Security analysis on in-vehicular intelligent transportation systems

Tilmans, H.J.

Security analysis on in-vehicular Intelligent Transportation Systems

H.J. (Bert) Tilmans

Department of Mathematics and Computer Science
Eindhoven University of Technology

Graduation supervisor
dr.ir. L.A.M. (Berry) Schoenmakers (TU/e; DM)

Graduation tutors
ir. T.A. (Timo) van Roermund (NXP Semiconductors N.V.)
drs. C.M. (Kees) Moerman (NXP Semiconductors N.V.)

November 2012

Abstract

In this master thesis, we present the results of a risk analysis, performed on the generic communication architecture of an in-vehicular Intelligent Transportation System (ITS), or ITS Station. Subsequently, based on the risk analysis results, prioritised hard- and software requirements for an ITS Station are argued. These requirements are reconsidered when we partition the generic communication architectures on several hardware components. Finally, recommendations for designing an ITS Station are provided.

Acknowledgements

I wish to express my gratitude to several people who have helped me carrying out the graduation assignment. First of all I would like to thank Berry Schoenmakers and Timo van Roermund for their excellent support and supervision. Their help was very pragmatic and steered my in the right direction when it was necessary. I really appreciate their discerning views as this helped me professionalise my thesis. Also, I would like to thank Kees Moerman for his contributions.

Furthermore, I would like to thank my colleagues at NXP. They made my stay at NXP very pleasant. And, last but not least, I would like to thank my parents and my girlfriend for supporting me throughout my entire study period.

Bert Tilmans
Eindhoven, November 2012

Contents

List of Figures

List of Tables

List of Abbreviations

BSA	Basic Set of Applications
BTP	Basic Transport Protocol
CA	Certificate Authority
CAM	Cooperative Awareness Message
CC	The Common Criteria for Information Technology Security Evaluation
DENM	Decentralised Environmental Notification Message
ETSI	European Telecommunications Standard Institute
GN	GeoNetworking
HMI	Human-Machine Interface
IEEE	Institute of Electrical and Electronics Engineering
ITS	Intelligent Transportation System
ITS radio network	ITS frequency channel (5.85-5.95 GHz)
LDM	Local Dynamic Map
PKI	Public-Key Infrastructure
RSU	Road-Side Unit
SE	Secure Element
TEE	Trusted Execution Environment
ToE	Target of Evaluation
TVRA	Threats, Vulnerability and Risk Analysis
VA	Verification Accelerator
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
V2X	Vehicle-to-X (i.e, V2V and V2I)

CHAPTER 1

Introduction

Modern society strongly depends on transport, yet it comes at a price. The Europe Union (EU) counts over 40,000 casualties and 1,7 million injured persons by road transport accidents a year, of which 93% were due to human error. Also, about 10% of the road network is congested everyday, resulting in 1% GDP loss yearly [3]. And finally, road transport absorbs over 42% of EU's total oil consumption [3]. These figures show that current transport safety and efficiency are insufficient.

An intelligent transport system could increases traffic safety and efficiency. Such a system should not only combine vehicle's own sensory information, but also sensory information of surrounding vehicles and other traffic infrastructure units (e.g., traffic lights). As the vehicle topology is highly dynamic, vehicles should share their sensory information in an ad-hoc and wireless fashion. Such a system—digitally interconnecting vehicles and infrastructures—is jointly referred to as an Intelligent Transportation System (ITS).

Basically, every vehicle that contains a so-called *ITS Station* communicates on radio channels dedicated to ITS. The ITS Station will create some sort of a radio "bubble" which it will listen to and broadcast information on. Other ITS Stations, within this "bubble", will receive the broadcast information, e.g., such as periodic traffic safety messages. Periodic traffic safety messages contain vehicle conditions such as, for example, vehicle's geographical location, direction, dimension and speed. In case the ITS Station recognises a safety-critical situation, it may decide to broadcast more detailed safety messages (i.e., messages with additional information, e.g., "*Road obstacle at location* $\boldsymbol{x}$").

Furthermore, the exchange of messages not only can be used to increase traffic safety, but to increase convenience in transport as well. For example, ITS applications can reduce traffic congestion and provide info- and entertainment services to the driver. Also, driver convenience could be increased by creating so-called "vehicle trains", i.e., vehicles in such a "train" will automatically follow their predecessor at a fixed but safe distance. Of course, once secure Vehicle-to-X (i.e., vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I)) communications are at hand, many other ITS and non-ITS applications and services become possible.

Currently, ITS and ITS Stations are studied, discussed and developed by a network of Intelligent Transport Systems and Services stakeholders, such as public authorities, industry players, infrastructure operators, users organisations, international ITS associations, standardisation institutes, consortia and other organisations. Many of the stakeholders are connected to one or more umbrella foundations such as ITS Europe (ERTICO) [21], ITS America [28] and ITS Japan [29]. The ITS World Congress [30] provides a platform for global manifestation of the newest innovations, ideas and progress on ITS development.

Furthermore, the Car-2-Car Communication Consortium [15] (C2C-CC) is a non-profit industy-driven organisation, initiated by European vehicle manufacturers and supported by equipment suppliers, research organisations and other partners. The C2C-CC closely cooperates with European and international standardisation organisations, of which in particular, ETSI TC ITS. The main goal of the C2C-CC is to help development and release of an open European standard for ITS. NXP Semiconductors N.V. is one of the equipment suppliers. Aside from supplying, NXP Semiconductors N.V. also has an advisory role in the C2C-CC towards the ITS standardisation process.

However, if the security of such rich-featured ITS is insufficient, it may help people with malicious intent. This means, for example, that ITS data, such as periodic traffic safety messages, must be verifiable with respect to their integrity and authenticity. Not only at communication external to vehicles should this be verifiable, but on communication internal to the ITS Station as well. An attacker could, for example, modify traffic safety messages after they have been successfully validated with respect to their integrity and authenticity by a sub-component of the receiving ITS Station. Dozens of such threats to the internal working of an ITS Station exist and are important to be taken into account during designing ITS and ITS Stations. That is, a security analysis on the abstract and concrete ITS Station architecture is necessary.

1.1 Related Work

In previous section we pointed out the necessity of a security analysis on the architecture of an ITS Station. The European Telecommunications Standards Institute (ETSI)—which has been assigned the task to support the interoperability of co-operative systems for intelligent transport in the European community by the European Commission [18]—performed and published a Threat, Vulnerability and Risk Analysis (TVRA) on ITS [7]. This analysis provides a comprehensive set of threats, their potential undesirable consequences, corresponding risk estimations and possible countermeasures. However, for the sake of scoping, ETSI's TVRA assumes the ITS Station to be a single entity (or node) which is secure. That is, the ITS Station is assumed to be trusted and thus communications and actions performed within the ITS Station are considered to be secure. However, this is not necessarily the case. Our risk analysis withdraws this assumption and performs a TVRA, similar to ETSI's TVRA, but now focused on the ITS Station's internals. But, as currently no actual ITS Station implementation exists, the analysis is performed from the perspective of the abstract ITS Station architecture.

Furthermore, in contrast to ETSI's TVRA, the EVITA project [23] *does* focus on in-vehicle communication security. The objective of the EVITA project is to design, verify and prototype a security architecture for securing the communication between security-relevant nodes (e.g., ITS Station, human-machine interfaces, board-computer) and protection of sensitive data. Their architecture provides security features like secure tamper-resistant data storage, data encryption, node authentication etcetera. However, EVITA considers the ITS Station as a node in the in-vehicular network to which their security architecture can be "attached". But, the ITS Station consists of several components (i.e., sub-nodes), each of which have different security requirements, as will be pointed out in this thesis. For example, the ITS Station will comprise a radio transceiver on the roof of the car and a CPU near the vehicle's board computer. This gives rise to the question where to attach the EVITA security module: the radio transceiver, the CPU or to both? To answer this question, more knowledge on the internal working of an ITS Station is required. Hence, securing an ITS Station by attaching an EVITA security module is not a trivial task. Furthermore, the EVITA project provides comprehensive attack trees and potential countermeasures. However, they do not differentiate on the means (i.e., threat agents) which enable exploitation of certain vulnerabilities. Our risk analysis *does* differentiate on the threat agents per threat. The necessity for such differentiation will be explained in Chapter 4 of this thesis.

1.2 Research Goals

The main goals of this research are twofold:

i) *Analyse the requirements for an ITS Station—with respect to the information security—from the abstract perspective.* First, a risk analysis, on the abstract architecture of an ITS Station, is required to find and evaluate the threats to an ITS Station. Second, based on the threats and risk evaluation, security requirements for an ITS Station must be argued.

ii) *Reconsider the requirements, found in* i), *for two potential (hardware) implementations of an ITS Station.* The requirements, argued in *i*), must be reconsidered as the actual implementation of an ITS Station may change the requirements or even introduce new requirements.

Both research parts may help the design and developing a (more) secure ITS Station.

1.3 Thesis Outline

In the next chapter we will provide an overview on the “global” ITS architecture and Chapter 3 will describe the architecture of a vehicular ITS Station. In Chapter 4 we will discuss the applied risk analysis methodology, the attacker model and present the risk analysis results (of which the risk estimations can be found in Appendix A). Then the hard- and software requirements are argued for the logical architecture in Chapter 5 and for the hardware partitioning in Chapter 6. Finally, we will recapitulate the findings of the risk analysis and provide design recommendations for ITS Stations in Chapter 7.

CHAPTER 2

Global ITS Architecture

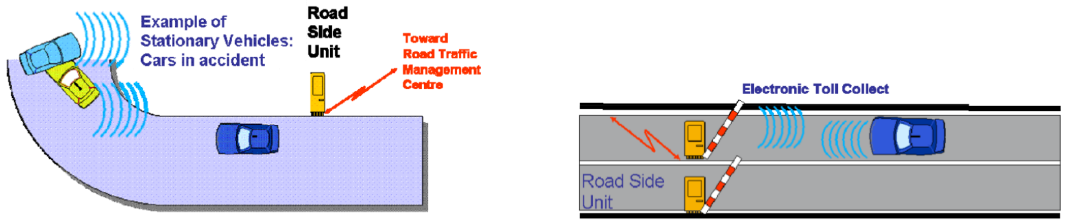
As discussed in the introduction, traffic in Europe should become safer, more efficient and more convenient. To achieve this, ITS must be designed, developed and penetrate the vehicle market. The American Institute of Electrical and Electronics Engineering (IEEE) developed and approved an amendment to the *IEEE 802.11* standard, the so-called *IEEE 802.11p* [25], to add Wireless Access in Vehicular Environments (WAVE). This WAVE standard will be the base for interoperability between ITS-compatible vehicles, i.e., vehicles equipped with an ITS Station. Furthermore, IEEE and ETSI will standardise higher protocol stack layers and corresponding functionalities as well. IEEE and ETSI cooperate in the ITS standardisation process, but both will publish their own version of the standards. Although their standards do not differ fundamentally (i.e., in design and security principles), they do in terminology. To prevent confusion, this thesis uses the ETSI ITS standards as it is viewed from a European perspective.

2.1 System Goals

ITS has the following (high-level) goals.

- *Traffic safety.* ITS must increase the effectiveness of traffic accident prevention and mitigation mechanisms. Where current mechanisms base their decisions on own in-vehicular sensory information only, ITS will enable such decision-making on basis information from surrounding vehicles as well; situational awareness will increase.

- *Traffic efficiency.* Aside from the impact to traffic safety, traffic congestion also financially impacts on transport in Europe. Hence, traffic must become more efficient. ITS will enable broadcast of more detailed and targeted traffic management information. Ad-hoc traffic management (not involving authorities) will be enabled as well. For example, vehicles could spread information on the vehicle density or traffic lights could broadcast advices for green waves. Traffic efficiency could also be improved by systems such as, for example, a virtual “train” of vehicles. Such “train” enables vehicles to virtually “attach” to its predecessor (which, in return, may be attached virtually to its predecessor as well) and automatically follow the predecessor at a safe distance but with the same speed. Basically, it is a more



(a) Stationary vehicle warning (from [5][Figure C.8]) (b) Electronic toll collect (from [5][Figure C.32])

Figure 2.1: Two ITS use cases

intelligent cruise-control system that adapts to the speed of your predecessor. Of course, many other intelligent applications become possible, such as, for example, over-the-air toll-payments.

- *Driver convenience.* Driver convenience is improved by safer and more efficient traffic. The virtual "train" (as explained at the previous goal) also increases driver convenience, as the attached vehicle will automatically follow its predecessor, hence not requiring further involvement of the vehicle driver.

- *Environmental impact.* Minimising environmental impact, such as, for example, exhaust gasses, is one of the goals of ITS as well [22]. More efficient traffic flow, i.e., shorter routes and less traffic congestion, will be the most significant factor in decreasing environmental impact.

- *Commerce.* As ITS will only be really effective if a significant amount of vehicles are ITS-compatible, implementing ITS in vehicles must be an attractive effort to vehicle manufacturers. This means, vehicle manufacturer's customers must be willing to pay extra for ITS-compatibility. However, safety and security are hard to sell; people generally prefer new convenience features (e.g., multimedia services) over new safety and security measures (e.g., ABS). Hence, ITS should implement commercial services as well, such as, for example, in-vehicle Internet access and info- and entertainment services.

2.2 Some Use Cases

ETSI discussed [5] some use cases for ITS, of which two are depicted in Figure 2.1a and 2.1b. Figure 2.1a is a safety-critical traffic situation which can be detected (before the vehicle driver could do) and a potential crash can be avoided. Figure 2.1b is an example in which ITS can help improve traffic efficiency, in this case, by enabling electronic wireless toll payments.

2.3 Entities

ITS enables vehicles to communicate with other vehicles and infrastructure units (e.g., traffic lights). Figure 2.2 sketches the "global" ITS architecture and shows all relevant entities for this security

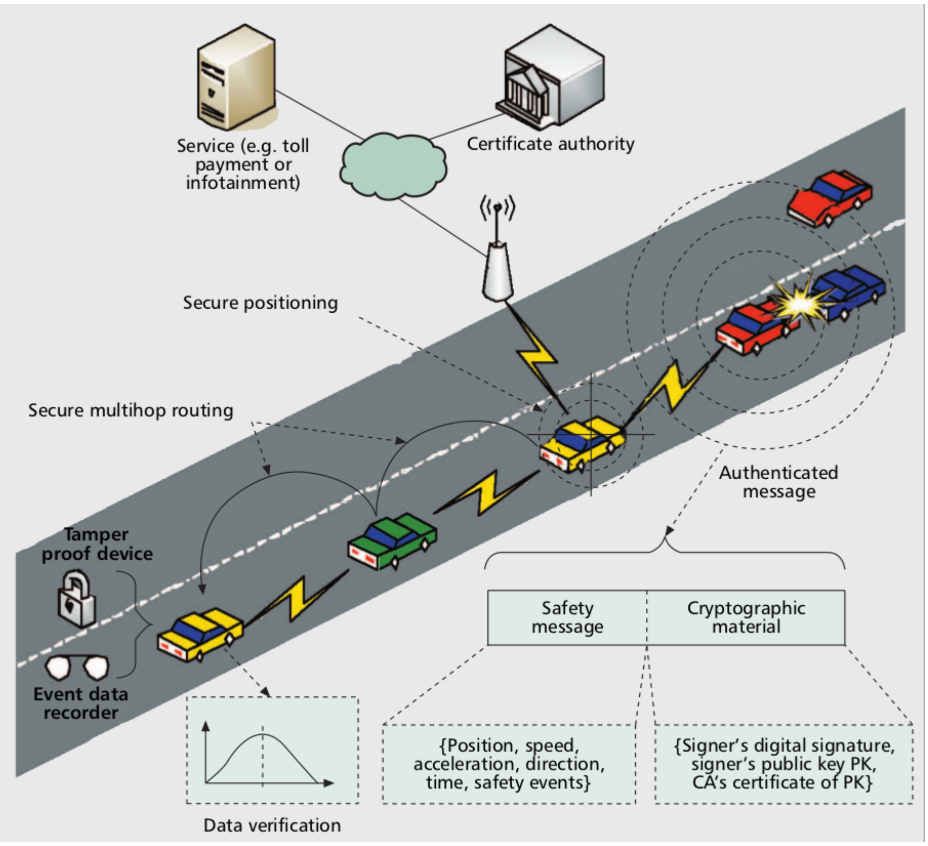

Service (e.g. toll payment or infotainment)
Certificate authority
Secure positioning
Secure multihop routing
Tamper proof device
Event data recorder
Authenticated message
Safety message
Cryptographic material
{Position, speed, acceleration, direction, time, safety events}
{Signer's digital signature, signer's public key PK, CA's certificate of PK}
Data verification

Figure 2.2: Intelligent Transport System (ITS) Overview (from [33, Figure 4])

analysis; *vehicles*, *Road-Side Units* (RSUs), *Certificate Authorities* (CAs) and *(third-party) service providers.*

- Vehicles
  ITS-compatible vehicles contain a unique ITS Station (which will be defined and discussed in Chapter 3). With this ITS Station the vehicle can communicate with other ITS-compatible vehicles and ITS-compatible RSUs, e.g., traffic lights. During vehicle production, all necessary technical and formal processes, corresponding to the ITS Station, are fulfilled. That is, the vehicle is registered at authorities as is done currently as well, but additionally, cryptographic attributes are installed and bound to that vehicle-specific ITS Station. These cryptographic attributes will enable assurance of integrity and authenticity of ITS data, e.g., messages. For example, digital signatures on ITS messages—i.e., messages broadcast by the ITS Station—allow other (surrounding) ITS-compatible vehicles or RSU's to verify the ITS message integrity and authenticity.

- Road-Side Units
  RSUs are also equipped with an ITS Station and have a fixed geographical location. However, the main difference between a vehicle and a RSU (aside from their geographical locations) is the privileges within ITS. RSUs may have different privileges, such as, for example, the privilege to broadcast traffic light specific warning messages. An RSU could be contained in many different devices, e.g., such as traffic lights, traffic and toll road barriers and info- and entertainment service access points.

- Certificate Authorities
  In order enable verifiability of integrity and authenticity of received ITS messages, and to enable non-repudiation of received or transmitted messages, all ITS messages are digitally signed. The trust in the digital signatures is established by the implementation of a public-key infrastructure hierarchy (see Figure 2.3). In this hierarchy, the root certificate authority (root CA) is the—trusted—security anchor. The root CA issues certificates to long-term certificate authorities and pseudonym certificate authorities, both of which issue certificates for the end users (i.e., ITS Stations) in turn. Long-term certificates will never be exposed on the ITS radio network and will be used to request a set of short-term certificates. The short-term certificates are used to digitally sign ITS messages that will be broadcast on the ITS radio network. However, the ITS Station may select a short-term certificate. This enables privacy as surrounding vehicles can not link a short-term certificate to a particular vehicle. However, the periods between a change of short-term certificate is crucial to the level of privacy it provides [14].

  - *Root Certificate Authority.* The root CA is the security anchor for data integrity and authenticity preservation in ITS and is comparable to root CAs as used for establishing an HTTPS Internet connection. All ITS Stations will securely store the certificate of one or more root CAs they trust, which will be used to verify integrity and authenticity of "lower-level" certificates, e.g., such as certificates of long-term CAs, short-term CAs and ITS Stations.
  - *Long-term Certificate Authority.* A long-term CA issues certificates for ITS Stations. The certificate of a long-term CA—which is signed by one or more root CAs—is used to generate unique long-term certificates for ITS Stations. Long-term certificates are

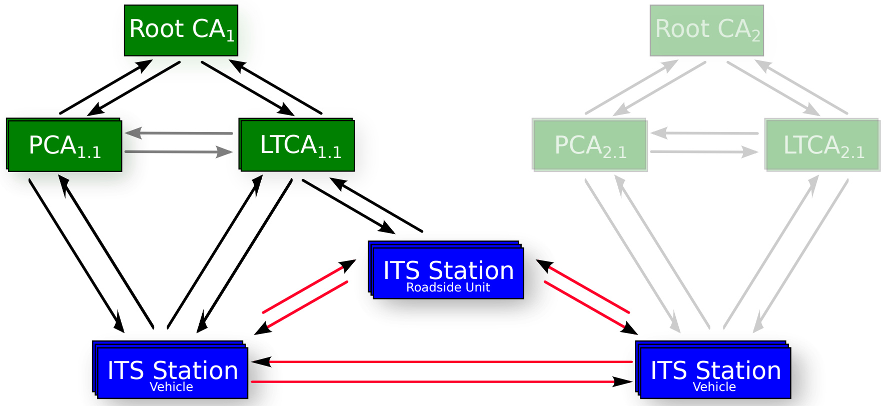


Figure 2.3: Public-key infrastructure for ITS

ITS Stations-specific and will only be used to request anonymous short-term certificates for ITS Stations. That is, long-term certificates are never exposed on the ITS radio network [1].

- *Short-term Certificate Authority.* A short-term CA, or pseudonym CA, is comparable with a long-term CA, but will only issue anonymous short-term certificates, or pseudonym certificates, for ITS Stations. ITS Stations will use pseudonym certificates to sign outgoing data and to verify integrity and authenticity of received communication. Although the link between the long-term certificates and the short-term certificates should not be exposed on the ITS radio network for privacy reasons, the link may be exposed for liability reasons. For example, misbehaving vehicles will need to be revocable from the ITS network.

- Other Authorities Other ITS authorities are public agencies or corporations with administrative powers and responsibilities, e.g., authorities for issuance and management of ITS Station user identities and credentials.

2.4 Entity Intercommunication

The standard of choice for ITS traffic safety and efficiency-related (radio) communication is *802.11p*[25].

The frequency range 5.85-5.96 GHz will be dedicated to ITS communications. This frequency band contains five sub-bands, three of which are dedicated to safety-related communication (so-called *ITS G5A* frequency band) and two of which to non-safety-related communication (so-called *ITS G5B* frequency band). However, in this security analysis, the focus is on safety-related communication only. Hence, for the rest of this thesis *ITS radio network* refers to the ITS G5A radio frequency band.

[1] The ITS radio network will be explained and discussed in the next section.

