS stations on a world map using their position registered in a location table.

5.2 Functional testing

The new implementation of the ETTE ITSC architecture described in chapter 4 has been tested by using Black-box Testing methods. Each test setup consists of a valid/invalid private key, public key and signature used as inputs to the sign/verify services. Further, seven valid/invalid ETSI ITS 103 097 V 1.1.1 pseudonym certificates and their corresponding private keys have been used for performing the tests. These certificates are provided by the Pilot PKI and have been locally stored on the test machine.

In each test setup, the originator signs the outgoing ITS messages by a private key bounded to a pseudonym certificate and transmits the certificate along with the message to the receiver. The receiver, on the other side; verifies the incoming messages using the public key stored in the received certificate. For example, the third row of Table 1 represents a test setup in which the signature of an outgoing ITS message has been crafted during the transmission. As shown in Table 1 the new ITSC stack has passed all the tests successfully.

#	Private key	Public key	Signature	Expected Verification result	Verification result
1	Invalid	Valid	Valid	Failed	Failed
2	Valid	Invalid	Valid	Failed	Failed
3	Valid	Valid	Invalid	Failed	Failed
4	Valid	Valid	Valid	Verified	Verified

Table 1. Basic sign/verification test results

5.3 Identified software vulnerabilities

The software vulnerabilities presented in Table 2 and Table 3 have been identified by performing White-box Testing on the implementation described in chapter four. These vulnerabilities have already been eliminated in the last iteration of implementation and testing of the system.

The presumption behind all of the identified software vulnerabilities in the clauses 5.3.1 and 5.3.2 is the intersection of the following elements:

- The existence of a flaw in the software/system
- The capability of the adversary to access the flaw
    - By gaining access to the original ITSC software by any means and investigating it for flaws
- The capability of the adversary to exploit the flaw
    - By sending crafted ITS messages which exploit the identified flaws.

5.3.1 Secured Message structure

Similar to the IPv6 header size, the Secured Message header and trailer have dynamic sizes and are defined as variable sized vectors. The size of the Secured Message header and trailer are specified at the beginning of each one. Furthermore, each secured message can carry multiple payloads of different types (such as signed, encrypted, unsecured, etc.) so the payload size is explicitly specified at the beginning of each payload. Each row in Table 2 illustrates the results of testing the ETTE ITSC stack with messages containing invalid values stored in vector length fields.

Furthermore, the information presented in the first row of Table 3 shows another test on the ETTE ITSC stack in which the system is tested by continuously receiving ITS packets with unknown values stored in the *protocol_version* and *security_profile* fields. Since the packet parser is unable to parse such packets and keeps buffering them without freeing the allocated buffer in the memory, it finally leads to a Denial of Service (DOS) attack by consuming all the memory on the target machine.

5.3.2 Certificate structure

The certificate structure has vectors of variable size too, so the ETTE ITSC stack has also been tested against using invalid vector length in the certificates. According to the results shown in Table 2, manipulation of the vector length values in a Certificate structure has no critical effects on the overall functionality of the ITSC stack. As described in the second row of Table 3 the behavior of the ETTE ITSC stack has also been tested against receiving packets with unknown *version* value.

#	Field names	Structure name	Stored value	Expected result	Actual result	Is a vulnerability
1	Header_fields<var> Payload_fields<var> Data<var> Trailer_fields<var>	Secured Message header	Less than real vector size/ More than packet size	Packet dropped	System crashed	Yes
2	Signer_info<var> Subject_attributes<var> Subject<name> Validity_restrictions<var>	Certificate	Less than real vector size/ More than packet size	Signature verification failed	Signature verification failed	No

Table 2. Test results for headers and certificates with invalid vector length

#	Field name(s)	Structure name	Stored value	Expected result	Actual result	Is a vulnerability
1	Protocol_version Security_profile	Secured Message	Unknown	Packet dropped	DOS attack. Consumes all the memory on the target machine	Yes
2	Version	Certificate	Unknown	Signature verification failed	Signature verification failed	No

Table 3. Test results for continuously receiving unknown values in protocol_version, security_profile and version.

5.4 Design flaw

The identified flaw in the design of the ETSI ITS security protocol is concerning the position of storing the signature in a secured message. According to ETSI ITS [12] and Figure 16 the ECDSA signature is stored inside the trailer of a secured message and it does not have a well-known position. As ETSI ITS [12] describes, other trailer fields (if there are any) are allowed to be stored before or after the signature field. Consequently, to find the beginning of the signature field, the parser should parse the trailer fields until it finds the trailer field with a *TrailerFieldType* of *signature*. Furthermore, according to the ETSI ITS [12], the payload and security header fields are always located before the trailer of a secured message and both are variable length vectors which implies that both the security header and payload fields should be parsed in order to find the beginning of the trailer field and check the validity of the signature. To parse the header and payload field vectors, the parser needs to know the length of each vector. According to ETSI ITS [12] specifications for secured message format, there are length fields at the following locations:

- The beginning of the header field vector
- The beginning of the payload vector which specifies the length of the vector. There is also another length field at the beginning of each possible payload type
- The beginning of the trailer field vector

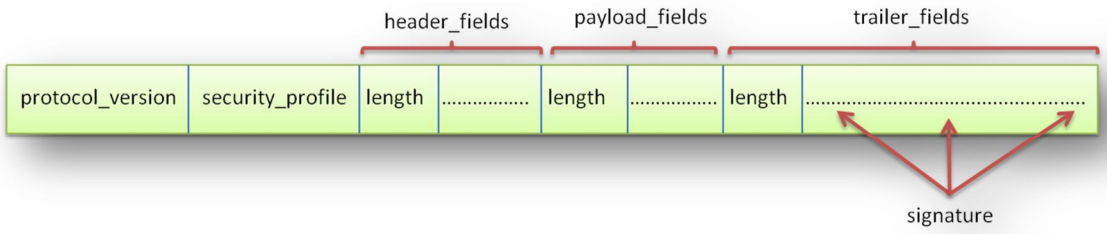

Figure 16. The position of signature in ETSI ITS secured message structure

Therefore, in order to find and verify the signature, the parser starts parsing from the beginning of the received secured message to find the certificate and then skips parsing the content of the header and payload fields using the specified length field at the beginning of each one until it finds the beginning of the trailer field.

As a result, it is not possible to check the integrity of the message without parsing the content of the message and this is a vulnerability that could be exploited by manipulating the value of the length field in any of the length fields described in this clause. This vulnerability enables an adversary to fool the parser and access the memory space out of the boundaries of the memory that is allocated to the received message. Privilege escalation, buffer overflow attack and arbitrary code execution are examples of what an adversary would be able to do by exploiting this vulnerability depending on the following conditions:

- The adversary's attack skills and level of knowledge specially regarding the architecture of the target machine
- Failure in observing secure programming guidelines when implementing data structures and the parser
- The OS layer that the parser process is running on. If the parser is run in kernel with super user privileges then the attack could be more effective.

5.5 Countermeasures

Identified software vulnerabilities

The existence of the identified vulnerabilities shown in Table 2 and Table 3 is the consequence of:

- Failure in observing the secure programming practices.
    - These vulnerabilities are all originated from programming mistakes such as lack of exception handling, input validation errors and memory safety violations.
- Failure in following the standard order of implementing procedures for encoding/decoding of sent/received data that is defined by ETSI ITS standard [44]

The following countermeasures eliminate all of the vulnerabilities identified in the implementation described in chapter 4 as well as reducing the possibility of exploiting similar vulnerabilities in future:

- Following secure programming guidelines (e.g. CERT [45]).
    - Proper use of exception handling for conditional statements and especially for parsing the content of the data fields that have dynamic definition (e.g. variable sized vectors, enums, etc.)
    - Validating the format, range and length of the data received from sources that could be either tampered by an adversary or be invalid due to hardware faults (e.g. time, GPS location) before encoding them into a new message
    - The boundaries and length of the input and output arguments should be double checked when using functions like *strncpy* and *memcpy*
- The memory allocated for data structures defined by crypto libraries (Specially the Openssl) should be freed using the functions provided by the crypto library itself
- Following the standard order of implementing procedures for encode/decode of the sent/received data as defined by ETSI ITS standard [44].
- The certificate signature verification should definitely be the first step to decode an incoming secured message that is signed
- Preventing the adversaries from manipulating the software installed on the communication device by signing the original software and verifying it every time the device boots up

Design Flaw

One solution to fix the identified flaw in the design of the ETSI ITSC security protocol concerning the position of storing the signature in a secured message is to specify a well-known position for the signature structure in the trailer field. It allows the parser to directly find the signature and parse it according to the specifications of the signature structure described in ETSI ITS [12] and finally check its validity and drop the message if the integrity check fails. As mentioned before in chapter 4 of this report, the location of the certificate containing the public keys required for the signature verification process could be found based on the value of the security profile that is always stored in the second octet of the received secured message.

A proposal for the position of storing the signature structure in an ETSI ITS secured message (containing exactly one payload) is the end of the trailer field. According to the ETSI ITS [12] the security profile of the received message specifies the symmetric algorithm used for generating the signature and the structure of the security header containing the certificate. The *field_ size* that defines the length of the vectors containing the signature point(s) can also be derived according to the symmetric algorithm. By knowing the value of the *field_size* it is possible to parse the content of the signature structure from the end of the received secured message and extract the signature to be verified using the public keys stored in the certificate.