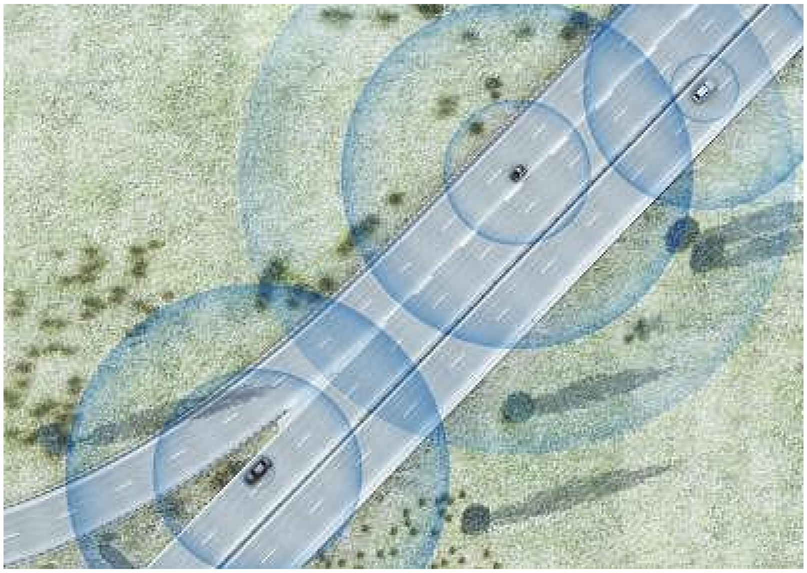


Figure 2.4: Radio “bubbles” for Vehicle-to-X communication

Basically, all ITS-compatible vehicles broadcast messages on the ITS radio network, thereby creating some sort of radio “bubble” (see Figure 2.4), with a radius of around one kilometer; creating an ad-hoc network. Message types can be divided into two groups, *single-hop* and *multi-hop* messages. With single-hop messages, other ITS-compatible vehicles, within the range of the message originator’s “bubble” will receive, interpret and act upon the broadcast messages. With multi-hop messages, receiving vehicles may also decide to forward the message, thereby increasing the range of a message. Multi-hop messages could, for example, warn vehicles on traffic jams.

The different types of ITS-related messages and the architecture of an ITS Station will be explained and discussed in the next chapter.

CHAPTER 3

ITS Station Architecture

The ITS-Station is not only the bridging component between vehicles and infrastructures, it also interprets and acts upon the ITS messages. Consequently, the ITS Station will not only implement a communication interface (e.g., for ITS radio network connectivity), but also a local virtual map, containing the vehicle's "situational awareness" which safety-related ITS applications act upon.

3.1 Reference Architecture

ETSI defined [6] an ITS reference architecture, or communication protocol stack for ITS (see Figure 3.1), which follows the principles of the well-known OSI model (ISO/IEC 7498-1) for layered communication protocols. OSI layers 1 and 2 are mapped to the *Access Layer*, OSI layers 3 and 4 to *Network Layer* [1] and OSI layer 5, 6 and 7 are mapped to the *Facility layer*. The *Application Layer* presents the ITS Station applications that employ ITS Station services. The *Management Layer* manages communication internal to the ITS Station. The *Security Layer* provides security services to all protocol stack layers. Basically, the Management and Security Layers are vertically aligned to emphasise the fact that both layers are an integral part of all protocol stack layers.
All protocol stack layers comprise a set of tasks and services, which will be described here briefly. More details on the tasks and services can be found in [6].

- *Access layer*. The Access Layer consist of two sub-layers. The physical layer (PHY) which physically connects the ITS Station to communication meda (e.g., ITS radio network). And the data link layer (DLL) which in turn comprises two sub-layers; the medium access control (MAC), for managing access to the communication medium, and logical link layer (LLC). More information about the (physical aspects of the) Access Layer can be found in [4].

- *Network layer*. The Network Layer implements one or several networking protocols (e.g., IPv6) and one or several transport protocols (e.g., UDP).

- *Facility layer*. The Facility Layer provides application support, information support, communication support and session support. The Facility Layer at least provides the following

[1] In fact, ETSI defined the Network layer as *Networking & Transport layer*, but for readability reasons, this layer is referred to as *Network Layer* in this thesis.

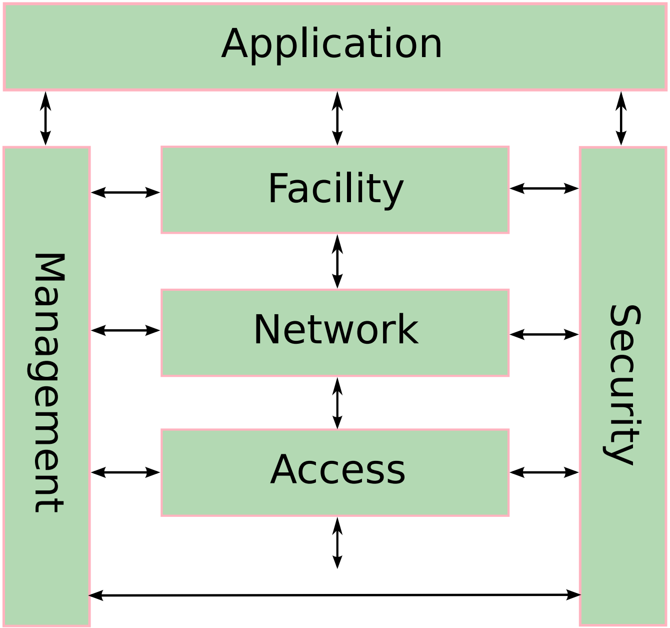


Figure 3.1: ITS Station reference architecture

functionalities or services:

- *Local Dynamic Map (LDM).* Maintaining a virtual map of the (estimated) situational awareness, which traffic safety-relevant applications will act upon. This map may include lane-specific information (e.g., curbs, pedestrian walking, bicycle paths) and road furniture (e.g., traffic signs and traffic lights). Furthermore, all dynamic objects that are directly sensed or indicated by other ITS Stations will be included in the LDM as well.
- *Message management.* Management of event-triggered messages, periodic messages and service messages (which will be discussed in Section 3.3).
- *Data representation.* Message coding and decoding.
- *Position and time.* Providing geographical positions of the ITS Station and the actual (local) time.

- *Application layer.* The Application Layer contains road safety, traffic efficiency and other ITS applications, which will employ the information presented by the Facility Layer.
- *Security Layer.* The Security Layer at least provides the following functionalities or services:
  - Secure processing and storage (e.g., by a hardware security module)
  - Authentication, authorisation and profile management.
  - Management of identities, cryptographic keys and certificate.
  - Firewall and intrusion management.

- *Management Layer.* The Management Layer at least provides the following functionalities or services:
  - Management of cross-interfacing, networking, communication, ITS applications, ITS Station and more.
  - Management of general congestion control.

3.2 Security Features

Security is an integral part of ITS message processing. In general, all protocol stack layers will apply and verify the plausibility and trustworthiness of ITS messages, however, each by different mechanisms and on different parts of the ITS message contents.

As explained in Chapter 2, integrity and authenticity of safety-related messages (broadcast on the ITS radio network) are cryptographically protected by a digital signature.

- *Access layer.* The Access Layer adds and verifies an error detection code (i.e., checksum value of the message) for detection of transmission errors. Note, this does not detect malicious modifications to messages as an attacker could change the message and recompute the checksum value.
- *Network layer.* The Network Layer adds and verifies digital signatures of ITS messages and may encrypt or decrypt (non-safety related) ITS messages. Due to radio bandwidth restriction, C2C-CC agreed on applying only one digital signature per ITS messages, instead of applying one digital signature per protocol stack layer (which would have enabled stronger security guarantees on ITS messages at all protocol stack layers).
- *Facility layer.* The Facility Layer estimates the plausibility of ITS message by reflecting message contents to the current estimated situation (presented by the LDM). Furthermore, it manages credentials and authorisation tickets (for granting ITS messages, processes or users access to certain ITS Station services).
- *Application layer.* The Application Layer contains ITS applications, e.g., such as traffic safety- and efficiency-related applications. Safety-related applications, for example, identify safety-critical situations in traffic, on basis of information that the Facility Layer provides.
- *Management Layer and Security Layer.* Actually, previous protocol stack layers do not implement the security and management mechanisms, they invoke the such mechanisms at the Management Layer or Security Layer. However, to indicate at what protocol stack level which security-related operations are performed, we mapped the operations to corresponding layers.

3.3 Message Types

ETSI differentiates between three types of ITS messages.

Periodic messages

Every ITS-compatible vehicle will periodically broadcast so-called Cooperative Awareness Messages (CAMs). CAMs are broadcast at a 1-10 Hz rate (depending on the traffic situation) and provide information of vehicle presence, positions, movement, sensor information and more to neighbouring ITS Stations (i.e., ITS Stations within the radio "bubble"). These messages are single-hop, which means that they will not be forwarded. Furthermore, CAMs are broadcast using the GeoNetworking communication protocol, which is defined in [10] and which will be discussed later in this chapter. More information on CAMs can be found in [9].

Event-triggered messages

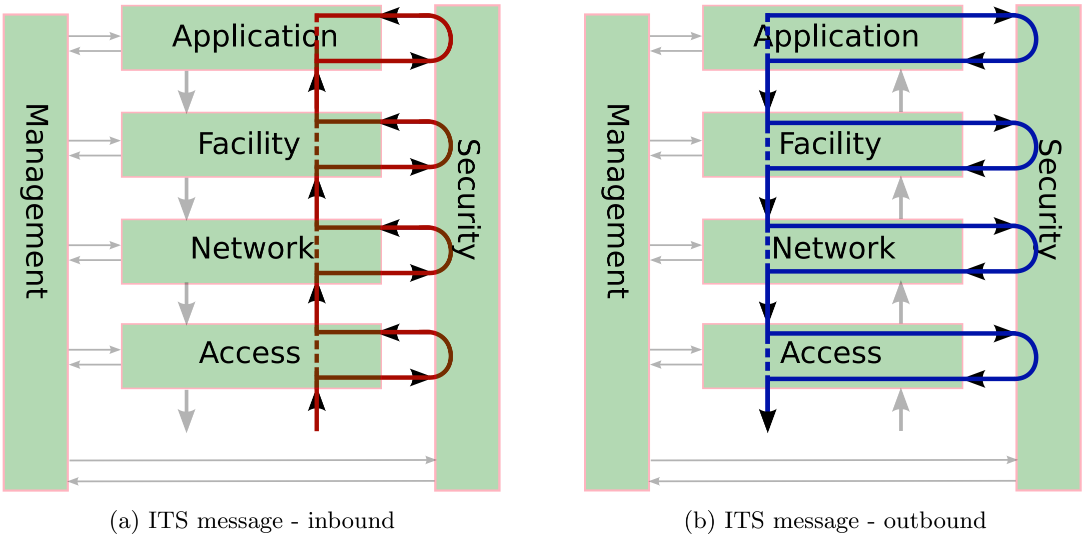
Some traffic or vehicular events (e.g., collision warning) may trigger broadcasting event messages, which typically are Decentralized Environmental Notification messages (DENMs). Event-triggered messages contain management information (e.g., event evolution or reliability level), situational information (e.g., event cause, type and time criticality) and location information (e.g., (estimated) event positions and event notification relevance area). More details on DENMs in [8]).

Service messages

Connections to (third-party) services are managed via service messages. These messages are not relevant to traffic safety, they are not broadcast on the ITS radio channel, but on the ITS G5B radio channel. More details on such message can be found in [27]).

3.4 Message Flow

In general, ITS messages are *inbound* (i.e., received from the ITS radio network) or *outbound* (i.e., broadcast on the ITS radio network). The red arrow in Figure 3.2a depicts the flow of inbound ITS messages the blue arrow in Figure 3.2b depicts the flow of outbound ITS messages. The dashed lines indicate the possibility of every protocol stack layer not to perform any (or all) security-related actions on the ITS messages. Also, the ITS messages may be forwarded first before their trustworthiness is verified, e.g., for performance reasons.



(a) ITS message - inbound

(b) ITS message - outbound

3.5 Message Formats

ETSI released a basic set of applications (BSA) [5] which an ITS Station must have implemented, such as, for example, *active road safety*, *cooperative traffic efficiency*, *cooperative local services* and *global Internet services*. The ITS applications use information that is provided by ITS messages, e.g., such as CAMs. Figure 3.2 depicts the CAM message formats per protocol stack layer. Furthermore, “Digested CAM” is the CAM data after it has been decoded and processed by the Application Support Service at the Facility layer.

ITS messages for BSA support, e.g., such as the CAM, are encapsulated by a Basic Transport Protocol (BTP) [12], which in turn is encapsulated by a GeoNetworking [10] (GN) message.

Most important observation from Figure 3.2 is that the Network Layer applies and verifies digital message signatures (“Msg sig.”). C2C-CC agreed on applying and verifying the digital signatures of ITS messages at the Network Layer.

3.5.1 Hop Counter

GeoNetworking message headers are not completely secured by the digital signature; some information is mutable (e.g., the `hop-limit` for multi-hop communication) and therefore can not be signed. Such information is put in the “GN header” instead of the “GN secured header”. Or can the `hop-limit` be secured? Although resolving this issue not really falls into the scope of the security analysis on the ITS Station’s internal working, it may help to improve to make the GeoNetworking standard more robust against attacks.

To limit the range of an ITS message (i.e., the geographical area that receives a (forwarded) ITS message), a `hop-limit` (comparable with the `Hop Limit` in IPv6 or `TTL` in IPv4) will be present in all such messages [11]. If the `hop-limit` has not reached the value 0 yet, every forwarding ITS Station (hereafter referred to as a node) decrements the `hop-limit` by 1, else the message must

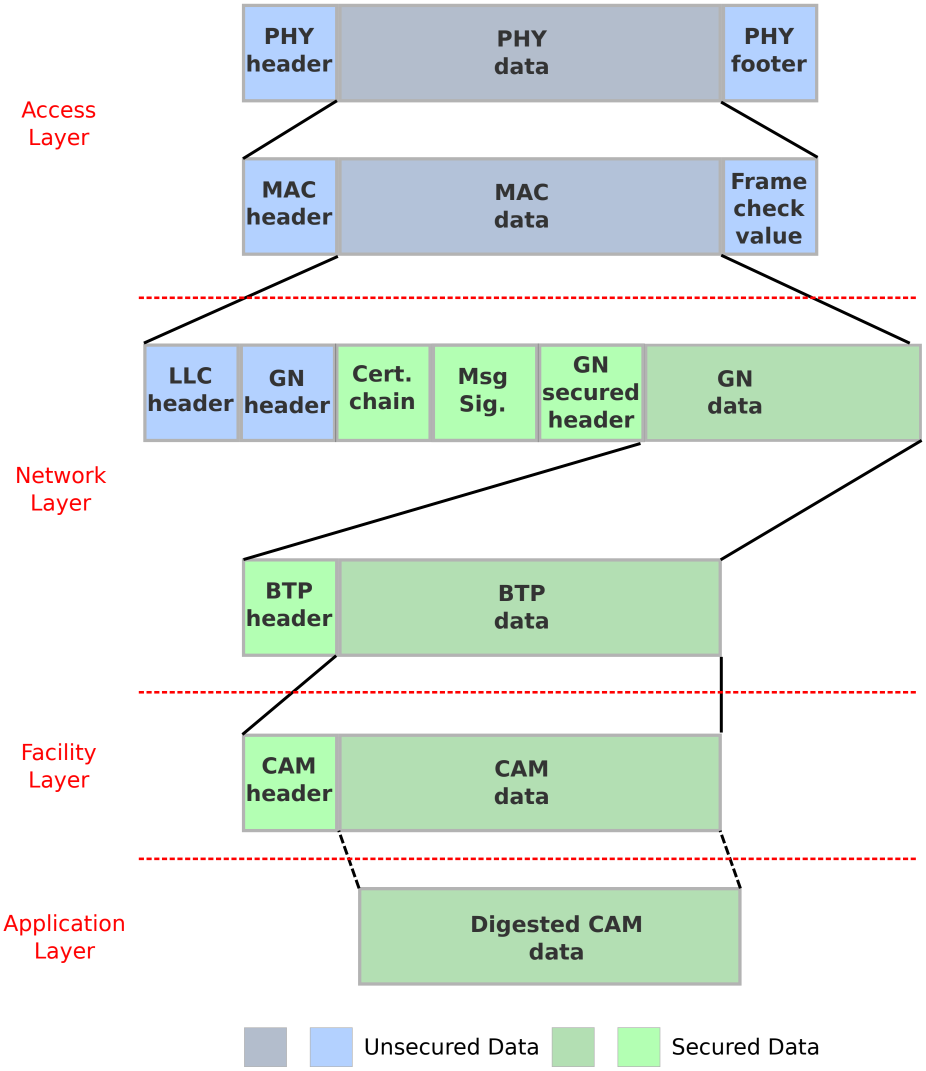


Figure 3.2: Message formats for a CAM

not be forwarded. But, as the `hop-limit` is mutable information, it cannot be signed.

Clearly, a malicious node can originate and generate an ITS message, containing an excessively high `hop-limit` to flood the ITS network. Intermediate and destination nodes can could mitigate the risk by applying plausibility checks on the message, but this will require additional computational resources. Furthermore, a malicious intermediate node could carry out the following attacks [37]:

- Increment the *hop-limit*
  - In case of broadcasting safety-related ITS messages, the `hop-limit` is only utilised for range limitation (instead of also providing route discovery information). But, malicious incrementing the `hop-limit` increase the range of the ITS message, thus inflicting unnecessary bandwidth usage and affecting availability of the ITS network.
  - Seemingly decreasing the travelled path of the message—i.e., the number of times it is forwarded—which affects routing protocols, e.g., such as AODV Routing [34]. Whenever such routing protocol determines the most efficient route based on the number of hops (contrary to Geographic (Position-Based) Routing protocols) an attacker thereby can increase the chance of being included in the selected route. Being part of a selected route could enables message eavesdrop, injection, modification and deletion.
- Decrement the *hop-limit* 0 or more than 1 times.
  - Maliciously decrementing the `hop-limit` could cause the message not to reach the (complete) intended destination area (i.e., area of relevance for the message), thereby affecting availability of ITS messages.
  - Seemingly increasing the travelled path of the message, which affects routing protocols (which base the most efficient route on the number of hops). Hence, an attacker decreases the change of being included in the selected route. Not part of a selected route could affect availability of ITS messages at other ITS Stations. Although an ITS Station could also just discard the ITS message, decrementing the `hop-limit` will be harder to detect by neighbouring ITS-compatible vehicles.

To secure the `hop-limit` against malicious modifications, it must be assured that this `hop-limit` can only and must only be decremented exactly once by forwarding nodes.

Zapata et al. proposed a solution [37], based on a hash-chain and digital signatures, to protect the `hop-limit` from maliciously increases. The hash-chain is used for gaining irreversibility of the `hop-count` (i.e., cryptographically secured from increasing). The digital signature (which is already present for ITS messages), enables verifiability of integrity of the `hop-limit`. Nevertheless, maliciously decrementing the `hop-limit` is not protected by this mechanism.

In the proposed solution, the message originator selects the intended number of maximum hops, `Max_HL`, randomly selects a secret seed, `s`, $\texttt{s} \in_R \mathbb{R}$, computes the "top" of the hash-chain, `H_Top`, by hashing `s` `Max_HL` times ($\texttt{H\_Top} = \mathcal{H}^{\texttt{Max\_HL}}(\texttt{s})$) and computes the first (pre-image) of `H_Top`, $h = \mathcal{H}^1(\texttt{s})$. Now, the message originator adds includes `H_Top` and `Max_HL` to the digital signature, and includes `H_Top`, `Max_HL`, `h` and the hop limit (i.e., the actual "hop counter"), $\texttt{hl} = 0$, to the message. Note, the integrity and authenticity of `H_Top` and `Max_HL` is assured by the digital signature.

On reception, the receiver checks if $\mathcal{H}^{\texttt{Max\_HL-hl}}(\texttt{h}) = Top\_HL$. If this equality holds, hl has not been decreased en route, assuming no one but the message originator knows `s`.

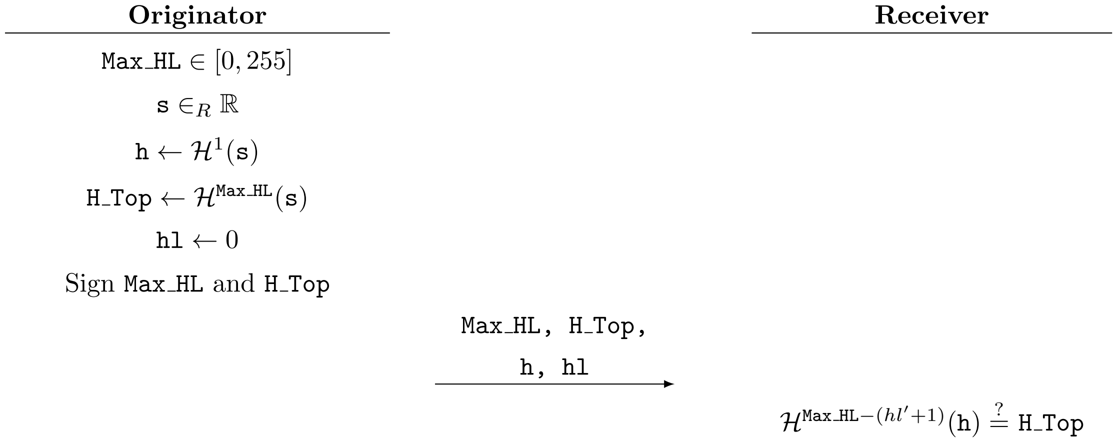

Originator
Receiver
$\texttt{Max\_HL} \in [0, 255]$
$\texttt{s} \in_R \mathbb{R}$
$\texttt{h} \leftarrow \mathcal{H}^1(\texttt{s})$
$\texttt{H\_Top} \leftarrow \mathcal{H}^{\texttt{Max\_HL}}(\texttt{s})$
$\texttt{hl} \leftarrow 0$
Sign Max_HL and H_Top
Max_HL, H_Top,
h, hl
$\mathcal{H}^{\texttt{Max\_HL}-(hl'+1)}(\texttt{h}) \overset{?}{=} \texttt{H\_Top}$

Figure 3.3: Communication protocol for securing the `hop-limit`

For example, assume A would like to communicate with B and sends a corresponding routing request (as defined in AODV [34]) to B, with Z_1, Z_2 forwarding the message sequentially to B (i.e., $A \to Z_1 \to Z_2 \to B$). Furthermore, lets assume A sets $\texttt{Max\_HL} = 4$. Then Z_1 received the unsigned (mutable) values `hl'`, `h'` and the signed (immutable) values $\texttt{Max\_HL} = 4$, `H_Top`. Z_1 then computes $\texttt{hl''} \leftarrow \texttt{hl'} + 1$ and checks if

$$\begin{aligned}
\mathcal{H}^{\texttt{Max\_HL - hl''}}(\texttt{h}) &= \mathcal{H}^{4-1} \\
&= \mathcal{H}(\mathcal{H}(\mathcal{H}(\texttt{h}))) \\
&= \mathcal{H}(\mathcal{H}(\mathcal{H}(\mathcal{H}(\texttt{s})))) \\
&= \mathcal{H}^4(\texttt{s}) \\
&\overset{?}{=} \mathcal{H}^{\texttt{Max\_HL}}(\texttt{s}) \\
&= \texttt{H\_Top}
\end{aligned}$$

If the equality holds, Z_1 computes $\texttt{h''} = \mathcal{H}(\texttt{h})$, replaces `hl',h'` by `hl',h''` respectively and forwards the message. Figure 3.3 depicts the communication protocol for securing the `hop-limit`.

Although an adversary could still maliciously increment `hl` multiple times or keep it remain unmodified [24], when `Max_HL` is protected by the message digital signature, integrity of the “intended hop limit” is still verifiable.

What are the implications on the performance and capacity of the ITS radio channel? `Max_HL` requires one byte extra and `h`, `Top_HL` both require one time the size of the applied hash algorithm output, per ITS message. Note that `hl` would be included in the original (unprotected) set-up form already, hence does not introduce additional bytes. Thus, the message size increases with one byte plus two times the size of the applied hash output. If, for example, *SHA-256* is the applied hash algorithm, the message size increases with $2 * 32 + 1 = 65$ bytes, which is around 25% of a typical ITS safety message ($\sim$256 bytes)

Previous analysis indicates that *Zapata*’s method will introduce an inapplicable overhead safety-related ITS messages. However, this method could still be applied to improve security of ad-hoc routing protocols that base decisions on the *hop-limit*.

However, Zapata's solution could be improved by implicitly including `H_Top` to the message, hence saving 32 bytes on the ITS message. Receiving nodes then first extract `H_Top` from the message by calculating

$$\mathcal{H}^{\texttt{Max\_HL - hl'}}(\texttt{h}) = \texttt{H\_Top'}$$

Now, the message receiver includes `H_Top'` to the received message (at a standardised location of the message) and verifies the digital signature subsequently. Whenever the digital signature is in accordance with the provided ITS message data and calculated `H_Top'`, the receiver knows that $\texttt{H\_Top'} = \texttt{H\_Top}$, and, moreover, that `hl` is not decremented.

This method saves around 50% (i.e., 32 bytes) on Zapata's method, thereby not increasing typical ITS messages with around 25% (i.e., 65 bytes) but with around 12.5% (i.e., 33 bytes). However, the latter method still requires any receiving nodes to hash $\texttt{h}$ $\texttt{Max\_HL} - \texttt{hl}$ times. As the current ETSI standard reserved one byte for the (unprotected form of the) hop counter [10], a forwarding node may need to hash $\texttt{h}$ $\texttt{Max\_HL} - \texttt{hl} = 255 - 1 = 254$ times. Although the computational impact may be negligible when hashing is hardware accelerated, the increase in message size will probably not be negligible. In particular not for periodically broadcast safety messages (e.g., CAMs), which are broadcast at 1-10Hz rate, during crowded traffic.

Further research is required to determine the exact impact on the ITS applications when the `hop-limit` is not secured.

Furthermore, route discovery protocols could also use the GPS location of ITS Stations. However, to secure such routing protocols, the GPS locations of intermediate nodes would need to be secured as well.

Finally, the GeoNetworking protocol [10] also defines a `TTL` value, which defines the remaining time-to-live of an ITS message, in milliseconds. Although this `TTL` could be used to limit the range of an ITS message as well, it is not included in the digital signature either. Intermediate nodes will decrease the `TTL` value and therefore cannot be included in the digital signature either. However, if all ITS Stations would have a synchronised clock, every ITS Station could compute the remaining `TTL'` by subtracting the current time with the time of signature generation (i.e., time that the ITS message was signed). Whenever for the resulting value, `TTL'`, it does not hold that $0 < \texttt{TTL'} \leq \texttt{TTL}$, the message should be discarded (as the time-to-live has exceeded).

CHAPTER 4

Risk Analysis

In this chapter, we will discuss the applied risk analysis methodology, the attacker model and the analysis results.

4.1 Approach

In order to obtain a better view on the security risks for an ITS Station, a Threat, Vulnerability and Risk Analysis (TVRA) is performed on the logical architecture (i.e., on the 'level' of protocol stack layers, as described in Figure 3.1). In this chapter we discuss the applied risk analysis methodology, its scope (Target of Evaluation), corresponding assets and the attacker model. In Section 4.5 we will discuss the results of the security analysis.

To avoid potential misunderstandings on the meanings of *assets*, *countermeasures*, *risks*, *threats*, *treat agents* and *vulnerabilities*, their definitions [1] are listed below.

Asset. Anything that has value to the organisation, its business operation and its continuity.

Countermeasure. Countermeasures protects the asset(s) against threats.

Risk. The potential that a given threat will exploit vulnerabilities of one or more assets, thereby causing harm to the organization.

Threat. Potential cause of an unwanted incident that affects one or more assets.

Threat agent. Anything that can exploit a vulnerability.

Vulnerability. A weakness of one or more assets that can be exploited by one or more threats.

4.2 Method

To comply with a widely accepted method for performing this Threat, Vulnerability and Risk Analysis (TVRA), this thesis applies the guidelines as specified in ETSI's *Threat, Vulnerability and*

Risk Analysis; Method and Proforma [1]. These guidelines are based on the guidelines, specified in *ISO/IEC 15408*, which in turn complies with the guidelines, specified in *The Common Criteria for Information Technology Security Evaluation* [19] (CC). Hence, by applying ETSI's TVRA methodology, the risk analysis results can be used in the context of a CC evaluation. The risk analysis will result in a risk estimation for each threat. Risks evaluate in one of the following three levels:

- Minor. *No essential assets are concerned, or the attack is unlikely. Threats causing minor risks have no primary need for countermeasures.*
- Major. *Threats on relevant assets are likely to occur although their impact is unlikely to be fatal. Major risks should be handled seriously and should be minimized by the appropriate use of countermeasures.*
- Critical. *The primary interests of the providers and/or subscribers are threatened and the effort required from a potential attacker's to implement the threat(s) is not high. Critical risks should be minimized with highest priority.*

The main difference between ETSI's and CC's methodologies is that CC assumes the system design being complete, thus the analysis assesses the risks of *already mitigated* threats. Whereas the purpose of ETSI's TVRA is to identify vulnerabilities that require countermeasures, by assessing the risks of threats which *are not mitigated* yet.

We intend our risk analysis to be compliant with ETSI's methodology. However, some deviations from it were necessary, as will be motivated now.

Opportunity factor

The "opportunity" of an attack is the time window that access to the asset is required to execute the attack, as well as the number of required assets. An example of an asset is a smart card that contains confidential information. Before the confidential information can be extracted maliciously, perhaps multiple smart cards are required to learn the internal working of the smart card. The opportunity values are

- *Not needed.* ("unnecessary").
- *Less than one day, or, number of assets* ≤ 10*.* ("easy".
- *Less than a month, or, number of assets* ≤ 50*.* ("moderate").
- *At least a month, or, number of assets* ≤ 100*.* ("difficult").
- *Not possible.* ("none").

As will be discussed in Section 4.5, the most important assets of the ITS Station are correctness of software and the integrity and authenticity of data. Moreover, ITS messages are public information, as they are broadcast on a public radio channel. Consequently, many evaluations of the opportunity factor would results in *easy*, thereby losing the granularity that this factor could provide to the risk estimations.

In our opinion, the attack location (in the ITS Station) strongly determines the opportunity. An attack which requires access to, for example, CPU's internals is less opportune than when only access on the communication interface between two hardware components is required.

The CC states that there are multiple methods of representing and quantifying the attack factors (e.g., such as the opportunity factor), and also, that other factors may be applicable for particular ToE [19, p. 422].

For the purpose of our risk analysis, we rather speak of "accessibility" than of "opportunity". Hence, the factor opportunity is renamed and (partly) redefined:

- *Not needed* ("unnecessary").
- *Less than one day, or, access between protocol stack layers* ("easy").
- *Less than one month, or, access between protocol stack layer components* ("moderate").
- *At least a month, or, access on/in protocol stack layer component(s)* ("difficult").
- *Not possible* ("none").

Intensity factor

Similar to previous deviation to ETSI's methodology, the "intensity" factor of successfully executed attack states the number of affected assets ("instances"). Possible intensity "values" are

- *1 instance.*
- *Moderate number of instances.*
- *High number of instances.*

However, ETSI TVRA considers one ITS Station as one instance. Although attacks on the ITS Station (internals) may affect multiple (surrounding) ITS Stations, it is more interesting to know what the impact of an attack at a certain place inside the ITS Station would be. That is, will the impact of the attack propagate to other protocol stack layers and will such attack be detectable? Hence, for the purpose of our risk analysis, the intensity "values" are redefined to

- *Affects only this protocol stack layer.*
- *Affects multiple protocol stack layers, but is detectable.*
- *Affects multiple protocol stack layers and is not detectable.*

Threat agent

ETSI TVRA *does not* differentiate on the way a certain attack is performed, i.e., by which threat agent. However, in our opinion, this is a crucial aspect to take into account when assessing the risks of a certain threat. For example, ETSI TVRA states that the required knowledge to execute a successful "Denial of access to incoming messages"-attack is *restricted*, i.e., it requires "knowledge, shared under a Non-Disclosure Agreement among developer organisations". However, such attack could easily be performed by a radio jammer which only requires knowledge of publicly available information [1]. Also, ETSI categorised the "required equipment" to perform such attack as *specialised*, i.e., "not readily available, but can be acquired without undue effort". However, radio jammers (with frequency agility) are readily available on the market [36]. Whenever ETSI would

[1] ITS radio network and corresponding communication standards are publicly available in international standards.

have differentiated between threat agents for each threat, it would result in more accurate risk estimations and, consequently, risk removal or mitigation could be prioritised more accurate.

Our risk analysis *does* differentiate between threat agents, when assessing the risk of a threat.

Attack locations

ETSI TVRA does not differentiate on the place in the ITS Station where the attack is performed. For example, "modification of transmitted ITS messages" is considered to be a "Critical" risk, which is an accurate risk estimation if ITS messages are not secured against malicious modifications. However, in our risk analysis, ITS messages are assumed to be digitally signed before being broadcast. This means, the digital signature will be verified at some place in the ITS Station. Modification of ITS messages before this verification process will be detected at the verification, but modifications of ITS messages after the verification process will go undetected, if no countermeasures are taken.

Hence, for the purpose of our risk analysis, we differentiate on the place of the attack in the ITS Station, i.e., between the protocol stack layers Access Layer, Network Layer, Facility Layer, Application Layer, Management Layer and Security Layer. Furthermore, each communication interface is considered to be bi-directional, i.e., we do not differentiate between in- and outgoing messages per protocol stack layer. The analysis of a protocol stack layer includes all of its communication interfaces, which may lead to some redundancy in the analysis for adjacent protocol stack layers.

4.3 Target of Evaluation

The Target of Evaluation (ToE) is the reference architecture for the ITS Station (as described in Chapter 3). Figure 4.1 depicts the ToE with the considered attack interfaces IDs (red coloured). Note, not only ITS messages will be communicated within the ITS Station, also messages for cross-layer management, station management and more [6]. In this analysis, such messages are referred to as "management messages".

Furthermore, a Human-Machine Interface (HMI) may be connected to the ITS Station, e.g., for notifying the vehicle driver on the ITS Stations' findings or actions, as well as for possible reconfigurations of the ITS Station. The HMI will be "connected" to the ITS Station through the Facility Layer. This connection is not depicted in Figure 4.1. However, in the analysis, communication over this connection is referred to as "user notifications" or "user configuration messages".

4.4 Adversary Model

As the attacker could gain physical access to ITS Station(s), it may tamper with any associated hardware component and is assumed to be capable of eavesdropping, injecting, modifying and deleting any data contained in the ITS Station. The attacker is only limited by the security measures, e.g., such as cryptographic measures or tamper-resistant hardware boxes.

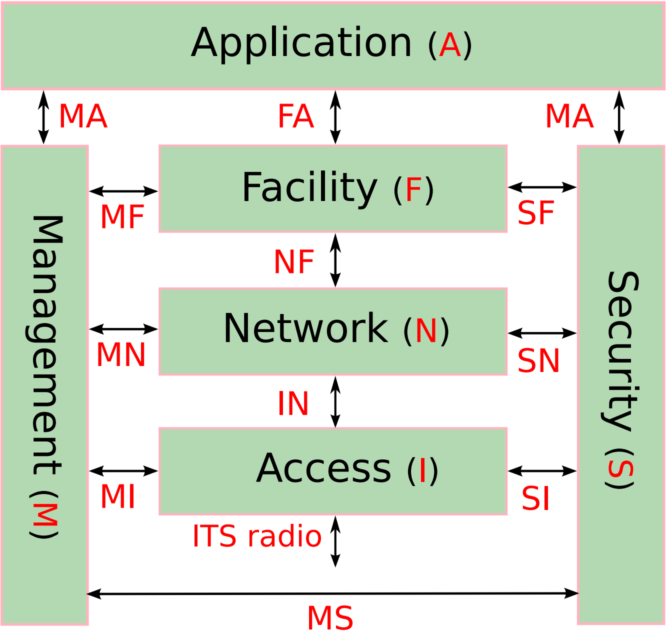


Figure 4.1: Target of Evaluation and attack locations

4.5 Results

In this chapter, we summarise the results of a Threats, Vulnerability and Risk Analysis (TVRA). Appendix A lists the risk estimations. But first, the considered assets, threats and threat agents are listed.

4.5.1 Assets

Crucial to traffic safety and efficiency is the correct operation of the ITS Station, i.e., the software must be able to respond correctly and adequately to traffic situations. Although multiple manufacturers will introduce ITS-compatible Stations (in the future), the Stations may differ on the exact implementation. Hence, for this analysis we abstractly consider an ITS Station to be a hardware box that comprises ITS-related processes and process data. Consequently, correctness, trustworthiness and confidentiality of the processes and process data may need to be assured.

In the risk analysis, the following assets are considered.

1. *Correctness of processes.* Incorrect processes may lead to incorrect or inadequate responses to traffic safety- or efficiency-related situations, even if, for example, trustworthiness of ITS messages is assured. Hence, correctness of processes must be assured.

2. *Trustworthiness of process data.* Untrustworthy data (e.g., maliciously modified ITS messages) may lead to incorrect or inadequate responses to traffic safety- or efficiency-related

situation, even if processes are correct. Hence, trustworthiness, i.e., integrity, authenticity and plausibility, of process data must be assured. [2]

3. *Confidentiality of data.* Clearly, confidential information, i.e., secret cryptographic material and personal information, should not be compromised. Compromise of the former affects assets 1 and 2, the latter will affect privacy. Also, compromise of manufacturer confidential information, e.g., algorithms or other processes, may affect manufacturer's businesses. Hence, confidential information must not leak.

4.5.2 Threats

The following threats have been indicated and are considered in our risk analysis. Note, insecure ITS Station components refers to incorrect or malicious ITS Station components. Furthermore, some threats may enable other threats. For example, execution of insecure firmware may enable modification of stored data, and vice versa.

1. **Execution of insecure firmware.** In this analysis, firmware refers to software on a (combination of) protocol stack layer(s) which is pre-installed during manufacturing. The firmware operates the hardware and is comparable with "kernel-space" software. Furthermore, firmware is considered to have unconditional access to all processes and data of the corresponding protocol stack layer. Hence, when executing insecure firmware at any protocol stack layer, assets 1, 2 and 3 of that particular protocol stack layer could be compromised. Also, assets at other protocol stack layers may be affected, for example, when the Security Layer leaks its private signing key (affecting asset 3), the trustworthiness provided by the digital signature is compromised (affecting assets 1 and 2). Hence, the ITS Station must be assured to execute secure firmware.

2. **Execution of insecure applications.** In this analysis, applications refer to software that is executed at the Application Layer, which is (pre-)installed by the manufacturer or a third-party. Application runs "on top" of the Application Layer firmware and are comparable to "user-space" software. In general, applications have have less privileges than firmware. Hence, execution of insecure applications could affect assets 1, 2 and 3, but not necessarily. Furthermore, assets at other protocol stack layers could be affected as well, for example, when the Application Layer incorrectly alerts of a traffic safety-related situation, resulting in incorrect broadcast of event-triggered ITS messages, possibly leading to accidents. Hence, the ITS Station must be assured to execute secure applications.

3. **Modification of stored data.** Firmware and applications may store data. Incorrect modifications of stored data could lead to incorrect execution of software. Vice versa, incorrect software could lead to incorrect modifications of stored data as well. Both threats could affect assets 1, 2 and 3. For example, an attacker could modify the memory unit containing the firmware to execute insecure firmware, therefore circumventing default firmware installation procedures. But, an attacker could also modify an application memory unit to inject user notifications, thereby circumventing processes which generate such notification. Hence, integrity, authenticity and confidentiality of the ITS Station must be assured.

[2] Note, authenticity implies integrity [31], but for clarity reasons, both properties are considered explicitly. Moreover, security measures such as checksum values do provide integrity of data, but not authenticity. Hence, explicit discussion of integrity and authenticity properties is required sometimes.

4. **Injection, modification or deletion of messages en route** ITS messages and management messages—exchanged between protocol stack layers—can be injected, modified or deleted, thereby possibly affecting 1, 2 and 3.

5. **Complete, selective or random DoS to processes** A Denial of Service to hardware or software may be the result of hard- or software corruption and will affect asset 1. Note, injection, modification and deletion of messages could also lead to a DoS to certain hard- or software, thereby possibly affecting assets 1, 2 and 3. For example, if all signature verification requests, sent to the Security Layer, are deleted, then the Network Layer is denied access to the Security Layer.

6. **Eavesdropping** All previous threats could lead to disclosure of confidential information which affects asset 3 (clearly), as well as assets 1 and 2 (e.g., when digital signatures are be compromised as well).

7. **Repudiation of ITS message transmission or reception** Safety-critical events will be logged for liability reasons, hence require trustworthiness assurance. Any of the previous threats could lead to compromise of such log-files, thereby possibly affecting assets 1, 2 and 3.

4.5.3 Threat Agents

In principle, any attack can originate from any logical entity in the ToE environment (see Figure 4.1) and may target any of the assets (see Section 4.5.1). A threat agent is the means that enables the exploitation of a vulnerability, possibly leading to one or more of the threats that have been indicated in previous section. Note, a threat agent may results in a threat, but also "create" a different threat agent. For example, injection of firmware updates by injecting messages may lead to the execution of insecure firmware. On the other hand, execution of insecure firmware may lead to malicious modifications to stored data. Nevertheless, although threat agents may overlap, they are considered independently in the risk analysis.

Furthermore, attack locations refer to the locations as depicted in Figure 4.1. Attack interfaces "S_x" and "M_x" refer to all interfaces to the Security Layer and Management Layer respectively [3]. Furthermore, the following terminology (defined in [1]) is used for the estimations of the *development difficulty* of a threat agent.

- *Standard.* Readily available (possibly part of asset).
- *Specialised.* Not readily available, but can be acquired without undue effort.
- *Bespoke.* Not readily available, and requires high costs or is highly specialised.

The following terminology (defined in [1]) is used for the estimations of the *utilisation difficulty* of a threat agent.

- *Unnecessary.* Access to the asset is not needed.
- *Easy.* Access to the asset for less that 1 day and attack interface is between ITS protocol layers.

[3] S_x and M_x both refer to interface MS. This redundancy does not affect the analysis, hence can be neglected.

- *Moderate.* Access to the asset for less that 1 day and attack interface is between ITS protocol layer component(s).
- *Difficult.* Access to the asset for multiple days and attack interface is on/in ITS protocol layer component(s).
- *None.* No opportunity exists.

The following threat agents are considered in this thesis.

1) **Radio jammer.** Any kind of equipment, capable of jamming the ITS radio network.

Attack location	ITS radio network
Threats	1,2: When over-the-air software updates are jammed (i.e., blocked), 4: Deleting (i.e., jamming) messages en route (while in transport over ITS radio network), 5: Making ITS radio network (partly) unavailable
Development difficulty	*Standard*: Radio jammers with frequency agility are readily available [36].
Utilisation difficulty	*Easy*: Whenever the targeted ITS Station(s) are within range of the radio jammer.

2) **Radio transceiving equipment.** Any kind of radio transceiver, capable of communicating with ITS Stations, including genuine but insecure ITS Stations.

Attack locations	ITS radio network (any radio transceiver), all interfaces (genuine but insecure ITS Stations).
Threats	1,2: e.g., masquerading genuine software author for over-the-air updates. 3: sending outdated ITS messages. 4: e.g., replaying or forwarding outdated ITS messages. 5: ITS radio network saturation. 6: e.g., following a particular vehicle and observe its pseudonym certificate changes
Development difficulty	*Specialised*: Genuine ITS Stations require little modifications to become insecure. Non-genuine ITS Stations can be obtained without undue effort as all development standards, as well as hardware, are publicly available.
Utilisation difficulty	*Easy*: Whenever the targeted ITS Station(s) are within range of the radio transmitting equipment.