6 Conclusion and future work

This thesis assesses the design flaws and software vulnerabilities in the ETSI ITS secured message, certificate format and identity management services by implementing and testing sign/verify services into an existing ETSI ITSC platform. The assessments identified 6 software vulnerabilities in the implementation of the Secured Message structure. The identified vulnerabilities can be exploited for DoS attack or forcing the receiver ITSC stack process to crash. The root cause of the identified software vulnerabilities in the secured message format is programming mistakes caused by misinterpretation of the standard which is a consequence of complexity and ambiguity in the standard.

Furthermore, the assessments also identified a flaw in the design of ETSI ITS security protocol concerning the position of storing the signature in a secured message as well as proposing a solution to fix it. The flaw is due to the dynamic structure of the ETSI ITS secured message which implies parsing the content of a signed message to find its signature before checking the integrity of the message. The identified flaw has critical effects on the functionality of the ITS stack and could be exploited for privilege escalation, buffer overflow attack and arbitrary code execution. Therefore, the identified flaw needs to be addressed in the newer versions of the protocol.

The software vulnerabilities have been eliminated by following the secure programming guidelines (e.g. CERT) as well as following the standard order of implementing procedures used in sign/verify services defined by ETSI ITS. Furthermore, an additional requirement or recommendation to follow secure programming guidelines should be added to the standard in order to protect the receiving ITS station from unexpected behavior caused by wrong encoding of the received packets.

Since this thesis only covers the sign/verify services as a part of the security services in ETSI ITSC, further vulnerability assessment with a focus on the other security services such as encryption/decryption and certificate management is needed.

One lesson learned from this thesis was that integrating security services into an existing ITSC stack leads to remarkable implementation difficulties and stresses the fact that security should be taken into consideration from the very beginning steps of the system design and implementation.

Bibliography

1. **ETSI TS 102 636-5-1 V1.1.1 (2011-02).** "Intelligent Transport Systems (ITS);Vehicular Communications;GeoNetworking;Part 5: Transport Protocols;Sub-part 1: Basic Transport Protocol". [Online]

2. Black-box testing. *Wikipedia.* [Online] [Cited: 13 September 2013.] http://en.wikipedia.org/wiki/Black-box_testing.

3. **ETSI TS 102 636-4-1 Ver. 1.1.1.** "Intelligent Transport System (ITS);Vehicular communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications;Sub-part 1: Media-Independent Functionality". [Online]

4. **ETSI TS 102 636-6-1:.** *"Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 6: Internet Integration; Subpart 1: Transmission of IPv6 Packets".*

5. **ETSI TS 102 637-3:.** *"Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Application; Part 3: Specification of Decentralized Environmental Notification Basic Service".*

6. **ETSI EN 302 665 V1.1.1:.** "Intelligent Transport Systems (ITS);Communications Architecture". September 2010.

7. Proof of concept. *Wikipedia.* [Online] [Cited: 20 09 2013.] http://en.wikipedia.org/wiki/Proof_of_concept#In_Security.

8. Vulnerability assessment. *Wikipedia.* [Online] [Cited: 13 September 2013.] http://en.wikipedia.org/wiki/Vulnerability_assessment.

9. **Williams, Laurie.** *White-Box Testing.* 2006.

10. **ETSI TR 102 893 V1.1.1 (2010-03).** *"Intelligent Transport Systems (ITS); Security;Threat, Vulnerability and Risk Analysis (TVRA)".*

11. **Cooper, H.** *Synthesizing Research: A Guide for Literature Reviews.* 1998.

12. **ETSI TS 103 097 Ver. 1.1.1 (2013-04).** *"Intelligent Transport Systems (ITS); Security; Security header and certificate formats ".*

13. **Offor, Patrick I.** *Vehicle Ad Hoc Network (VANET): Safety Benefits and Security Challenges.* s.l. : Nova Southeastern University, 2012.

14. **Broberg, Henrik.** *C2C cc | security model, In-Vehicle perspective.* s.l. : Volvo cars EE dep, 2012.

15. OSI model. *Wikipedia.* [Online] [Cited: 21 July 2013.] http://en.wikipedia.org/wiki/OSI_model.

16. **ETSI TS 102 723-10:.** *"Intelligent Transport Systems; OSI cross-layer topics; Part 10: Interface between access layer and network and transport layers".*

17. **ETSI TS 102 723-3:.** *"Intelligent Transport Systems; OSI cross-layer topics; Part 3: Interface between management entity and access layer".*

18. **ETSI TS 102 723-7:.** *"Intelligent Transport Systems; OSI cross-layer topics; Part 7: Interface between security entity and access layer".*

19. **ETSI TS 102 636 (all parts):.** *"Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking".*

20. **ISO/IEC 21210:.** *"Intelligent Transport Systems - Communications access for land mobiles (CALM) - IPv6 networking".*

21. **ISO/IEC 29281:.** *"Intelligent Transport Systems - Communications access for land mobiles (CALM) - Non-IP networking".*