3) **Malicious (non-ITS Station) equipment.** We define the following three non-ITS Station equipment, i.e., equipment not part of a genuine ITS Station.

   a) **Stored data monitoring or influencing device.** Any kind of device, capable of monitoring, inserting, modifying or deleting stored or being processed data.

Attack locations	ToE environment, all interfaces.
Threats	1,2: if a software installation can be modified. 3: by definition. 5: e.g., when corrupting critical firmware data. 6: e.g., when reading plaintext data, stored in external memory. 7: e.g. when corrupting log-files.
Development difficulty	*Specialised.* Even when no or insufficient security measures for these threats are applied, an attacker would still require familiarity with the used ITS Station hard- and software.
Utilisation difficulty	*Difficult.* Whenever physical access to the targeted ITS Station is gained. Note, the actual execution of an attack may require significantly less time of physical access to the targeted ITS Station(s) than (is available) during development of the threat agent.

b) **Man-In-The-Middle (MITM) equipment.** Any kind of device, capable of monitoring, inserting, modifying or deleting messages en route, i.e., message communicated between protocol stack layers.

Attack locations	ITS radio network, IN, NF, FA, M_x, S_x.
Threats	1,2: e.g., when modifying software updates en route. 4: by definition. 5: e.g., when deleting message to the Security Layer. 6: by definition. 7: by definition.
Development difficulty	*Specialised.* Genuine ITS Stations require little modifications to become insecure. Other non-ITS Station equipments which are capable of modifying in-ITS Station communication can be developed without undue effort and costs.
Utilisation difficulty	*Moderate.* ITS radio network: whenever the targeted ITS Station is within range of the radio transceiving equipment. Between ITS protocol stack layers or ITS protocol stack layer components, whenever physical access to the targeted ITS Station(s) is gained. Nevertheless, attack that require physical access to the targeted ITS Station(s) require the developer of the exploit and threat agent to be familiar with the hard- and software of the the ITS Station.

c) **Physical environment monitoring or influencing equipment.** Any kind of device or environmental parameter (e.g., magnetic fields), enabling monitoring or influencing any kind of parameter in the ToE environment. For example, an attacker may deduce confidential information from a electromagnetic side-channel [16].

Attack locations	All attack locations, including the ToE environment
Threats	All. For example, data may get modified (1,2) corrupted (3,4,5) or eavesdropped (6, and ITS Station components may become malfunctioning (7).
Development difficulty	*Bespoke.* Although the device may not be sophisticated, to discover which actions lead to a "good" result will require a significant amount of time (except for threat 7).
Utilisation difficulty	*Difficult.* Physical access to a genuine ITS Station is required during development, as well as during attack execution. However, the actual execution of an attack may require significantly less time of physical access to the targeted ITS Station(s).

4) **Stolen or malfunctioning ITS Station (component(s)).** Any stolen (genuine) ITS Station (component).

Attack locations	I, N, F, A, M, S
Threats	1,2: e.g., when stealing a memory unit of an emergency vehicle, possibly containing digital certificate with more privileges. 3: replacing any data by interchanging ITS Station components of different ITS Stations. 5: removing ITS Station components will typically result in a malfunctioning ITS Station. 6: Stealing/copying the memory unit which comprises Pseudonym Certificates of the attacked ITS Station will affect its privacy. 7: e.g., by removing the memory unit which comprises log-files.
Development difficulty	*Moderate.* Stealing or destructing ITS Station components is fairly easy, once physical access to the targeted ITS Station is gained. However, to exploit a stolen ITS Station component, the attacker must be familiar with corresponding hard- and software.
Utilisation difficulty	*Moderate.* Once physical access to the targeted ITS Station is gained, stealing or destructing particular ITS Station components is easy to moderate, depending on the type of components (e.g., an external memory unit is stolen more easily than an in-CPU component.

5) **Counterfeit, insecure ITS Station (component(s)).** Any unauthentic (insecure) ITS Station (component) that can substitute corresponding component(s) of the targeted ITS Station.

Attack locations	I, N, F, A, M, S
Threats	All threats, depending on the counterfeited component.
Development difficulty	*Bespoke.* The attacker is required to be familiar with the to be counterfeit ITS Station component(s).
Utilisation difficulty	*Moderate.* Once physical access to the targeted ITS Station is gained, the substitution is still considered to be relatively difficult.

6) **Insecure application.** Any insecure application, executed at the Application Layer. Note, execution of insecure applications is a threat, but could also be the means to any of the defined threats.

Attack locations	A.
Threats	1: if the application exploits a vulnerability in the Application Layer firmware. 2: by definition, possibly affecting other applications as well. 3,4,5,6: e.g., when the application processes confidential information or generates user notifications.
Development difficulty	*Bespoke.* Development environments for third-party applications may be publicly available, which will make development of applications significantly easier. However, to find an exploitable vulnerability in the Application Layer software will require the attacker to be familiar with the Application Layer, and possibly its firmware as well.
Utilisation difficulty	*Easy.* Whenever an attacker is able to install applications by default installation procedures, installation of insecure applications will be easy. *Moderate.* Whenever the attacker circumvents the default installations procedures.

7) **Insecure firmware.** Insecure firmware at any protocol stack layer. Note, execution of insecure firmware is a threat, but could also be the means to an other threat.

Attack locations	I, N, F, A, M, S.
Threats	All threats, as it gives unconditional privileges to the assets at the attacked protocol stack layer. Assets at other protocol stack layers may be affected as well.
Development difficulty	*Bespoke* Development of insecure firmware that can be installed or injected is considered to be very time-consuming and in-depth knowledge of the internal working of the targeted ITS Station (protocol layer) is required.
Utilisation difficulty	*Easy.* Installation of insecure firmware by using default installation procedures is considered to be easy, but may be detectable. *Difficult.* Installation of insecure firmware by circumventing default installation procedures (i.e., “firmware injection” is considered to be very difficult as it will require physical access to the targeted ITS Station.

CHAPTER 5

Security Analysis - Logical

In this section, the threats and estimated risks (see Appendix A) will be discussed, which will lead to certain requirements for removing or mitigating the risks.

5.1 Preliminaries

In the risk analysis, certain assumptions and observations are made, which will support certain requirements. Some requirements are protocol stack layer specific and will be denoted as

Requirement <protocol stack abbreviation >.<requirement number>,

e.g., **Requirement F.2** denotes requirement 2 for the Facility Layer.

Furthermore, as explained in Chapter 4, risks are valued either *Minor*, *Major* or *Critical.* Based on the risk estimations, this risk analysis will argue and prioritise requirements. Prioritisation is conform the *MoSCoW* method [17]. That is,

- *Minor* risks map to a requirement that COULD be fulfilled.
- *Major* risks map to a requirement that SHOULD be fulfilled.
- *Critical* risks map to a requirement that MUST be fulfilled.

5.2 Analysis

For the risk analysis we assume the following:

Assumption 1. *ITS messages are digitally signed before broadcast on the ITS radio network.*

Assumption 2. *Digital signatures include a time stamp of generation time.*

Assumption 3. *Digital signatures for ITS messages are generated and verified the Network Layer level of the protocol stack.*

Assumption 4. *If necessary, ITS messages are encrypted/decrypted at the Network Layer level of the protocol stack. Note, safety-related ITS messages are public and therefore need not to be encrypted/decrypted.*

Assumption 5. *Only encrypted confidential information may be broadcast on the ITS radio network.*

Additionally, the following definitions apply to the risk analysis. Other definitions will be described along the way of the analysis.

Definition 1. Software *refers to firmware and applications.*

Definition 2. Insecure software *is incorrect, i.e., containing design flaws or implementation faults, or is malicious. Note, insecure firmware does not necessarily lead to a compromise of one of the assets.*

Definition 3. *Software is considered* installed *whenever default installation procedures* are not *circumvented.*

Definition 4. *Software is considered* injected *whenever default installation procedures* are *circumvented.*

Definition 5. *ITS Station components are the hardware parts that (partially) implement one or more ITS protocol stack layers.*

5.2.1 Access Layer

5.2.1.1 Insecure firmware

The risk of execution of insecure firmware at the Access Layer is *Minor*, as the Access Layer does not act upon the contents of the ITS message. Except for the threat agent 'Radio transceiver' which has a *Major* risk, as masquerading a genuine software author for over-the-air software updates is easier than by the other threat agents.

For inbound communication, the Access Layer converts the analog signals to a digital form and, consecutively, forwards the MAC data of the (digital) ITS message to the Network Layer (see Figure 3.2). For outbound communication, the Access Layer performs the reverse operation.

Observation 1. *The Access Layer only converts between the (analog) ITS radio network and the (digital) in-ITS Station network, unconditionally of the contents of higher protocol stack layer messages. Hence the Access Layer can be considered as (part of) the ITS radio network.*

Because of Observation 1, the Access Layer firmware has lower security requirements higher protocol stack layer firmware(s). Yet, if would be beneficial to prevent tampering with the Access Layer's functions.

Requirement I. 1. *Access Layer firmware COULD be protected against malicious modifications.*

5.2.1.2 Modification of stored data

Taking into account Observation 1, malicious modifications to stored data (in the Access Layer) is *Minor* risk. Hence, stored data could be secured against malicious.

Requirement I. 2. *Data, stored at the Access Layer, COULD be protected against malicious modifications.*

5.2.1.3 Message injection, modification or deletion

Taking into account Assumptions 1 and 3 and Observation 1, ITS message injections or modification are detectable at higher the Network Layer.

However, the Access Layer could delete ITS messages, which will go undetected at higher protocol stack layers. But, deletion of messages could also occur en route, i.e., at the ITS radio network interface. Hence, it is considered to be a *Minor* risk.

Also, the Access Layer could inject ITS message by replaying recorded ones. Hence, replayed ITS messages must be detectable at the higher protocol stack layers, e.g., by using the time stamp that is present in each digital signature of the ITS message (see Assumption 2).

Observation 2. *Each digital signature of an ITS message contains a time stamp, which can be verified at the higher protocol stack layers. Hence, no additional mechanisms for replay detection are required at the Access Layer.*

Furthermore, the Access Layer could maliciously inject, modify or delete (over-the-air broadcast) software updates for higher protocol stack layers. As the risks of execution of insecure firmware at the higher protocol stack layers is estimated to be *Major* or *Critical*, integrity and authenticity of software updates must be assured during the download phase (i.e., while in transit over the ITS radio network).

Requirement 1. *Integrity and authenticity of software updates MUST be assured when transmitted on the ITS radio network.*

5.2.1.4 Complete, selective or random DoS to processes

By Observation 1, the Access Layer is considered as (part of) the ITS radio network. Hence, only services that use the ITS radio network can be denied access to.

A DoS to the ITS radio network is easily established by wire-cutting, damaging Access Layer components or radio jamming. As the first two methods are mainly hardware issues, tamper-resistant boxes will sufficiently mitigate such threats. Also, the ITS Station user could be notified of a tampering attempt or malfunctioning ITS Station component by performing periodic built-in self-tests (BISTs), e.g., during boot-time of the ITS Station.

Definition 6. *A device is considered to be* tamper-resistant *if device tampering will not be beneficial for the attacker.*

Definition 7. *A device is considered to be* tamper-evident *whenever unerasable evidence of device tampering will be present.*

Requirement 2. *The ITS Station components SHOULD be encapsulated in a tamper-resistant hardware box. Additionally, the hardware box could be tamper-evident.*

Also, damages or misconfiguration of the Access Layer could lead to a DoS of the ITS radio network. Hence, integrity of the Access Layer must be verified during boot time and run time of the ITS Station.

Requirement 3. *The ITS Station MUST perform a check on hardware and software malfunctions or misconfiguration during boot time (i.e., during a so-called "cold boot"). Also, it MUST notify the ITS Station user on the results of the check. Additionally, if Requirement 2 is fulfilled, the tamper-evidence COULD be part of the check procedure.*

Note, the Access Layer configuration could also be modified maliciously by injecting, modifying or deleting Management Layer messages . This will be discussed later, in Section 5.2.6.3

Radio jammers and radio transceivers could cause a DoS to ITS radio network as well, e.g., by saturating the ITS radio network channels. ETSI discussed some countermeasures to these threats in [7], but unfortunately none are applicable as they requiring severe redesign of the Access Layer (standards).

Observation 3. *ITS Stations, as currently standardised, are not resistant to radio jammers or radio transceivers that saturate the ITS radio network.*

5.2.1.5 Eavesdropping

By Assumption 5 and Observation 1, the Access Layer will not process plaintext confidential information, leading to a *Minor* risk for eavesdropping. Hence it does not require additional measures to assure information confidentiality assurance.

Observation 4. *The Access Layer will not process plaintext confidential information, hence does not require security measures to prevent leakage of confidential information.*

Furthermore, traffic safety-related ITS messages are not confidential as they are intrinsically public. Hence, traffic safety-related ITS messages do not require confidentiality.

Observation 5. *Safety-related ITS messages are intrinsically public, hence do not require confidentiality assurances.*

However, as will be discussed next, safety-critical ITS messages (e.g., DENMs, but not CAMs), will be logged. As log-files will differ for every ITS Station, knowledge of the log-files may enable unique identification of vehicles. Hence, log-files should be protected from eavesdropping, e.g., by encryption.

Requirement 4. *Confidentiality of log-files SHOULD be assured, e.g., by encryption.*

5.2.1.6 Repudiation of message transmission or reception

All protocol stack layers may maintain a log-file of transmitted and received messages and processes for liability reasons, statistics generation or ITS Station maintenance purposes. However, ETSI requires only traffic safety-critical ITS messages (e.g., DENMs, but not CAMs) to be logged for reducing the required memory space [7]. Nevertheless, unauthorised users or processes should not be able to modify the log-files.

Requirement 5. *The log-files MUST be protected against malicious modifications.*

5.2.2 Network Layer

5.2.2.1 Insecure firmware

Execution of insecure Network Layer firmware gives unconditional privileges at the Network Layer, which is a *Major* risk. However, as the unconditional privileges enable threats with *Critical* risks, the Network Layer firmware must be protected from malicious modification. As this threat is estimated to be a *Critical* risk when firmware can be updated over-the-air, this must be protected against malicious modifications.

Requirement N. 1. *Network Layer firmware MUST be protected against malicious modifications.*

The Network Layer interacts with the Security Layer for encryption/decryption and digital signature generation/verification purposes.

For outbound communication, the Network Layer could inject corrupt, but digitally signed, ITS messages. As the Access Layer will not verify the contents of higher protocol stack layer messages (see Observation 1), this will go undetected at the originating or forwarding ITS Station.

For inbound communication, the Network Layer could inject malicious messages, e.g., by modifying or discarding signature verification results. By Assumption 3, integrity and authenticity of ITS messages is not assured by a digital signature anymore after the Network Layer performed the signature verification. Hence, integrity and authenticity of messages must be verifiable by other means at the Facility Layer and Application Layer. This could be achieved by appending the verification results as an attribute to the verified message. Clearly, integrity and authenticity of the verification attribute must be assured as well.

Definition 8. *A* verification attribute *is an attribute that contains the verification results of an inbound ITS message in a manner that assures integrity and authenticity.*

Requirement N. 2. *The verification results of a digital signature MUST be attached to corresponding ITS message in a* verification attribute*.*

Although the Network Layer will check the relevance of inbound ITS messages on basis of the GeoNetworking message headers , it will not act upon the BTP data (see Figure 3.2). Hence, malicious outbound ITS messages will go undetected at the Network Layer and Access Layer.

Observation 6. *The Network Layer will check the relevance of inbound ITS messages based on GeoNetworking headers but based on the* BTP data *(see Figure 3.2). Hence, maliciously injected or modified* BTP data *at higher protocol stack layers will go undetected for outbound ITS messages.*

5.2.2.2 Maliciously modifying stored data

Even if all previous requirements are fulfilled, process data can still be modified. Modifications to process data is considered to be a *Major* risk. Hence, data, stored at the Network Layer, should be secured against malicious modifications.

Requirement N. 3. *Data, stored at the Network Layer, SHOULD be be secured against malicious modifications.*

5.2.2.3 Message injection, modification or deletion

Even when all previous requirements are fulfilled, messages could still be injected, modified or deleted by a MITM attack on the communication between protocol stack layers. As this threat is considered to be *Critical*, integrity and authenticity of messages exchanged between protocol stack layers must be assured.

Definition 9. *An* authenticated message *has its integrity and authenticity assured.*

Definition 10. Message validity *is the determination of the message relevance (to an ITS Station or ITS Station components).*

Requirement 6. *Messages exchanged between protocol stack layers MUST be authenticated and MUST be verifiable on their validity.*

Also, even when all previous requirements are fulfilled (secure firmware, authenticated messages, etc.), messages could can still be deleted or corrupted by a MITM attack between protocol stack layers. As this is considered to be a *Critical* risk, the threat of deleting or corrupting message, during transmission, must be mitigated [1]. Note, as communication between protocol stack layers most likely will not be resistant to rigorous tampering (either by a MITM attack or by influencing environmental parameters), complete or random deletion of messages will always be possible.

Observation 7. *The threat of complete or random deletion of messages can not be removed completely as severe tampering to the communication interface(s) is difficult to resist.*

However, selective deletion or corruption of ITS messages based on their contents requires the attacker to be capable of reading the message contents; messages must be made indistinguishable. In order to mitigate the risk of an attacker deleting messages based on their ciphertext, probabilistic encryption must be used. When probabilistically encrypting two identical messages, both ciphertexts will be different. More information on probabilistic encryption can be found in [31].

Requirement 7. *Messages exchanged between protocol stack layers MUST be made indistinguishable, e.g., by probabilistic encrypting.*

[1] Note, deleting messages could lead to inaccessibility of certain protocol stack layers, which will immediately result in a DoS

5.2.2.4 Complete, selective or random DoS to processes

The risk of a DoS by hardware tampering or destruction will be mitigated by Requirements 2 and 3. The risk of a DoS by deleting messages on the communication channels will be mitigated by Requirement 7 and by invalidating messages signature validation results by Requirement N.2. Insecure software that causes a DoS, will be mitigated by Requirement N.1.

5.2.2.5 Eavesdropping

The Network Layer changes pseudonym certificates for digital signature generation every once in a while. [2] However, when the set pseudonym certificates for a specific ITS Station is known, privacy of that ITS Station is lost.

Observation 8. *Although pseudonym certificates are non-confidential information, knowledge on the (sub-)set of pseudonym certificates of an ITS Station will enable tracking of that ITS Station and will affect privacy.*

Hence, pseudonym certificates should be stored in such a way that they are not accessible or readable by unauthorised users of processes.

Requirement N. 4. *Pseudonym certificates SHOULD be stored securely, i.e., not accessible or readable by unauthorised users or processes.*

5.2.2.6 Repudiation of message transmission or reception

In case the Network Layer maintains a log-file for liability reasons, Requirement 5 applies to the Network Layer as well.

5.2.3 Facility Layer

5.2.3.1 Insecure firmware

Execution of insecure Facility Layer firmware gives unconditional privileges at the Facility Layer, which is a *Major* risk. However, as the unconditional privileges enable threats with *Critical* risks, the Facility Layer firmware must be protected from malicious modification. For example, the Facility Layer processes ITS message to maintain the Local Dynamic Map (LDM). However, insecure Facility Layer firmware can modify the LDM without the need of any ITS message. Also, it can generate incorrect user notifications, possibly leading to vehicle driver confusion or even accidents. Hence, the Facility Layer firmware must be protected against malicious modifications.

Requirement F. 1. *Facility Layer firmware MUST be protected from malicious modifications.*

Furthermore, because of Requirement N.2, the Facility Layer requires mechanisms to verify the integrity and authenticity of verification tickets.

Requirement F. 2. *The Facility Layer MUST be able to verify integrity and authenticity of ITS message's* verification attribute*.*

[2] The intervals for changing pseudonym certificates has not been standardised yet

5.2.3.2 Maliciously modifying stored data

Even if Requirements N.1 and F.1 are fulfilled, process data can still be modified. Modifications to stored data may directly or indirectly modify the LDM, which a *Major* threat. Hence, data at the Facility Layer must be secured against malicious modification of stored data

Requirement F. 3. *Data, stored at the Facility Layer, SHOULD be be secured against malicious modifications.*

5.2.3.3 Message injection, modification or deletion

ITS messages can be injected, modified or deleted at the Access Layer or Network Layer and may propagate to the Facility Layer, which is considered to be a *Critical* risk. Hence, not only messages must be authenticated (see Requirement 6), the Facility Layer must also perform plausibility checks on inbound ITS messages. Note, plausibility checks on inbound ITS messages is a mandatory functionality to be implemented in the Facility Layer [6].

Requirement F. 4. *The Facility Layer MUST perform plausibility checks on inbound traffic safety-related ITS messages, as mandated by ETSI [6]. For example, by reflecting the ITS message contents to the LDM contents.*

But, Requirement F. 4 does not protect against an insecure Application Layer that incorrectly notifies the Facility Layer on a traffic safety-critical event. Hence, the Facility Layer MUST perform plausibility checks on event notifications from Application Layer as well. For example, the plausibility checking mechanisms for inbound ITS messages could be extended to detect incorrect event-notifications.

Requirement F. 5. *The Facility Layer MUST perform plausibility checks on safety-relevant event notifications (initiated/generated by the Application Layer).*

5.2.3.4 Complete, selective or random DoS to processes

Requirements 2 and 3, which mitigate the risk of hardware tampering and damaging, apply to the Facility Layer as well.

The threat of complete or random deletion of ITS messages or Application Layer's event notifications probably can not be removed completely (see Observation 7). However, the Facility Layer must protect against message deletion. Hence, Requirement 6 applies to the Facility Layer as well.

5.2.3.5 Eavesdropping

The Facility Layer not only processes public ITS messages, it also contains and maintains confidential information. For example, vehicle IDs, vehicle owner information and session information (e.g., for IPv6 communications) will be stored at the Facility Layer. Eavesdropping on the Facility Layer's confidential information is considered to be a threat with *Major* risk. Hence, the Facility Layer must not leak confidential information.

Requirement F. 6. *The Facility Layer SHOULD store confidential information (such as the LDM) securely, i.e., not accessible or readable by unauthorised users or processes.*

Although the LDM only contains public processed ITS messages or event notifications, knowledge on an ITS Station's LDM may enable unique identification and localisation of that particular ITS Station. Hence, the LDM is considered to be an asset which not only requires integrity and authenticity assurance, but also confidentiality assurance.

Observation 9. *The LDM is considered to be data that requires integrity, authenticity and confidentiality assurance.*

5.2.3.6 Repudiation of message transmission or reception

In case the Facility Layer maintains a log-file for liability reasons, Requirement 5 applies to the Facility Layer as well.

5.2.4 Application Layer

5.2.4.1 Insecure firmware

Execution of insecure Application Layer firmware gives unconditional privileges at the Application Layer, which is a *Major* risk. However, as the unconditional privileges enable threats with *Critical* risks, the Application Layer firmware must be protected from malicious modification.

Requirement A. 1. *Application Layer firmware MUST be protected from malicious modifications.*

5.2.4.2 Execution of insecure applications

Execution of insecure applications, especially traffic safety-related ITS applications, could have a severe impact on the ITS Station's operation. Although ITS applications are only executed at the Application Layer, injections of insecure applications or modifications to genuine applications could occur at any protocol stack layer. Application updates could be modified or injected by any other protocol stack layer and is considered to be a *Critical* risk, when performed by a MITM attack. Hence, application updates must be secured against malicious changes.

Insecure applications can be *installed* or *injected* (see Definitions 3 and 4). Contrary to insecure firmware (which gives unconditional privileges at corresponding protocol stack layer), insecure applications could also have elevated privileges, but are bound to the privileges of the firmware it operates on.

Requirement A. 2. *Applications MUST be secured against malicious modifications.*

Furthermore, applications may share the resources of the firmware they run on. However, applications must not be able to affect other applications, which is especially important for traffic safety-related applications. Hence, applications must be protected from malicious modifications or eavesdropping by other applications.

Requirement A. 3. *Applications MUST not be able to monitor or influence other applications.*

Finally, some ITS applications have different privileges than others, e.g., ITS applications for an emergency vehicle. Clearly, non-emergency vehicles should not be able to run such applications. However, the privileges an ITS Station has been assigned are included in its digital certificates.

ITS messages, initiated/generated by applications with higher or different privileges will be detectable whenever they are not signed with the corresponding digital certificate. Hence, restricted applications could be protected from being stolen or copied.

Requirement A. 4. *Restricted applications, i.e., applications not intended for all ITS Stations, COULD be protected from execution on unauthorised ITS Station.*

5.2.4.3 Maliciously modifying stored data

Malicious modifications to stored application data may lead to, for example, incorrect event notifications. Even if Requirement A.3 is fulfilled, malicious changes to stored data may affect traffic safety-related applications or circumvent the need to install insecure applications. Whenever Requirement F.5 is fulfilled (i.e., plausibility checks on event notifications at Facility Layer are implemented), the *Critical* risk of maliciously modifying stored data at the Application Layer is mitigated. However, as the plausibility checks may not detect all malicious event notifications, it is beneficial to secure data that is stored at the Application Layer.

Requirement A. 5. *Data, stored at the Application Layer, COULD be secured against malicious modifications.*

5.2.4.4 Message injection, modification or deletion

Messages exchanged between the Facility Layer and Application Layer could be injected, modified or deleted by a MITM attack, which is considered to be a *Critical* risk. Hence, such messages must be authenticated and verifiable on their validity (see Definitions 9 and 10). Hence, the Application Layer must fulfill Requirement 6.

5.2.4.5 Complete, selective or random DoS to processes

Messages communicated to and from the Application Layer can be deleted selectively (see Observation 7). Hence, these messages must be communicated indistinguishably. Hence, the Application Layer must fulfill Requirement 7.

5.2.4.6 Eavesdropping

The Application Layer may process (third-party) service subscription information or other personal information that require confidentiality. The confidential information, stored at the Application Layer, should be protected against eavesdropping. This could be achieved by encrypting the confidential data (before storing it).

Requirement A. 6. *The Application Layer SHOULD store confidential information securely, i.e., not accessible or readable by unauthorised users or processes.*

5.2.4.7 Repudiation of message transmission or reception

In case the Application Layer maintains a log-file for liability reasons, Requirement 5 applies to the Application Layer as well.

5.2.5 Security Layer

5.2.5.1 Insecure firmware

Execution of insecure firmware at the Security Layer is a *Critical* risk as it would remove all confidentiality, integrity and authenticity guarantees that the Security Layer may provide. For example, unauthentic (inbound) ITS messages which contain an invalid signature could be processed anyway. Digital signatures of outbound ITS messages could be corrupted or not generated at all. Root certificates could be modified, possibly leading to the processing of invalid ITS messages (signed with an invalid digital certificate). Hence, the Security Layer firmware must be protected from malicious modifications. Moreover, as the Security Layer is the trust anchor for integrity and authenticity assurances for ITS messages, the Security Layer requires to be highly secure.

Requirement S. 1. *Security Layer firmware MUST be protected from malicious changes, in a highly secure fashion.*

5.2.5.2 Maliciously modifying stored data

Process data, corresponding to digital signature verification processes, is less security-critical as only public information is processed. However, the verification results must not be maliciously manipulable.

Observation 10. *The process of digital signature verification only concerns public information. However, the verification results must not be maliciously manipulable.*

However, processes that generate digital signatures of perform encryption/decryption process secret (cryptographic) information. Clearly, these processes must be protected against malicious manipulation or eavesdrop. Moreover, such information must in no case be maliciously modifiable as otherwise the integrity, authenticity and confidentiality assurances may become compromised.

Requirement S. 2. *Data, stored at the Security Layer, MUST be secured against malicious modifications, in a highly secure fashion.*

As tampering with the Security Layer may affect Security Layer firmware or data, the Security Layer must be resistant to tampering.

Requirement S. 3. *Operations at the Security Layer MUST be resistant to tampering.*

5.2.5.3 Message injection, modification or deletion

Injecting, deleting or modifying messages to or from the Security Layer is considered to be a *Critical* risk. As ITS message signatures are generated/verified or encrypted/decrypted at the Network Layer protocol stack level (see Assumptions 3 and 4), the messages exchanged between the Network Layer and Security Layer must be authentic and verifiable on their validity. Hence, the Security Layer must fulfill Requirement 6 as well.

5.2.5.4 Complete, selective or random DoS to processes

The Security Layer must be implemented in a tamper-resistant hardware box and be included in the built-in self-test, hence Requirements 2 and 3 apply to the Security Layer as well.

Furthermore, messages communicated to and from the Security Layer can be deleted selectively (see Observation 7). Hence, these messages must be communicated indistinguishably, thus the Security Layer must fulfill Requirement 7.

5.2.5.5 Eavesdropping

Clearly, the Security Layer contains secret (cryptographic) information which must not leak. Especially leakage of secret certificate keys (for signature generation) and encryption/decryption keys must remain confidential, as otherwise the integrity, authenticity or confidentiality assurance the signatures and encryption provides may be compromised. Hence, beside having secure firmware and being tamper-resistant (see Requirements S.1 and S.refreq:tamperresSEnt), the Security Layer must also be resistant to side-channel attacks. This prevents from leakage of confidential information through a covert channel (see [31] for more information on covert channels).

Requirement S. 4. *Processes, communication and storage at the Security Layer, which process secret (cryptographic) information, MUST be resistant to side-channel attacks to prevent from leakage of confidential information.*

5.2.5.6 Repudiation of message transmission or reception

In case the Security Layer maintains a log-file for liability reasons, Requirement 5 applies to the Security Layer as well.

5.2.6 Management Layer

As no standards on the Management Layer communication interfaces have been published (i.e., MI, MN, MF, MA and MS in Figure 4.1), the following is assumed for this risk analysis.

Assumption 6. *The Management Layer* does not *process complete ITS messages, but it* does *manage their "movements" within the ITS Station.*

5.2.6.1 Insecure firmware

Execution of insecure firmware at Management Layer gives unconditional privileges at the Management Layer. Typically the Management Layer does not store or process confidential information. However, it manages cross-layer communications and may configure any protocol stack layer. Also, the Management Layer decides when to change pseudonym certificates. Hence, the Management Layer firmware is integral to the correct operation of the ITS Station and requires protection against malicious modifications.

Requirement M. 1. *Management Layer firmware MUST be protected from malicious changes.*

5.2.6.2 Maliciously modifying stored data

Malicious modifications to data, stored in the Management Layer, may disturb the correct operation of the ITS Station. For example, swapping verification attributes of two verified ITS messages (see Definition 8) may result in discarding valid ITS messages or digesting invalid ones. This is comparable to the Network Layer that incorrectly forwards ITS messages and is considered to be a *Major* risk.

Requirement M. 2. *Data, stored at the Management Layer, SHOULD be secured against malicious modifications.*

5.2.6.3 Message injection, modification or deletion

By Assumption 6, the Management Layer manages communications (e.g., cross-layer), but will not process complete messages. Hence the Management Layer can not inject or modify ITS messages, but can have ITS messages discarded (e.g., labeling as “sent”) or processed (e.g., by labeling them as “valid”).

Furthermore, the Management Layer may reconfigure other protocol stack layers by in-ITS Station communication. Clearly, such management messages must be trustworthy, thus authenticated (see Definition 9. Also, the communication between the Management Layer and other protocol stack layers must be secured against replay attacks, hence the Management Layer must fulfill Requirement 6 as well.

5.2.6.4 Complete, selective or random DoS to processes

The Management Layer should be encapsulated in a tamper-resistant hardware box and must be included in the built-in self-test (see Requirements 2 and 3). Furthermore, messages communicated to and from the Management Layer can be deleted selectively (see Observation 7). Hence, these messages must be communicated indistinguishably, thus the Management Layer must fulfill Requirement 7.

5.2.6.5 Repudiation of message transmission or reception

In case the Management Layer maintains a log-file for liability reasons, Requirement 5 applies to the Management Layer as well.

5.3 Countermeasures

In this section we will suggest some countermeasures to the threats, corresponding to the requirements that have been argued in previous section.

5.3.1 Countermeasure List

The countermeasures in Table 5.1 could be used to mitigate the threats that have been indicated in previous section. When applied, corresponding requirements, that have been indicated in previous section, may be fulfilled. Note, this is a non-exhaustive list of countermeasures. More information on the impact of each countermeasure on the ITS Station architecture can be found in Appendix B.

cID	Countermeasure
1	Add a Message Authentication Code to each in-vehicular message (see Section B.1)
2	Add a unique identifier to each message (see Section B.2)
3	Authenticate ITS Station user (see Section B.3)
4	Authenticate ITS Station processes (see Section B.4)
5	Authenticate ITS Station hardware (see Section B.5)
6	Compartmentalise software executions (see Section B.6)
7	Comply to NIST firmware bootstrap and update guidelines (see Section B.7)
8	Digitally sign data (see Section B.8)
9	Encrypt data (see Section B.9)
10	Include application permissions in each ITS Station's certificate. (see Section B.10)
11	Managing access of applications, that require privileges outside their "compartment", by Management Layer (see Section B.11)
12	Perform plausibility checks on received messages (see Section B.12)
13	Perform a Built-In Self-Test (BIST) (see Section B.13) during boot time and during run time of the ITS Station
14	Use a secure element (see Section B.14)
15	Use hardware that is resistant to side-channel attacks (see Section B.15)
16	Use tamper-resistant hardware (see Section B.16)
17	Use tamper-evident hardware (see Section B.17)

Table 5.1: List of countermeasures

5.3.2 Countermeasure Mapping

In this section, we provide a list of countermeasures which fulfill one or more requirements as have been argued in previous section.

Requirements	cID	Notes
Requirement 1. *Integrity and authenticity of software updates must be assured.*	8	Provides assurance of integrity and authenticity up to the Network Layer level of the protocol stack, as the Network Layer hass access to signature verification mechanisms.
	1	Provides assurance of integrity and authenticity at higher protocol stack layer levels as well.
Requirement 2. *The ITS Station SHOULD be encapsulated in a tamper-resistant hardware box. Additionally, the hardware box could be tamper-evident.*	16	
	17	
Requirement 3. *The ITS Station MUST perform a check on hardware and software malfunctions or misconfiguration during boot time (i.e., during a so-called "cold boot") and notify the ITS Station user on the results. Additionally, if Requirement 2 is fulfilled (including tamper-evidence), this tamper evidence COULD be involved in the malfunction check procedure.*	1	Mitigates the risk of a MITM who maliciously modifies BIST results or user notifications.
	5	Enables detectability of unauthentic hardware components.
	13	Could be performed at other times than boot time as well.
	17	The evidence must be readable by the BIST mechanism.
Requirement 4. *Confidentiality of log-files SHOULD be assured, e.g., by encryption.*	9	Mitigates the risk of "simply" reading the memory unit that contains the log-files.
Requirement 5. *The log-files, maintained for liability reasons, must be protected against malicious modifications.*	11	Mitigates the risk protocol stack layers, maliciously modifying log-files.
	16	Mitigates the risk of corrupting the log-files.
Requirement 6. *Messages exchanged between protocol stack layers MUST be authenticated and be verifiable on their validity.*	1	For integrity and authenticity assurance.
	2	Preventing message replay
	12	Helps message validity validation.
Requirement 7. *Messages exchanged between protocol stack layers MUST be made indistinguishable, e.g., by encrypting the messages.*	9	Mitigates the risk of deleting messages on basis of their contents.

Table 5.2: Measures to fulfill requirements - All protocol stack layers

Requirements	cID	Notes
Requirement I.1. *Access Layer firmware COULD be protected against malicious modifications.*	1	Provides assurance of integrity and authenticity at higher protocol stack layer levels as well.
	7	Mitigates the risk of firmware injection (i.e., circumventing standard firmware installation procedures).
	8	Provides assurance of integrity and authenticity of firmware updates, up to the Network Layer level of the protocol stack, as the Network Layer has access to signature verification mechanisms.
Requirement I.2. *Data, stored at the Access Layer, COULD be protected against malicious modifications.*	3	Mitigates the risk of stealing an ITS Station (and abusing it)
	4	Mitigates the risk of stealing ITS Station processes (and abusing them)
	5	Mitigates the risk of stealing ITS Station (components) data (and abusing them)
	8	Mitigates the risk of malicious elevation of process privileges.
	9	Will not mitigate the risk of random data modifications.
	15	Mitigates the risk of, for example, data erase by physical environment influencing equipment.
	16	

Table 5.3: Measures to fulfill requirements - Access Layer (I)

Requirements	cID	Notes
Requirement N.1 . *Network Layer firmware must be protected from malicious modifications.*		Same as for Requirement I.1.
Requirement N.2 . *The verification results of a digital signature should be attached to corresponding ITS message in a* verification attribute*.*	1	The party who generates the MAC, as well as the party who verifies the MAC require a shared (secret) key.
	2	Mitigates the risk verification results replay
Requirement N.3. *Data, stored at the Network Layer, SHOULD be secured against malicious modifications.*		Same as for Requirement I.2.
Requirement N.4. *Pseudonym certificates SHOULD be stored securely, i.e., not accessible or readable by unauthorised users or processes.*		Same as for Requirements 4 and 5.

Table 5.4: Measures to fulfill requirements - Network Layer (N)

Requirements	cID	Notes
Requirement F.1. *Facility Layer firmware MUST be protected from malicious modifications.*		Same as for Requirement I.1.
Requirement F.2. *The Facility Layer MUST be able to verify integrity and authenticity of ITS message's* verification attribute.	1	The party who generates the MAC, as well as the party who verifies the MAC require a shared (secret) key.
	2	Mitigates the risk verification results replay
	12	Mitigates the risk of digesting spurious messages that contains a valid verification attribute.
Requirement F.3. *Data, stored at the Facility Layer, SHOULD be secured against malicious modifications.*		Same as for Requirement I.2.
Requirement F.4. *The Facility Layer MUST perform plausibility checks on inbound traffic safety-related ITS messages, as mandated by ETSI [6]. For example, by reflecting the ITS message contents to the LDM contents.*	12	Mitigates the risk of digesting spurious messages that contains a valid verification attribute.
Requirement F.5. *The Facility Layer MUST perform plausibility checks on safety-relevant event notifications (initiated/generated by the Application Layer).*	12	The plausibility check mechanism for Requirement F.4 could be extended to verify outbound messages as well.
Requirement F.6. *The Facility Layer SHOULD store confidential information (such as the LDM) securely, i.e., not accessible or readable by unauthorised users or processes.*		Same as for Requirements5

Table 5.5: Measures to fulfill requirements - Facility Layer (F)

Requirements	cID	Notes
Requirement A.1. *Application Layer firmware MUST be protected from malicious modifications.*		Same as for Requirement I.1.
Requirement A.2. *Applications must be secured against malicious modifications.*	1	Mitigates the risk of malicious modifications to application update between higher protocol stack layers (i.e., from the Network Layer and higher).
	7	Mitigates the risk of firmware injection (i.e., circumventing standard firmware installation procedures).
	11	Mitigates the risk of malicious application update modifications up to, up to the Network Layer level of the protocol stack, as the Network Layer has access to signature verification mechanisms.
Requirement A.3. *Applications must not be able to monitor or influence other applications.*	6	Mitigates the risks of applications influencing or monitoring other application's processes and data.
	8	Mitigates the risk of applications that (maliciously) elevated their privileges to modify other application's data.
Requirement A.4. *Restricted applications, i.e., applications not intended for all ITS Stations, COULD be protected from execution on unauthorised ITS Station.*	10	The Network Layer can verify the privileges of applications according to the ITS Station's certificate.
Requirement A.5. *Data, stored at the Application Layer, COULD be secured against malicious modifications.*		Same as for Requirement I.2.
Requirement A.6. *The Application Layer SHOULD store confidential information securely, i.e., not accessible or readable by unauthorised users or processes.*		Same as for Requirement I.5

Table 5.6: Measures to fulfill requirements - Application Layer (A)

Requirements	cID	Notes
Requirement S.1. *Security Layer firmware MUST be protected from malicious changes, in a highly secure fashion.*		Same as for Requirement I.1.
	14	Includes countermeasures 6, 7, 16, 17 and 18.
Requirement S.2. *Data, stored at the Security Layer, MUST be secured against malicious modifications, i.e., not accessible or readable by unauthorised users or processes.*		Same as for Requirement I.2.
	14	Includes countermeasures 6, 7, 16, 17 and 18.
Requirement S.3. *Operations at the Security Layer MUST be resistant to tampering.*	14	Includes countermeasures 6, 7, 16, 17 and 18.
Requirement S.4. *Processes, communication and storage at the Security Layer, which involve secret (cryptographic) information, MUST be resistant to side-channel attacks to prevent from leakage of confidential information.*	14	Includes countermeasures 6, 7, 16, 17 and 18.

Table 5.7: Measures to fulfill requirements - Security Layer (S)

Requirements	cID	Notes
Requirement M.1. *Management Layer firmware MUST be protected from malicious changes.*		Same as for Requirement I.1
Requirement M.2. *Data, stored at the Management Layer, SHOULD be secured against malicious modifications.*		Same as for Requirement I.2.

Table 5.8: Measures to fulfill requirements - Management Layer (M)

CHAPTER 6

Security Analysis - Partitioned

In this chapter, we analyse two potential hardware partitionings. That is, the protocol stack layers from the reference architecture for the ITS Station (see Figure 3.1) are mapped to certain hardware components. The hardware partitioning may change the requirements as have been discussed in previous chapter, or introduce new requirements. When we refer to assumptions, observations or requirements—without the extension *PI.* or *PII.*—we refer to those as have been argued in Section 4.5. Furthermore, both partitionings may have advantages and disadvantages with respect to the requirements from Section 4.5 and will be indicated by *PRO Px.* and *CON Px.*, where x is either *I* or *II*.

6.1 Partitioning I

For partitioning I (PI), the following hardware components are assumed to implement certain (parts of) protocol stack layers. The hardware components are assumed to be interconnected as depicted in Figure 6.1.

- Radio transceiver. The radio transceiver is an 802.11p-compatible radio modem and tuner. It implements the Access Layer. Beside radio transmission error detection and correction mechanisms, the radio transceiver does not comprise any information security-related features.
- CPU. A single processor implements the Network Layer, Facility Layer and Application Layer (and parts of the Management Layer and Security Layer).
- Verification Accelerator. The Verification Accelerator (VA) is a hardware accelerator for digital signature verifications, hence partially implements the Security Layer. The VA is assumed to be incapable of storing complete certificate chains, hence it must be able to obtain the (root) certificates on-the-fly.
- Secure Element. The Secure Element (SE) is assumed to provide highly secure, tamper-resistant storage and processing, i.e., data and processes in the SE can not be monitored or influenced maliciously. Root certificates and confidential information will be stored in the SE,

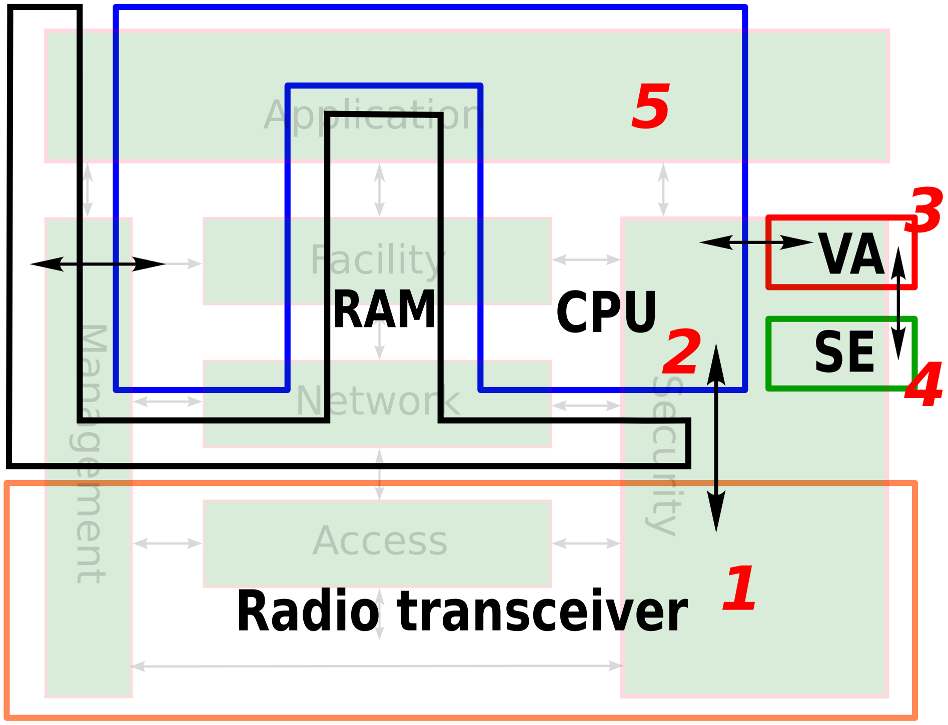


Figure 6.1: Hardware Partitioning I

e.g., for the VA to verify a certificate chain. An example of such SE is the (Common Criteria AEL6+ certified) *SmartMX2* secure smart card product family by NXP Semiconductors [32].