22. **ETSI TS 102 723-11:.** *"Intelligent Transport Systems; OSI cross-layer topics; Part 11: Interface between network and transport layers and facilities layer".*

23. **ETSI TS 102 723-4:.** *"Intelligent Transport Systems; OSI cross-layer topics; Part 4: Interface between management entity and network and transport layers".*

24. **ETSI TS 102 723-8:.** *"Intelligent Transport Systems; OSI cross-layer topics; Part 8: Interface between security entity and network and transport layers".*

25. **ETSI EN 302 895:.** *"Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM) Specification".*

26. **ETSI TS 102 637-2:.** *"Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Co-operative Awareness Basic Service".*

27. **ETSI TS 102 723-5:.** *"Intelligent Transport Systems; OSI cross-layer topics; Part 5: Interface between management entity and facilities layer".*

28. **ETSI TS 102 723-9:.** *"Intelligent Transport Systems; OSI cross-layer topics; Part 9: Interface between security entity and facilities layer".*

29. **ETSI TS 102 723-6:.** *"Intelligent Transport Systems; OSI cross-layer topics; Part 6: Interface between management entity and security entity".*

30. **ETSI TS 102 637-1 V1.1.1 (2010-09).** *"Intelligent Transport Systems (ITS);Vehicular Communications;Basic Set of Applications;Part 1: Functional Requirements".*

31. **ETSI TS 102 940 V1.1.1 (2012-06).** *Intelligent Transport Systems (ITS);Security;ITS communications security architecture and security management.*

32. **ETSI TS 102 941 V1.1.1 (2012-06).** *"Intelligent Transport Systems (ITS); Security;Trust and Privacy Management".*

33. ETTE - Tekniska möjliggörare för effektiva transporter. *http://www.vinnova.se*. [Online] http://www.vinnova.se/sv/Resultat/Projekt/Effekta/ETTE----Tekniska-mojliggorare-for-effektiva-transporter/.

34. ETTE Project. *http://www.vinnova.se*. [Online] Vinnova, August 23, 2012. [Cited: July 02, 2013.] http://www.vinnova.se/PageFiles/605258335/1ETTE-projektkonfFFItrans23Aug2012.ppt.

35. *Actia.* [Online] [Cited:] http://www.actia.se/.

36. **Draft, ETSI EN 302 637-3 V1.0.0 (2012-10).** *"Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Application; Part 3: Specification of Decentralized Environmental Notification Basic Service".*

37. **ETSI TS 102 894 - 2 V0.0.46.** "Intelligent Transport Systems (ITS); Users and applications requirements; Facility layer structure, functional requirements and specifications;Part 2: Applications and facilities layer common data dictionary; ". [Online]

38. *Kapsch.* [Online] [Cited: 30 July 2013.] http://www.kapsch.net/.

39. Public-key cryptography. *Wikipedia.* [Online] [Cited: 4th August 2013.] https://en.wikipedia.org/wiki/Asymmetric_key_algorithm.

40. **FIPS PUB 186-3.** *"FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, Digital Signature Standard (DSS)".* s.l. : National Institute of Standards and Technology, 2009.

41. Google C++ Style Guide. [Online] [Cited: 1st June 2013.] http://google-styleguide.googlecode.com/svn/trunk/cppguide.xml.

42. OpenSSL: The open source toolkit for SSL/TLS. *openSSL.* [Online] [Cited: 1st June 2013.] http://www.openssl.org/.

43. Python Programming Language – Official Website. [Online] [Cited: 24th April 2013.] http://www.python.org/.

44. **ETSI TS 103 096-3 V1.1.1 (2013-07).** *"Intelligent Transport Systems (ITS); Testing; Conformance test specification for TS 102 867 and TS 102 941; Part3: Abstract Test Suite (ATS) and Protocol Implementaion eXtra Information for Testing (PIXIT)".*

45. CERT Secure Coding Standards. *CERT.* [Online] [Cited: 25 Auguest 2013.] https://www.securecoding.cert.org/confluence/display/seccode/CERT+Secure+Coding+Standards.

46. **Schoch, Elmar, et al.** *Security Headers and Formats Document.* s.l. : CAR 2 CAR Communication Consortium, Workgroup Security, 2012.

47. ETTE - Tekniska möjliggörare för effektiva transporter. *http://www.vinnova.se.* [Online] Vinnova. [Cited: 01 July 2013.] http://www.vinnova.se/sv/Resultat/Projekt/Effekta/ETTE----Tekniska-mojliggorare-for-effektiva-transporter/.

48. Project DRIVE C2X. [Online] [Cited: 28th July 2013.] http://www.drive-c2x.eu/project.

49. ESCRYPT - Embedded Security. *ESCRYPT.* [Online] [Cited: 25 07 2013.] https://www.escrypt.com/.