Typically, the SE uses a physical interface which is not supported by most CPUs/MCUs. As the VA *is* capable of interfacing with the SE, as well as with CPUs, the SE communicates with the CPU via the VA. Furthermore, whenever the VA and SE combination is considered as a single entity, integration of the communication at the CPU is eased.

- External memory. Most likely, the higher protocol stack layers will require a storage size that is larger than the internal memory of CPU, e.g., for storing pseudonym certificates and maintaining the LDM. Hence, an external memory unit will be connected to the CPU.

With PI, a typical path for an inbound ITS message is: 1) AD conversion by radio transceiver, 2) process by CPU (i.e., Network Layer), 3) digital signature check by VA, 4) if necessary, request (root) certificates from SE, 5) process by CPU (i.e., Network Layer, Facility Layer and Application Layer).

6.1.1 Risk Evaluation

- Radio transceiver. The Access Layer is completely implemented in the radio transceiver, hence, all Access Layer-related requirements hold for the radio transceiver. Furthermore, the radio transceiver will implement (parts of) the Management Layer and Security Layer as well.

According to ETSI, when partitioning the reference architecture on several components—as is the case here, clearly—every unit contains an instance of the Management Layer [13, p.11]. However, most likely, every unit will contain a *partial* instance of the Management Layer, as the higher protocol stack layers will require more or other management services than, for example, the Access Layer. Nevertheless, as no details on the exact implementation standards of the Management Layer are published yet, all Management Layer-related requirements are considered to hold for every unit.

Observation PI. 1. *All requirements of the Access Layer and Management Layer apply to the radio transceiver.*

Observation PI. 2. *Each unit that implements (parts of) the reference architecture for an ITS Station contains a (partial) instance of the Management Layer.*

Finally, although the radio transceiver will implement (parts of) the Security Layer, it will not perform security-critical operations, e.g., authenticity verification or encryption/decryption. Basically, it converts analog and digital signals (see Observation1). [1] Hence the requirements for the Security Layer are less stringent for the radio transceiver.

Observation PI. 3. *Contrary to Observation PI.1, the requirements for the Security Layer do not (necessarily) apply to the radio transceiver.*

- CPU. The Network Layer, Facility Layer and Application Layer are completely implemented in the CPU, hence, all corresponding requirements hold for the CPU as well. However, some redundancy in requirements may be present. For example, these three protocol stack layers must all have measures to protect against malicious modifications of their firmware (see Requirements N.1, F.1 and A.1). But, as they are implemented on the same CPU, a single firmware will operate all three layers. Hence, the firmware protection only needs to be applied once for the CPU.

PRO PI. 1. *As the Network Layer, Facility Layer and Application Layer are combined into one firmware at the CPU, only one firmware protection measure is required for the three protocol stack layers. Consequently, whenever the requirements with respect to the firmware of one of these protocol stack layers are fulfilled, those of the other two are as well.*

However, as a direct consequence of PRO PI.1, execution of insecure firmware may directly affect all the three protocol stack layers.

CON PI. 1. *As the Network Layer, Facility Layer and Application Layer are combined into one firmware at the CPU, execution of insecure firmware at the CPU may directly affect the Network Layer, Facility Layer and Application Layer.*

Also, if insecure applications are executed, the Application Layer firmware may be affected,

[1] The radio transciever will also (try to) detect and correct erroneous radio transmissions, with the help of a checksum value which is attached to each ITS message. However, an attacker who maliciously modifies message and corresponding checksum value before it is received, will go undetected, thus uncorrected. Hence, these error detection and correction mechanisms are not considered in this analysis.

hence, the Network Layer and Facility Layer as well.

CON PI. 2. *If the Network Layer, Facility Layer and Application Layer are combined into one firmware at the CPU, execution of insecure applications may directly affect the Network Layer, Facility Layer and the Application Layer.*

To reduce the risk of protocol stack layers—implemented on the same CPU—affecting the other protocol stack layers, they should be executed compartmentalised. That is, every protocol stack layer is executed in an isolated environment which is executed "on top" of the "kernel" of the CPU.

Requirement PI. 1. *To reduce the risk of protocol stack layers affecting each other, they SHOULD be executed compartmentalised.*

In order to distinguish between protocol stack layer firmware and CPU firmware (or, kernel), we use the following definition.

Definition PI. 1. *Whenever multiple protocol stack layers are implemented on the same CPU and the protocol stack layers are executed in a compartmentalised environment, the firmware of the protocol stack layers is referred to as the* <protocol stack layer> firmware *(e.g., Network Layer firmware). Protocol stack layer firmware "runs on top" of the firmware of the CPU, which is referred to as the* CPU firmware.

When a protocol stack layer "breaks out" of its compartment, it may gain permissions on CPU firmware "level", thus the ability to affect other compartmentalised protocol stack layers. Hence, the choice of the compartmentalisation mechanism is dependant for the actual security is provides. The following non-exhaustive list indicates the compartmentalisation mechanisms that are considered in this analysis. The advantages and disadvantages are discussed briefly for each mechanism.

- Hardware separation. Separating protocol stack layer(s) by implementing them on different physical components, e.g., CPUs.
  - *Advantage.* Protocol stack layers can only influence other protocol stack layers—implemented on the other physical unit—via the communication interface between the components.
  - *Disadvantage.* Communication between the physical components may require integrity and authenticity assurance. This may imply that the physical components have to establish and securely store a shared key, e.g., for MAC generation/verification.
- Trusted Execution Environment. CPUs that offer Trusted Execution Environments [2] (TEE), can execute software either in a "normal world" or in a "secure world" (see Figure 6.2. The "secure world" provides secure execution of software and secure storage, whereas the "normal world" typically executes (third-party) applications. For example, ARM TrustZone® supports TEE.
  - *Advantage.* Software, executed in the "normal world ", can only interact with software that is executed in the "secure world" via strict and highly secure TEE kernel.

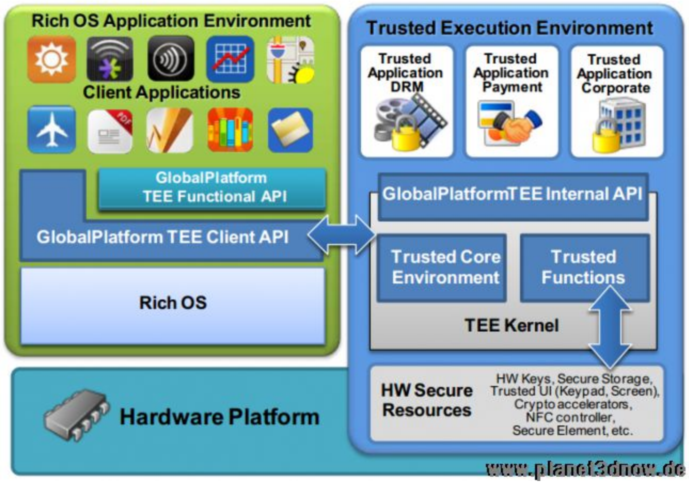

Rich OS Application Environment
Client Applications
GlobalPlatform TEE Functional API
GlobalPlatform TEE Client API
Rich OS
Hardware Platform
Trusted Execution Environment
Trusted Application DRM
Trusted Application Payment
Trusted Application Corporate
GlobalPlatformTEE Internal API
Trusted Core Environment
Trusted Functions
TEE Kernel
HW Secure Resources
HW Keys, Secure Storage, Trusted UI (Keypad, Screen), Crypto accelerators, NFC controller, Secure Element, etc.
www.planet3dnow.de

Figure 6.2: Trusted Execution Environment (from [35])

Additionally, as communication between protocol stack layers—executed in different worlds—occurs internal to the CPU and are strictly controlled by the CPU firmware, no additional security measures for this communication is required.

∗ *Disadvantage.* Due to the complexity of CPUs with TEE, development of software for this CPU is more difficult, hence more subject to faulty (insecure) implementation. Furthermore, TEE creates "only" two compartments: the "secure world" and the "normal world". Finally, TEE-enabled CPUs are, in general, more expensive than standard non-TEE-enabled CPUs.

– Virtualisation. Virtualised software is executed in an isolated user-space, which is executed "on top" of the kernel of the operating system (see Figure 6.3. Virtualised software can only communicate via the kernel.

∗ *Advantage.* Does not require a (probably more expensive) TEE-enabled CPU. Furthermore, in general, an unlimited number of virtualised compartments can be created. [2] Hence, virtual compartmentalisation allow more flexible software design than TEE-enabled CPUs, for example.

∗ *Disadvantage.* Virtual compartmentalisation is less secure than TEE and hardware separation. Hence, the risk of virtualised software "breaking out" of its compartment is higher than with the other two methods.

Now, because of Requirement PI.1, all three protocol stack layers at the CPU should be compartmentalised. Hardware separation is not an option as all three protocol stack layers are implemented on one CPU. Hence, the protocol stack layers can be compartmentalised virtually or by using a TEE.

Most likely, the Network Layer, Facility Layer and Management Layer are developed by trusted manufacturers, and as these protocol stack layers are implemented on the same CPU

[2] Clearly, the number of virtualised compartments that can be created is limited to the available resources of the CPU.

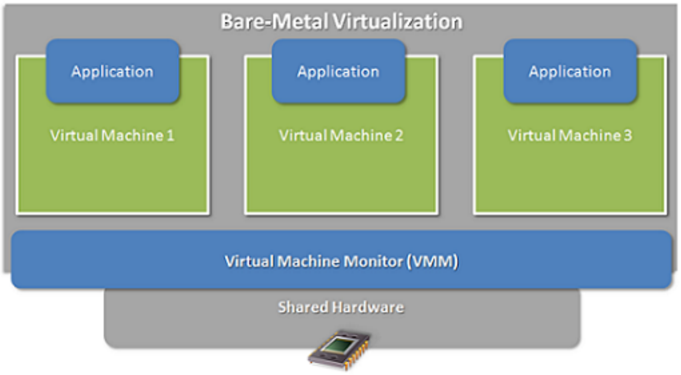


Figure 6.3: Software Virtualisation (from [26]

the risk of MITM attacks on their intercommunication is not critical. Hence, both protocol stack layers not necessarily need to be executed in separate compartments. The Application Layer, on the other hand, may support (untrusted) third-party applications, which may be insecure. Hence, the Application Layer must be separated from the Network Layer, Facility Layer and Management Layer. The Network Layer, Facility Layer and Management Layer firmware must not be maliciously modifiable (see Requirements N.1, F. 1 and M. 1). And, because a "break out" of Application Layer from its compartment may compromise the other protocol stack layers, compartmentalisation by TEE is recommended. Hence, the Network Layer, Facility Layer and Management Layer should be executed in the "secure world" and the Application Layer in the "normal world".

Requirement PI. 2. *The CPU SHOULD compartmentalise protocol stack layers by using Trusted Execution Environments. The Application Layer should always be separated from the other protocol stack layer (compartments). Also, the lower protocol stack layers must be implemented in the most secure compartment(s).*

Then, some requirements may overrule others. For example, data, stored at the Network Layer *should* be protected from malicious modification, whereas data, stored at the Facility Layer *must* have such protection (see Requirements N. 3 and F.3). In such case, the most stringent security requirement must be fulfilled. In this case, the memory data *must* be protected from malicious modifications, as the Facility Layer requires so.

Observation PI. 4. *If two requirements contradict, one requirement should be adapted to the requirement that demands higher security.*

Furthermore, communication between the Network Layer, Facility Layer and Application Layer occurs internal to the CPU or via the external memory. Hence, the likelihood of MITM attacks between the Network Layer, Facility Layer and Application Layer is decreased (for communication internal to the CPU).

PRO PI. 2. *The risk of MITM attacks on communication between the Network Layer, Facility*

Layer and Application Layer decreased because the communication is internal to the CPU. Hence, such communication does not necessarily have to be authenticated or (probabilistically) encrypted (see Requirements 6 and 7)

However, if protocol stack layers communicate via the external memory unit, the communication *does* require integrity and authenticity assurance (see Requirements 6 and 7). Furthermore, according to Requirement N.4, pseudonym certificates should be stored in a manner such that they are not accessible or readable by unauthorised users or processes. But, as these certificates are likely to be stored in the *shared* external memory unit, they are required to be stored inaccessible and unreadable for unauthorised users and processes. The Management Layer should manage the authentication and authorisation of access to the external memory unit. One way to achieve this is to have the CPU's protocol stack layers and the external memory unit communicate through the Management Layer always.

Requirement PI. 3. *Access of multiple protocol stack layers—implemented on the same CPU—to a shared (external) memory unit MUST go via the Management Layer always. This reduces the risk of one protocol stack layer maliciously modifying other protocol stack layers' data (stored in shared memory).*

Clearly, the Management Layer is the "trusted party", in this solution. But, if an attacker is capable of maliciously modifying the Management Layer firmware, it probably is capable of modifying other protocol stack layer firmware as well. Furthermore, the CPU firmware and protocol stack layer firmware are secured when NIST's firmware guidelines [20] are met (see Countermeasure B.7).

Observation PI. 5. *Whenever complying with NIST's firmware guidelines [20] (see Countermeasure B.7, the CPU firmware and protocol stack layer firmware are protected from malicious modification.*

- Verification Accelerator. The Security Layer must have secure processing and storage, and must be resistant to tampering and side-channel attacks (see Requirements S.1, S.2, S.3 and S.4). However, the Verification Accelerator only concerns public information as it only verifies digital certificates (see Observation 10). Hence, the requirements corresponding to the Security Layer, do not necessarily apply to the VA.

  Furthermore, digital signature verification results must be attached to the ITS messages as a *verification attribute* (see Requirement N.2), such that the verification results can not be modified maliciously by an attack at the Network Layer or by a MITM attack on the communication interfaces. Although this measure is necessary to prevent from the latter attack, the former attack is not really prevented, as will be discussed next. For the rest of the analysis we assume that a MAC will assure the integrity and authenticity of the verification results .

Assumption 7. *A Message Authentication Code (MAC) will assure the integrity and authenticity of the digital signature verification results.*

Whenever protocol stack layers *are not* executed compartmentalised and the Network Layer firmware is insecure, then the Facility Layer and Application Layer firmware is affected also, as they are operated by the same firmware (see CON PI.1. If so, insecure firmware could still maliciously modify verification results, even if the verification results' integrity and authenticity is assured by a MAC. Hence, the MAC will only assure integrity and authenticity of the verification results during communication, when protocol stack layers are not executed compartmentalised.

However, when protocol stack layers *are* executed compartmentalised, insecure Network Layer firmware not necessarily affects the firmware of the Facility Layer or Application Layer. Hence, the MAC does protect from malicious modifications to the verification results by the Network Layer as well.

But, if the Network Layer is insecure (or not trusted), the verification results must be communicated from the VA to the Facility Layer in an end-to-end secure manner, as otherwise the Network Layer could still modify the verification results. This means, the VA and Facility Layer must establish a shared secret key, say `kVF`, unknown to the Network Layer. However, for assuring integrity and authenticity of communication between the VA and CPU, the VA and Network Layer require to establish a shared secret key, say `kVN`, as well. Consequently, each verification result will require two MACs. One for assuring intergrity and authenticity of the results while communicated between the VA and Network Layer, and one for providing such assurance while processes by the Network Layer [3]:

$$MAC_{kVN}(MAC_{kVF}(\textit{verification results}))$$

However, attaching two MACs per verification result may be an unacceptable overhead on the computational resources of the VA and on the message size.

On the other hand, protocol stack layers are compartmentalised by using a TEE (see Requirement 2) and an attacker is capable of maliciously modifying Network Layer firmware, it probably is also capable of maliciously modifying other firmware. Hence, in this partitioning, the verification results not necessarily have to be assured in an end-to-end manner (i.e., from the VA to the Facility Layer).

Observation PI. 6. *Whenever the Network Layer and Facility Layer are implemented on the same CPU and compartmentalised together (or in equally secure but different compartments), no MAC is required to assure integrity and authenticity of the verification results after being communicated from the VA to the Network Layer.*

However, whenever the Network Layer and Facility Layer are implemented in different CPUs, at least two MACs will be required.

Observation PI. 7. *Whenever the Network Layer and Facility Layer are implemented on different CPUs, the VA—which will be connected to the Network Layer's CPU—will require at least two secret (shared) keys. One for assuring integrity and authenticity of communication between the VA and Network Layer's CPU, and one for assuring integrity and authenticity of the verification results that will be checked at the Facility Layer. The latter is required whenever the Network Layer is not trusted.*

[3] $MAC_x(y)$ means: the MAC of y, generated with key x.

Furthermore, the verification process itself must be protected as well. Hence, the VA firmware must be protected from malicious modifications (see Requirement S.1).

Also, whenever the VA obtains (root) certificates for verifying a certificate chain, integrity and authenticity of the certificates must be assured. Hence, Requirements 6 and 7 apply to the communication between the VA and SE as well. Consequently, the VA requires secure memory (see Requirement S.2) for storing the secret (shared) key that will be used to assure the integrity and authenticity (e.g., for MAC generation). Nevertheless, fulfillment of this requirement could be implemented in a less stringent fashion (compared with the SE) as the VA does not process long-term secret information such as, for example, long-term certificate keys.

Observation PI. 8. *Although the Verification Accelerator (VA) in general only concerns public information, it does require secure processing and secret information to ensure integrity and authenticity of the verification results. Hence, the requirements, corresponding to the Security Layer, do apply to the VA but may be fulfilled in a less stringent fashion (compared with the SE).*

Finally, key-establishment is a crucial part in securing the communication between two components (e.g., the VA and CPU). First of all, all three methods require secure memory for storing pre-shared information. Hence, this is not a discriminating factor. Also, by all methods a fresh session key can be generated.

Requirement PI. 4. *All ITS Station components MUST assure integrity and authenticity of their communications require secure memory for storing pre-shared information.*

The following non-exhaustive list indicates the key-establishment methods that are considered in this analysis.

i) Pre-sharing a (shared) key, e.g., during manufacturing of the ITS Station component(s).
   - *Advantages.* No key-establishment required when the ITS Station is in an untrusted environment, unless a fresh session key will be established, for example, at boot time of the ITS Station.
   - *Disadvantages.* Key-establishment may be difficult if ITS Station components are manufactured by different manufacturers.

ii) Establishing a shared key—in a secure environment—when the complete ITS Station is assembled. For example, by storing the shared key in one-time-programmable memory.
   - *Advantages.* Same advantage as method *i*. And, moreover, the disadvantage of *i* does not apply.
   - *Disadvantages.* The key will be stored long-term. Once compromised, integrity and authenticity can not be assured anymore.

iii) Establishing a shared key on-the-fly by digital certificates, i.e., using a key-establishment protocol.
   - *Advantages.* Contrary to methods *i*, *ii*, components such as third-party peripherals—unknown to the ITS Station during manufacturing—can be authenticated and pos-

sibly enrich ITS Station features. If the certificates are not pre-shared, integrity and authenticity of the certificates must be protected via a public-key infrastructure.

- *Disadvantages.* The communicating components must either have stored each others digital certificate such that integrity and authenticity of the digital certificates is assured. Or, a public-key infrastructure must be implemented, with which the components can verify integrity of the certificates on-the-fly.

Now, as multiple manufactures will manufacture (different) ITS Station components, method *i* seems to be infeasible. However, for components that have the same manufacturer and need to communicate, there is no difference in method *i* and *ii*. Nevertheless, most likely this will not be the case for all components. Hence, method *i* is not recommended. Then, the only advantage of method of method *iii* over *ii* is that third-party peripherals can be authenticated on-the-fly. However, the first ITS Stations on the market are probably not compatible with third-party peripherals. Moreover, as stealing ITS Station components is estimated to be a *Major* or *Critical* risk, establishing a shared key only once—in a secure environment—assures that the components will not be usable in other ITS Stations. Hence, method *ii* is recommended.

Requirement PI. 5. *ITS Station components SHOULD establish a shared key—only once—in a secure environment.*

Note, key diversification must be ensured to mitigate the impact of a compromise of a key. However, the risks of MITM attacks between any of the protocol stack layers are equal. Hence, key diversification must be implemented at least on ITS Station level, i.e., different shared key for each ITS Station. Key diversification for specific ITS Station-specific components could be implemented as well. [4].

Requirement PI. 6. *Each ITS Station MUST (at least) have a different (e.g., random) key for assuring integrity and authenticity of communication internal to the ITS Station. Every two ITS Station components SHOULD establish a shared key, different from the shared key between other components.*

- <u>Secure Element</u>. The task of the Secure Element (SE) is twofold; *i*) highly secure storage of (secret) information and, *ii*) execution of cryptographic processes that use the (secret) information (e.g., digital signature generation and encryption/decryption).

  Contrary to the VA, the Secure Element (SE) *does* contain secret information [5], hence it does require resistance to tampering and side-channel attacks (see Requirement S.3 and S.4). Furthermore, processes and storage at the Security Layer must be highly secured, as secret information is involved (see Requirements S.2 and S.1), contrary to the VA (see Observation PI.8).

  Furthermore, when the Network Layer request the SE to digitally sign an ITS message, the unencrypted and encrypted ITS messages will pass the VA always (see Figure 6.1. However, the

[4] That is, every two communicating ITS Station components establish a shared key, different from the shared key between other components

[5] The VA *should* contain secret information as well (for assuring integrity and authenticity of the verification results), but this is an extrinsic requirement for the digital signature verification process, as signature verification only concerns public information.

VA has less stringent requirements regarding resistance to tampering and side-channel attacks. Hence, if the VA executes insecure firmware or is influenced otherwise, it could maliciously modify or delete the unencrypted and encrypted ITS messages. Hence, the communication between the Network Layer and SE should be authenticated and (probabilistically) encrypted (see Requirements 6 and 7). Clearly, the Network Layer and the signature verification mechanism are not implemented on the same unit, hence the communication between the CPU and SE should be secured in an end-to-end manner.

Requirement PI. 7. *Communication between the Network Layer and the Secure Element (SE) MUST be authenticated and (probabilistically) encrypted (see Requirements 6 and 7) to prevent the VA from maliciously modifying or selectively deleting messages that are exchanged between the Network Layer and SE. Note, this MUST be implemented in an* end-to-end *fashion, as otherwise the VA could still modify or selectively delete the messages.*

- Memory The external memory unit, or external memory, is shared between multiple protocol stack layers. The Management Layer should manage authentication and authorisation for access to this memory unit (see Requirement PI.3). Clearly, the Management Layer is the "trusted party" in this. However, its integrity and authenticity is verified whenever the CPU firmware is verified (see Observation PI.5. Hence, if an attacker is capable of maliciously modifying the Management Layer firmware, it will probably also be capable of modifying other protocol layer firmware.

  Furthermore, the protection requirements for data stored at the Network Layer, Facility Layer and Application Layer, contradict. However, because of Observation PI.4, the requirement with highest security demands must be fulfilled. Hence, the data storage of all three protocol stack layers should be protected from malicious modifications.

6.2 Partitioning II

For partitioning II (PII), the following hardware components are assumed to implement certain (parts of) protocol stack layers. The hardware components are assumed to be interconnected as depicted in Figure 6.4.

- Radio transceiver. See PI (Section 6.1) for the description.
- CPU-north. A first physical processor unit, referred to as "CPU-north", implements the Application Layer (and parts of the Management Layer and Security Layer).
- CPU-south. A second physical processor unit, referred to as "CPU-south", implements the Network Layer and Facility Layer (and parts of the Management Layer and Security Layer).
- Verification Accelerator. See PI (Section 6.1) for the description.
- Secure Element. See PI (Section 6.1) for the description.
- Memory-north. Most likely, the Application Layer requires a storage size that is larger than the internal memory of CPU-north, e.g., for storing application data. Hence, an external memory unit, referred to as "memory-north", will be connected to CPU-north.

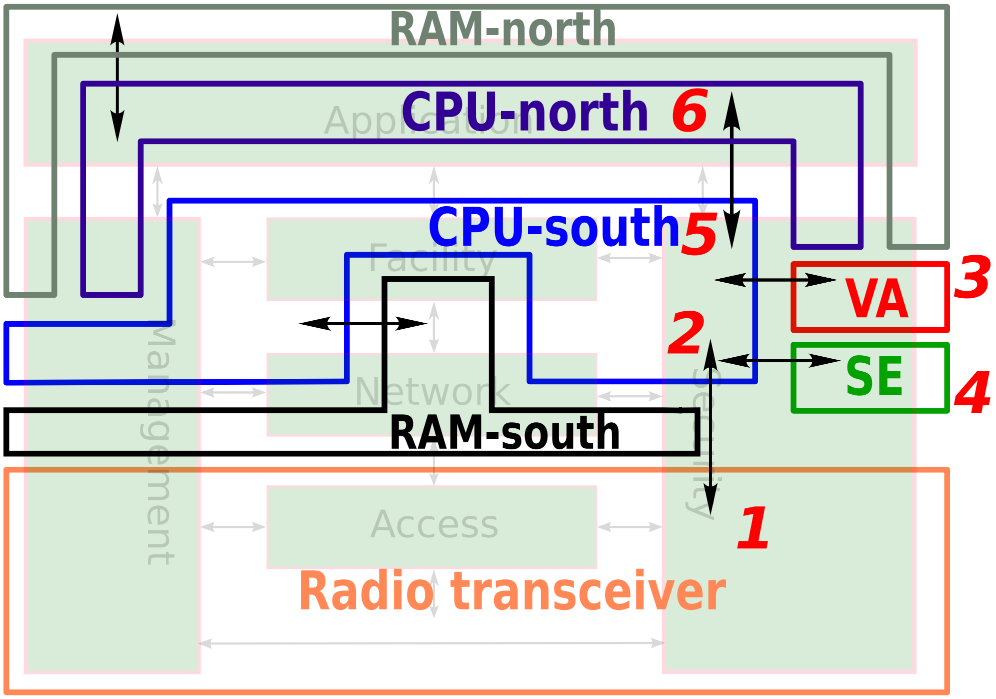


Figure 6.4: Hardware Partitioning II

- Memory-south. As with partitioning I, the Network Layer and Facility Layer will most likely require a storage size that is larger than the internal memory of CPU-south, e.g., for storing pseudonym certificates and maintaining the LDM. Hence, an external memory unit, referred to as "memory-south", will be connected to CPU-south.

With PII, a typical path for an inbound ITS message is: 1) AD conversion by radio transceiver, 2) process by CPU-south (i.e., Network Layer), 3) digital signature check by VA, 4) if necessary, request (root) certificates from SE *via CPU-south*, 5) process by CPU-south (i.e., Network Layer and Facility Layer) and 6) process by CPU-north (i.e., CPU-north).

6.2.1 Risk Evaluation

The main differences between PI and PII are the partitioning of the Application Layer and the interconnections of the CPU, VA and SE.

- Radio transceiver. Partitioning I and II do not differ on their implementation of the radio transceiver, hence see previous section for the security analysis on the radio transceiver.

- CPU-north. The Application Layer is separated from the lower protocol stack layers by hardware (compartmentalisation). Hence, contrary to PI, insecure Application Layer firmware or applications do not (necessarily) affect the lower protocol stack layers.

PRO PII. 1. *Execution of insecure applications not necessarily affects the Network Layer and Facility Layer.*

However, contrary to PI (see PRO PI. 1), PII requires security measures twice, to prevent from malicious modification CPU firmware, i.e., for CPU-north and for CPU-south.

CON PII. 1. *Securing both CPUs from executing insecure firmware requires corresponding countermeasures to be implemented twice, i.e., for CPU-north and for CPU-south.*

Furthermore, contrary to partitioning I, communication between the Facility Layer and Application Layer does not occur internal to one CPU. Hence, the likelihood of MITM attacks between the Facility Layer and Application Layer is higher, compared to partitioning I. Therefore, such communication must be authenticated and (probabilistically) encrypted (see Requirements 6 and 7). To establish an authenticated and (probabilistically) encrypted communication channel between CPU-south and -north, the same reasoning for the key-establishment as with PI applies (see Requirement PI.7, PI.5 and PI.6).

- <u>CPU-south</u>. As the Application Layer is separated from the lower protocols stack layers by hardware (compartmentalisation) and because of PRO PII.1, the lower protocol stack layers not necessarily require TEE compartmentalisation. Although the Network Layer, Facility Layer and Management Layer require protection from malicious modifications to their firmware (see Requirements N1, F.1 and M.1, most likely they will be developed by trusted manufacturers. Hence, the compartmentalisation of the Network Layer, Facility Layer and Application Layer by virtualisation is sufficient.

Requirement PII. 1. *The Network Layer, Facility Layer and Management Layer SHOULD (at least) be compartmentalised by virtualisation.*

- <u>Verification Accelerator</u>. The security analysis of the Secure Element (SE) in PII is slightly different from that of partitioning I, because the VA is not the "man in the middle" for communication between the Network Layer and the SE. The risk of the VA maliciously deleting messages, exchanged between the Network Layer and SE is reduced, but not removed, as MITM attacks on the communication interface between CPU-south and the SE are still possible. Clearly, whenever the communication between CPU-south and the SE is authenticated and (probabilistically) encrypted, an end-to-end authenticated channel is established. Hence, Requirement PI.7 will be fulfilled also.

PRO PII. 2. *The risk of having the Verification Accelerator (VA) inserting, modifying, deleting or eavesdropping communication between the CPU and SE is removed, as the VA is not an intermediate "node" on that communication interface.*

However, if the VA requires (root) certificates for verifying a certificate chain, the CPU will be the "man in the middle". Nevertheless, if the CPU is compromised, it could unconditionally generate ITS messages and modify verification results. Hence, no end-to-end secured communication channel is necessary between the VA and SE.

Requirement PII. 2. *A compromised CPU will be able to generate malicious ITS messages* *and modify verification results. Hence, an end-to-end secured communication channel between the VA and SE is not a necessity, but COULD be implemented.*

Also, as the CPU is the "man in the middle" whenever the VA requires to obtain (root) certificates from the SE, signature verification latency may be increased (compared with PI).

CON PII. 2. *Signature verification may be slower (compared with PI) as the (root) certificate requests from the VA will go through the CPU always.*

Furthermore, the reasoning for key-establishment and key diversification applies to PII as well (see Requirements PI.5 and PI.6).

- Secure Element. Contrary to PI, the VA can not maliciously modify or delete messages exchanged between the CPU and SE (see PRO PII.2). This reduces the risk of MITM attacks, but also improves the performance of signature generation and encryption/decryption.
- Memory-north. The security analysis for external memory-north is analogous to the analysis of the external memory in partitioning I.
- Memory-south. The security analysis for external memory-south is analogous to the analysis of the external memory in partitioning I.

6.3 Security Comparison

In this section, we summarise the advantages of one partitioning over the other.

6.3.1 PI over PII

Firstly, with PI the risk of MITM attacks between the Facility Layer and Application Layer is reduced, as both protocol stack layers are implemented on one CPU. This could also reduce the latency that will be introduced by the communication between the Facility Layer and Application Layer, as no additional integrity and authenticity assurances are required on that communication.

Secondly, with PI, the VA has a direct connection to the SE, which decreases the risk of having a MITM inserting, modifying or deleting messages on their communication. Furthermore, as the VA can request (root) certificates from the SE directly (i.e., without an intervening CPU) the performance of signature verifications will be better than with PII.

6.3.2 PII over PI

Firstly, execution of insecure firmware or applications not necessarily affect the Network Layer or Facility Layer, as both protocol stack layers are implemented on different CPUs. Although compartmentalisation of firmware and application executions in PI will also decrease this risk, both CPUs in PII *physically* compartmentalise the protocol stack layers by hardware. As software can not "break out" of such "physical compartment", the risk of protocol stack layers affecting other protocol stack layers is lower compared to PI. Hence, with PII, CPU-north and -south not necessarily require a TEE.

Secondly, with PII, the risk of a MITM attack on communication between CPU-south and the SE is lower, as the VA does not intervene. This, in particular, is beneficial whenever the Network Layer would like to have a message signed, for example.

Thirdly, no end-to-end secured communication is necessary between the SE and VA.

6.3.3 Partitioning Recommendations

As discussed in previous sections, both hardware partitionings introduce slight, but important, differences on the requirements as have been discussed in Section 4.5.

If multiple protocol stack layers are implemented on one CPU, they must be executed compartmentalised. The Application Layer should be compartmentalised in a different compartment than the compartment that contains the Network Layer or Facility Layer. Preferably, two different CPUs are used to establish this separated compartmentalisation. If the Network Layer, Facility Layer (, Management Layer) and Application Layer are implemented on the same CPU, this CPU should support a Trusted Execution Environment. The Network Layer, Facility Layer and Management Layer should be compartmentalised in the most secure compartment(s) of the TEE-enabled CPU and the Application Layer in a different compartment. If the Application Layer is implemented on a different CPU than the Network Layer and Facility Layer, then the possibly insecure third-party applications can not directly affect the lower protocol stack layers, hence virtual compartmentalisation of the protocol stack layers is sufficient. Access to shared resources, by any of the protocol stack layers, should be managed by the Management Layer, e.g. access to the external memory unit.

Furthermore, both the VA and the SE should have an end-to-end secured (logical or physical) connection to the CPU that implements the Network Layer, if that does not affects signature verification performance (severely). That is, if the VA can obtain the required (root) certificates for certificate chain verification in time. If the performance is insufficient then the VA and SE should be physically interconnected directly. Possibly, if both the VA and SE can communicate with the CPU via different physical connections, as well as with each other via a physical connection, then we have obtained the best of both worlds. That is, end-to-end security for communication between the VA and CPU, between the VA and SE, and between the SE and CPU. Note, this requires the SE to have a second communication channel as well.

CHAPTER 7

Conclusions

The main goals of this research are twofold, finding security requirements for an ITS Station from the abstract perspective, and reconsidering the found requirements for an ITS Station from the perspective of two potential hardware implementation. In the next section we will recapitulate the findings of both research parts and in Section 7.2 we will provide advises for designing an ITS Station.

7.1 Recapitulation

Typically, ITS messages are public information and are broadcast over the ITS radio network. Hence, confidentiality of ITS messages is not a security issue. On the other hand, as processing incorrect or spurious ITS messages may affect traffic safety and efficiency, integrity and authenticity of ITS messages should be assured. Every protocol stack layer will perform checks on the trustworthiness of ITS messages by different mechanisms.

Although the Access Layer uses error detection and correction mechanisms to verify integrity of received ITS messages, the Access Layer can not detect maliciously injected, modified or deleted ITS messages. Nevertheless, the integrity and authenticity of ITS messages are assured by the digital signature. Hence, no additional security measures are required. Furthermore, although message deletion by the Access Layer is a critical risk, countermeasures will not be able to remove the threat of message deletion, as message deletion could have occurred before reception already, e.g. by jamming the radio network. Also, message replay will be detectable by the time stamp that is included in each ITS message signature at the time of signature generation. Hence, the Access Layer can be considered to be a plain AD and DA converter which not necessarily requires additional security measures.

The Network Layer requests the Verification Accelerator (VA) to verify the digital signature of each ITS message. However, the higher protocol stack layers will not be able to verify integrity and authenticity of the ITS message by the digital signature. Firstly, because the higher protocol stack layers will not perform a (second) digital signature verification due to performance reasons. Secondly, because the higher protocol stack layers will not have access to the complete ITS message, which is necessary for signature verification. Hence, integrity and authenticity of ITS messages must be assured by other means, after the verification. In general, the VA should attach the verification results to corresponding ITS message in a manner, such that the Facility Layer can detect malicious modification to the verification results, e.g. by a malicious Network Layer. However, if the Network Layer and Facility Layer are implemented in the same CPU and are compartmentalised together (or in different, but equally secure compartments), then assurance of integrity and authenticity of the communication between the VA and that particular CPU is sufficient. That is, no end-to-end security between the VA and Facility Layer is required.

The Facility Layer will "absorb" the ITS message contents into a database that represents the "situational awareness" of the traffic. ITS messages that are considered to be inconsistent with this database, e.g. by a plausibility check, may be discarded. Also, the Facility Layer should check the trustworthiness of messages, received from the Application Layer (e.g., collision warnings), to enable detectability of incorrectly or maliciously generated traffic or user notification.

The Application Layer will have to perform plausibility checks on the messages, received from the Facility Layer, as well.

The Management Layer will (partially) be implemented in all processing components (i.e., CPUs, VA and Secure Element), but will not act upon the ITS messages. However, the Management Layer should manage access to shared resources if multiple protocol stack layers are implemented on the same CPU. The Security Layer, and in particular the Secure Element (SE), must be highly secure, as digital signature generation, encryption/decryption and (root) certificate storage may not be influenced or monitored (maliciously).

The risk analysis indicated several threats to the ITS Station. Firstly, Man-In-The-Middle (MITM) attacks between protocol stack layers are the most critical threats. Stolen or unauthentic hardware or software could be considered the second most critical threat. To mitigate the risk of MITM attacks between protocol stack layers, integrity and authenticity of the messages must be assured. Hence, every pair of processing components must establish a shared key. Key-establishment should occur in a trusted environment, e.g., at the final stage of ITS Station assembly, and should only be performed once. Key-establishment in this way eases the process of key-establishment, as different ITS Station *component* manufacturers not necessarily need to cooperatively establish shared keys. Furthermore, whenever two ITS Station components have established a shared key, they can check the other's knowledge of this key afterwards. If the other component does not know the shared key (anymore), something may be wrong, e.g., one of the components may be substituted. Note, for the latter it is crucial to ensure key-diversification. That is, every ITS Station must (at least) have a different, or random, key for assuring integrity and authenticity of internal communication. Additionally, but not necessarily, different keys may be used to assure integrity and authenticity of communication between different ITS Station components pairs, e.g., one key for communication between the CPU-north and CPU-south, and one key for communication between CPU-south and the VA. A stolen or maliciously substituted ITS Station component can be detected as the shared keys will differ. Clearly, the shared key(s) must not leak. Hence, components must store the shared key(s) in secure memory.

Although integrity and authenticity assurance of communication internal to the ITS Station is protected, unauthentic software can still be installed by masquerading a genuine software author. Hence, software must be digitally signed and verified on its integrity and authenticity before installation. Furthermore, the process of updating software must comply with NIST's BIOS guidelines [20]. If compliance is realised, the firmware is protected from malicious modifications, can not be installed without the presence of the ITS Station user (e.g., vehicle owner) and is protected against rollback.

Finally, protocol stack layers must not be able to (maliciously) modify data of other protocol stack layers. If multiple protocol stack layers are implemented on one CPU, they must be executed compartmentalised. The Application Layer may support execution of (insecure) third-party applications which may affect lower protocol stack layers. Hence, if the Network Layer, Facility Layer, Management Layer and Application Layer are implemented on the same CPU, this CPU should support a Trusted Execution Environment, to mitigate the risk of insecure or malicious applications affecting the lower protocol stack layers. The Network Layer, Facility Layer and Management Layer should be compartmentalised in the most secure compartment of the (TEE-enabled) CPU and the Application Layer in a different compartment. On the other hand, if the Application Layer is implemented on a different CPU than the Network Layer and Facility Layer, virtual compartmentalisation of the protocol stack layers is sufficient, and hence, none of the CPU's require a TEE.

7.2 Design Recommendations

The risk analysis indicated several security critical aspects that should be taken into account when designing and implementing an ITS Station.

As the signature generation process, (root) certificate storage and encryption and decryption are the trust anchors for integrity, authenticity and confidentiality assurance, a secure element is required which provides (highly) secure processing and storage. An example of such secure element is the (Common Criteria AEL6+ certified) *SmartMX2* secure smart card product family by NXP Semiconductors [32].

As MITM attacks on the communication between protocol stack layers are the most critical security risk to the ITS Station, ITS Station components should establish a shared key in a secure environment. This should be done only once, to enable detectability of stolen ITS Station components. Note, it is crucial to ensure key-diversification among ITS Stations. That is, every ITS Station must (at least) have a different, or random, key for assuring integrity and authenticity of internal communication. Additionally, different keys should be used to assure integrity and authenticity of communication between different ITS Station components pairs, e.g., one key for communication between the CPU-north and CPU-south, and one key for communication between CPU-south and the VA. This will mitigate the impact of a compromise of a shared keys that is used to assure communication between protocol stack layers. Also, shared keys will enable detectability of ITS Station components that do not know the (previously) shared key. If an ITS Station component does not know the (previously) shared key anymore, one or both components could have been maliciously modified or substituted.

Firmware and application updates should be digitally signed and verified before installation. This enables detectability of masquerading attacks on genuine software authors. Furthermore, the process of installing or updating firmware or applications must comply with NIST's BIOS guidelines.

If multiple protocol stack layers are implemented on one CPU, they must be executed compartmentalised. If the Network Layer, Facility Layer, Management Layer and Application Layer are implemented on the same CPU, this CPU should support a Trusted Execution Environment to mitigate the risk of insecure or malicious applications affecting the lower protocol stack layers. The Network Layer, Facility Layer and Management Layer should be compartmentalised in the most secure compartment(s) of the TEE-enabled CPU and the Application Layer in a different compartment. Access to shared resources, by any of the protocol stack layers, should be managed by the Management Layer, e.g. access to the external memory unit. If the Application Layer is implemented on a different CPU than the Network Layer and Facility Layer, virtual compartmentalisation of the protocol stack layers is sufficient.

Furthermore, both the VA and the SE should have an end-to-end secured (logical or physical) connection to the CPU that implements the Network Layer, if that does not affects signature verification performance (severely). That is, if the VA can obtain the required (root) certificates for certificate chain verification in time. If the performance is insufficient then the VA and SE should be physically interconnected directly. Possibly, if both the VA and SE can communicate with the CPU via different physical connections, as well as with each other via a physical connection, then we have obtained the best of both worlds. That is, end-to-end security for communication between the VA and CPU, between the VA and SE, and between the SE and CPU. Note, this requires the SE to have a second (physical) connection as well.

The Local Dynamic Map (LDM), maintained by the Facility Layer, should be stored encrypted. This prevents attackers, who have access to the memory that contains the LDM, from (uniquely) identifying ITS Stations by tracking changes in the LDM.

The set of pseudonym certificates, maintained by the Network Layer, should be stored encrypted. This prevents attackers, who gained access to the memory that contains the pseudonym certificates, from (uniquely) identifying ITS Stations by tracking pseudonym certificate changes.

7.3 Future Work

Firstly, the risk analysis is performed on the generic communication architecture of an ITS Station. Based on corresponding risk estimations, prioritised requirements are argued and reconsidered for two hardware partitionings. However, these requirements may not be applicable to partitionings that differ from our partitionings. Hence, if the partitioning differs from ours, care must be taken when considering our requirements.

Secondly, our risk and security analyses were scoped down to safety-related traffic communication. Further research will be necessary to check if non-safety-related communication changes the requirements we indicated or even introduces new requirements.

Finally, further research is required to determine the exact impact on the ITS applications when the hop-limit value—part of multi-hop ITS messages—is not secured as suggested in Chapter 3.

Bibliography

[1] *Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Methods and protocols; Part 1: Method and proforma for Threat, Risk, Vulnerability Analysis*, Tech. report, European Telecommunications Standards Institute.

[2] *Trusted execution environments*, Website: `http://www.globalplatform.org/mediapressview.asp?id=838`.

[3] *ETSI ITS - Standards on the move*, Tech. report, European Telecommunications Standards Institute, 2008, Website: `http://www.etsi.org/WebSite/document/Technologies/ETSIpresentations/ITS_Congress_Geneva_Slideshow.pdf`.

[4] *Intelligent Transport Systems (ITS); European profile standard for the physical and medium access control layer of Intelligent Transport Systems operating in the 5 GHz frequency band*, Tech. Report ETSI ES 202 663, European Telecommunications Standards Institute, 2009.

[5] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions*, Tech. Report ETSI TR 102 638, European Telecommunications Standards Institute, 2009.

[6] *Intelligent Transport Systems (ITS); Communications Architecture*, Tech. Report ETSI EN 302 665, European Telecommunications Standards Institute, 2010.

[7] *Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA)*, Tech. Report ETSI TR 102 893, European Telecommunications Standards Institute, 2010.

[8] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specification of Decetralized Environmental Notification Basic Service*, Tech. Report ETSI TS 102 637-3, European Telecommunications Standards Institute, 2010.

[9] *Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*, Tech. Report ETSI TS 102 637-2, European Telecommunications Standards Institute, 2011.

[10] *Intelligent Transport Systems (ITS); Vehicular communications; GeoNetworking*, Tech. Report ETSI TS 102 636, European Telecommunications Standards Institute, 2011.

[11] *Intelligent Transport Systems (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*, Tech. Report ETSI TS 102 636-4, European Telecommunications Standards Institute, 2011.

[12] *Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 5: Transport Protocols*, Tech. Report ETSI TS 102 636-5-1, European Telecommunications Standards Institute, 2011.

[13] *Intelligent Transport Systems; OSI cross-layer topics; Part 1: Architecture and addressing schemes*, Tech. Report ETSI TS 102 723-1, European Telecommunications Standards Institute, 2012.

[14] L. Buttyán, T. Holczer, and I. Vajda, *On the effectiveness of changing pseudonyms to provide location privacy in vanets*, Springer, 2007, pp. 129–141.

[15] Car-2-Car Communication Consortium, Website: `http://www.car-to-car.org`.

[16] V. Carlier, H. Chabanne, E. Dottax, and H. Pelletier, *Electromagnetic side channels of an fpga implementation of aes*, 2004.

[17] Coley Consultion, *MoSCoW Prioritisation*, Website: `http://www.coleyconsulting.co.uk/moscow.html`.

[18] European Commission, *Standardisation mandate addressed to cen, cenelec and etsi in the field of information and communication technologies to support the interoperability of co-operative systems for intelligent transport in the european community*, October 2009, European Mandate M/453.

[19] Common Criteria, *Common Criteria for Information Technology Security Evaluation*, Website: `http://www.commoncriteriaportal.org/`.

[20] David Cooper, William Polk, Andrew Regenscheid, and Murugiah Souppaya, *Bios protection guidelines*, Tech. report, National Institute of Standards and Technology – Information Technology Laboraty – Computer Security Division, April 2011.

[21] ERTICO ITS Europe, Website: `http://www.ertico.com`.

[22] European Telecommunications Standards Institute, *ETSI ITS - Standards on the move*, Website: `http://www.etsi.org/WebSite/document/Technologies/LEAFLETS/CooperativeITS.pdf`.

[23] EVITA Project co-funded by European Union, *E-safety vehicle intrusion protection application*, Website: `http://www.evita-project.org`.

[24] A.B. Gürdag and M.U. Caglayan, *A formal security analysis of secure aodv (saodv) using model checking.*

[25] IEEE Standards Association, *IEEE 802.11p - Amendment 6: Wireless Access in Vehicular Environments*, 2010, Website: http://standards.ieee.org/getieee802/download/802.11p-2010.pdf.

[26] National Instruments, *Virtualisation*, Website: `http://www.ni.com/white-paper/8709/en`.

[27] *Intelligent Transport Systems - Communications access for land mobiles (CALM) - Management*, 2010.

[28] ITS America, Website: `http://www.itsa.org`.

[29] ITS Japan, Website: `http://its-jp.org/english`.

[30] ITS World Congress, Website: `http://www.itsworldcongress.com`.

[31] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot, *Handbook of applied cryptography*, 1st ed., CRC Press, Inc., Boca Raton, FL, USA, 1996.

[32] NXP, *SmartMX2 Interface Controller*, Website: `http://www.nxp.com/products/identification_and_security/smart_card_ics/smartmx2/`.

[33] P. Papadimitratos, V. Gligor, and J-P. Hubaux, *Securing vehicular communications - assumptions, requirements, and principles*, (2006), 5–14.

[34] C. Perkins, E. Belding-Royer, and S. Das, *Ad hoc on-demand distance vector (aodv) routing*, 2003.

[35] Planet3dNow.de, *Trusted execution environments*, Website: `http://www.planet3dnow.de/cgi-bin/newspub/viewnews.cgi?id=1339667038`.

[36] SESP, *Patrol-PX Radio Jammer*, Website: `http://www.sesp.com/PortableTacticalJammers.asp/`.

[37] M.G. Zapata and N. Asokan, *Securing ad hoc routing protocols*, Proceedings of the 1st ACM workshop on Wireless security, ACM, 2002, pp. 1–10.

Appendices

APPENDIX A

Security Risk Estimations

Section A.1 explains aspects of the risk estimations table of Section A.2.2

A.1 Explanation of risk aspects

Time The worst-case complete time span—i.e., shortest time span—required to identify one or more vulnerabilities, create an attack and successfully execute the attack.

value	*description*
(0)	$\leq$ *1 day*
(1)	$\leq$ *1 week*
(4)	$\leq$ *1 month*
(13)	$\leq$ *3 month*
(26)	$\leq$ *6 month*
(999)	$>$ *6 month*

Expertise Required generic knowledge of the underlying principles, product type or attack methods to develop and execute a successful attack.

value	*description*
(0)	*Laymen*: not knowledgeable compared to experts or proficient persons, i.e., no particular expertise
(2)	*Proficient*: familiar with the security behaviour of the product of system type
(5)	*Expert*: familiar with the underlying algorithms, protocols, hardware, structures, security behaviour, principles and concepts of security employed, techniques and tools for definition of new attacks, cryptography, classical attacks for product type, attack methods, etc., implemented in the product or system type.

Knowledge Required knowledge of the asset.

value	*description*

(0)	*Public*: publicly available information about the asset (e.g., as gained from the Internet)
(1)	*Restricted*: restricted information about the asset, i.e, knowledge that is controlled within the developer organisation and shared with other organisations under a non-disclosure agreement
(4)	*Sensitive*: sensitive information about the asset, i.e., knowledge that is shared between discreet teams within the developer organisation, access to which is constrained only to members of the specified teams
(10)	*Critical*: critical information about the asset, i.e, knowledge that is known by only a few individuals, access to which is very tightly controlled on a strict need-to-know basis and individual undertaking

Opportunity The time window required to execute the attack and the difficulty to access the attack interface determine the opportunity to execute a successful attack.

value	*description*
(0)	*Unnecessary*: access to the asset is not needed
(1)	*Easy*: access to the asset for less that 1 day and attack interface is between ITS protocol layers
(4)	*Moderate*: access to the asset for less that 1 day and attack interface is between ITS protocol layer component(s)
(12)	*Difficult*: access to the asset for multiple days and attack interface is on/in ITS protocol layer component(s)
(999)	*None*: no opportunity exists

Equipment IT hardware/software or other equipment refers to the equipment required to identify or exploit a vulnerability.

value	*description*
(0)	*Standard*: readily available (may be part of the asset)
(3)	*Specialised*: not readily available, but can be acquired without undue effort
(7)	*Bespoke*: not readily available and requires high cost and/or is highly specialised

Asset impact The asset impact is scoped down to the ITS Station, i.e., the basis of business of an ITS Station.

value	*description*
(1)	*Low*: concerned party is not harmed, possible damage is low
(2)	*Medium*: threatening interest of subscriber/provider, non-negligible
(3)	*High*: basis of business is threatened, severe damage

Intensity In case of a successfully executed attack, the number of affected protocols stack layers is indicated by the Intensity factor.

value	*description*
(0)	*Single instance*: only one ITS protocol layer is affected
(1)	*Moderate intensity*: multiple ITS protocol layers are affected, but is detectable
(2)	*High intensity*: multiple ITS protocol layers are affected and is not detectable

A.2 Risk Estimations

A.2.1 Example

For example, consider the threat *Execution of insecure firmware*, achieved by means of an *Radio transceiving equipment : replay (injection only)* attack. The complete *time* to develop and execute this attack is considered to be (i) "less or equal to 1 week" (corresponding value: *0*, the attacker *expertise* is required to be (ii) "at least familiar with the attacked security behaviour" (corresponding value: *2*), he should have (iii) *knowledge* of "shared secret information (information under an non-disclosure agreement, only known among discrete teams within developers organisation)" (corresponding value: *4*), the *opportunity* to execute the attack is present whenever he has (iv) "access to the targeted ITS-S for less than a day and can be deployed between protocol layer implementations" (corresponding value: *1*), and the *equipment* with which the attack is executed is (v) "not readily available but can be acquired with undue effort" (corresponding value: *3*). Furthermore, execution of the attack on or at the *access layer* has "low *impact*", (vi), (corresponding value: *1*) on the ITS-S and the (vii) *intensity* is "low as only that layer is affected" (corresponding value: *0*). The final risk determination is calculated as follows: $(i) + (ii) + (iii) + (iv) + (v) = a$
Occurence, (b), of the attack is considered to be "unlikely" (corresponding value: *1*) whenever $a \geq 15$, "possible" (corresponding value: *2*) whenever $7 \leq a \leq 4$ or "likely" (corresponding value: *3*) whenever $1 \leq a \leq 6$. The total impact, (c), (i.e., combination of *asset impact* and *intensity*) of a successful attack is simply the sum of both ratings (limited up to *3*), thus $(c) = (vi) + (vii)$ and $1 \leq c \leq 3$. The final risk estimation is considered to be "Minor" whenever $(b)(c) \in 1, 2$ ("b times c equals either 1 or 2"), "Major" whenever $(b)(c) \in 3, 4$ and "Critical" whenever $(b)(c) \in 6, 9$. Note, in this example, the attack location (as indicated in column 5 of Table A.2.2 is determined by the evaluated threat agent (which is applied on the air interface in this case). Estimating the risk of such an attack at the, say, Network layer, thus is not applicable.

In this risk analysis an attack can be executed on different layers in the ITS protocol stack. That is, the used threat agent is executed on or at the Access Layer, Network Layer, Facility Layer, Application Layer or Security Layer.

A.2.2 Risk Table

Threat group	Threat agent	Attack				Risk
		Factor	Range	Attack Location	Imp.+Int.	
Execution of insecure firmware	MITM equipment	Time	:≤ 1 month (4)	Access	3 + 2	Critical
		Expertise	: Expert (5)	Network	2 + 1	Critical
		Knowledge	: Restricted (1)	Facility	3 + 2	Critical
		Accessibility	: Easy (1)	Application	3 + 2	Critical
		Equipment	: Specialised (3)	Security	3 + 2	Critical
		Difficulty	: Moderate (14)	Management	2 + 1	Critical
		Likelihood	: Possible (2)			
	Radio transceiver (masquerading genuine software author)	Time	:≤ 1 month (4)	Access	3 + 2	Critical
		Expertise	: Expert (5)	Network	N/A	N/A
		Knowledge	: Restricted (1)	Facility	N/A	N/A
		Accessibility	: Easy (1)	Application	N/A	N/A
		Equipment	: Specialised (3)	Security	N/A	N/A
		Difficulty	: Moderate (14)	Management	N/A	N/A
		Likelihood	: Possible (2)			
	Maliciously modifying stored data	Time	:≤ 3 months (13)	Access	1 + 1	Minor
		Expertise	: Expert (5)	Network	2 + 1	Major
		Knowledge	: Sensitive (4)	Facility	3 + 2	Major
		Accessibility	: Difficult (12)	Application	3 + 2	Major
		Equipment	: Specialised (3)	Security	3 + 2	Major
		Difficulty	: > High (37)	Management	2 + 1	Major
		Likelihood	: Unlikely (1)			
	Stolen firmware components	Time	:≤ 1 week (1)	Access	1 + 0	Minor
		Expertise	: Proficient (2)	Network	1 + 1	Minor
		Knowledge	: Restricted (1)	Facility	2 + 1	Major
		Accessibility	: Difficult (12)	Application	2 + 1	Major
		Equipment	: Standard (0)	Security	3 + 2	Major
		Difficulty	: High (16)	Management	1 + 1	Minor
		Likelihood	: Unlikely (1)			
	Counterfeit insecure ITS-S components	Time	:≤ 3 months (13)	Access	1 + 1	Minor
		Expertise	: Expert (5)	Network	2 + 1	Major
		Knowledge	: Sensitive (4)	Facility	3 + 2	Major
		Accessibility	: Difficult (12)	Application	3 + 2	Major
		Equipment	: Bespoke (7)	Security	3 + 2	Major
		Difficulty	: > High (41)	Management	2 + 1	Major
		Likelihood	: Unlikely (1)			

Threat group	Threat agent	Attack				Risk
		Factor	Range	Attack Location	Imp.+Int.	
	Physical environment monitoring or influencing equipment	Time	:> 6 months(999)	Access	1 + 1	Minor
		Expertise	: Proficient (2)	Network	2 + 1	Major
		Knowledge	: Public (0)	Facility	3 + 2	Major
		Accessibility	: Difficult (12)	Application	3 + 2	Major
		Equipment	: Specialised (3)	Security	3 + 2	Major
		Difficulty	: > High (1016)	Management	2 + 1	Major
		Likelihood	: Unlikely (1)			
Execution of insecure applications (at Application Layer)	Use of insecure firmware	Time	:≤ 3 months (13)	Access	2 + 2	Major
		Expertise	: Expert (5)	Network	2 + 2	Major
		Knowledge	: Sensitive (4)	Facility	3 + 2	Major
		Accessibility	: Moderate (4)	Application	3 + 2	Major
		Equipment	: Bespoke (7)	Security	3 + 2	Major
		Difficulty	: > High (30)	Management	2 + 2	Major
		Likelihood	: Unlikely (1)			
	Radio transceiving equipment (masquerading software author)	Time	:≤ 1 month (4)	Access	2 + 2	Critical
		Expertise	: Expert (5)	Network	N/A	N/A
		Knowledge	: Restricted (1)	Facility	N/A	N/A
		Accessibility	: Easy (1)	Application	N/A	N/A
		Equipment	: Specialised (3)	Security	N/A	N/A
		Difficulty	: Moderate (14)	Management	N/A	N/A
		Likelihood	: Possible (2)			
	MITM equipment	Time	:≤ 1 month (4)	Access	2 + 2	Critical
		Expertise	: Proficient (2)	Network	2 + 2	Critical
		Knowledge	: Restricted (1)	Facility	2 + 2	Critical
		Accessibility	: Easy (1)	Application	2 + 2	Critical
		Equipment	: Specialised (3)	Security	3 + 2	Critical
		Difficulty	: Moderate (11)	Management	2 + 2	Critical
		Likelihood	: Possible (2)			
	Process (data) monitoring or influencing equipment	Time	:≤ 3 months (13)	Access	1 + 0	Minor
		Expertise	: Expert (5)	Network	2 + 1	Major
		Knowledge	: Sensitive (4)	Facility	2 + 2	Major
		Accessibility	: Moderate (4)	Application	2 + 2	Major
		Equipment	: Specialised (3)	Security	3 + 2	Major
		Difficulty	: > High (29)	Management	2 + 1	Major
		Likelihood	: Unlikely (1)			

Threat group	Threat agent	Attack				Risk
		Factor	Range	Attack Location	Imp.+Int.	
	Stolen firmware component(s)	Time	:≤ 1 week (1)	Access	1 + 0	Minor
		Expertise	: Proficient (2)	Network	2 + 1	Major
		Knowledge	: Restricted (1)	Facility	2 + 1	Major
		Accessibility	: (14)	Application	2 + 1	Major
		Equipment	: Standard (0)	Security	3 + 2	Major
		Difficulty	: High (16)	Management	2 + 1	Major
		Likelihood	: Unlikely (1)			
	Counterfeit insecure ITS-S components	Time	:≤ 3 months (13)	Access	1 + 0	Minor
		Expertise	: Expert (5)	Network	2 + 2	Major
		Knowledge	: Sensitive (4)	Facility	2 + 2	Major
		Accessibility	: Difficult (12)	Application	2 + 2	Major
		Equipment	: Bespoke (7)	Security	3 + 2	Major
		Difficulty	: > High (41)	Management	2 + 2	Major
		Likelihood	: Unlikely (1)			
	Physical environment monitoring or influencing equipment	Time	:> 6 months(999)	Access	1 + 0	Minor
		Expertise	: Proficient (2)	Network	2 + 2	Major
		Knowledge	: Public (0)	Facility	2 + 2	Major
		Accessibility	: Easy (1)	Application	2 + 2	Major
		Equipment	: Specialised (3)	Security	3 + 2	Minor
		Difficulty	: > High (1005)	Management	2 + 2	Major
		Likelihood	: Unlikely (1)			
Malicious or mischievous manupulation of process data	Insecure firmware	Time	:≤ 3 months (13)	Access	1 + 1	Minor
		Expertise	: Expert (5)	Network	3 + 1	Major
		Knowledge	: Sensitive (4)	Facility	3 + 2	Major
		Accessibility	: Moderate (4)	Application	3 + 2	Major
		Equipment	: Bespoke (7)	Security	3 + 2	Major
		Difficulty	: > High (41)	Management	3 + 1	Major
		Likelihood	: Unlikely (1)			
	Insecure applications	Time	:≤ 1 month (4)	Access	N/A	N/A
		Expertise	: Proficient (2)	Network	N/A	N/A
		Knowledge	: Public (0)	Facility	N/A	N/A
		Accessibility	: Easy (1)	Application	3 + 2	Critical
		Equipment	: Specialised (3)	Security	N/A	N/A
		Difficulty	: Moderate (10)	Management	N/A	N/A
		Likelihood	: Possible (2)			

Threat group	Threat agent	Attack				Risk
		Factor	Range	Attack Location	Imp.+Int.	
	Stolen firmware components	Time	:≤ 1 week (1)	Access	1 + 0	Major
		Expertise	: Proficient (2)	Network	1 + 1	Critical
		Knowledge	: Restricted (1)	Facility	2 + 1	Critical
		Accessibility	: Easy (1)	Application	2 + 1	Critical
		Equipment	: Standard (0)	Security	3 + 2	Critical
		Difficulty	: Basic (5)	Management	1 + 1	Critical
		Likelihood	: Likely (3)			
	Counterfeit insecure ITS-S components	Time	:≤ 3 months (13)	Access	1 + 1	Minor
		Expertise	: Expert (5)	Network	3 + 2	Major
		Knowledge	: Sensitive (4)	Facility	3 + 2	Major
		Accessibility	: Moderate (4)	Application	3 + 2	Major
		Equipment	: Bespoke (7)	Security	3 + 2	Major
		Difficulty	: > High (41)	Management	3 + 2	Major
		Likelihood	: Unlikely (1)			
	Physical environment monitoring or influencing equipment	Time	:> 6 months(999)	Access	1 + 1	Minor
		Expertise	: Expert (5)	Network	3 + 1	Major
		Knowledge	: Restricted (1)	Facility	3 + 2	Major
		Accessibility	: Difficult (12)	Application	3 + 2	Major
		Equipment	: Specialised (3)	Security	3 + 2	Major
		Difficulty	: > High (1020)	Management	3 + 1	Major
		Likelihood	: Unlikely (1)			
Message injection, modification or deletion	Radio transceiving equipment; replay (injection only)	Time	:≤ 1 week (1)	Access	1 + 1	Major
		Expertise	: Proficient (2)	Network	N/A	N/A
		Knowledge	: Public (0)	Facility	N/A	N/A
		Accessibility	: Easy (1)	Application	N/A	N/A
		Equipment	: Specialised (3)	Security	N/A	N/A
		Difficulty	: Moderate (7)	Management	N/A	N/A
		Likelihood	: Possible (2)			
	Radio transceiving equipment; self-created (injection only)	Time	:≤ 1 week (1)	Access	1 + 1	Major
		Expertise	: Proficient (2)	Network	N/A	N/A
		Knowledge	: Public (0)	Facility	N/A	N/A
		Accessibility	: Easy (1)	Application	N/A	N/A
		Equipment	: Specialised (3)	Security	N/A	N/A
		Difficulty	: Moderate (7)	Management	N/A	N/A
		Likelihood	: Possible (2)			

Threat group	Threat agent	Attack				Risk
		Factor	Range	Attack Location	Imp.+Int.	
	MITM equipment	Time	:$\leq$ 1 month (4)	Access	2 + 1	Critical
		Expertise	: Proficient (2)	Network	2 + 2	Critical
		Knowledge	: Restricted (1)	Facility	3 + 2	Critical
		Accessibility	: Moderate (4)	Application	3 + 2	Critical
		Equipment	: Specialised (3)	Security	3 + 2	Critical
		Difficulty	: Moderate (14)	Management	2 + 2	Critical
		Likelihood	: Possible (2)			
	Malicious or insecure firmware	Time	:$\leq$ 6 months (26)	Access	1 + 1	Minor
		Expertise	: Expert (5)	Network	2 + 1	Major
		Knowledge	: Sensitive (4)	Facility	3 + 2	Major
		Accessibility	: Moderate (4)	Application	3 + 2	Major
		Equipment	: Bespoke (7)	Security	3 + 2	Major
		Difficulty	: $>$ High (43)	Management	2 + 1	Major
		Likelihood	: Unlikely (1)			
	Malicious or incorrect applications	Time	:$\leq$ 3 months (13)	Access	N/A	N/A
		Expertise	: Expert (5)	Network	N/A	N/A
		Knowledge	: Sensitive (4)	Facility	N/A	N/A
		Accessibility	: Easy (1)	Application	3 + 2	Major
		Equipment	: Bespoke (7)	Security	N/A	N/A
		Difficulty	: $>$ High (30)	Management	N/A	N/A
		Likelihood	: Unlikely (1)			
	Stolen firmware components	Time	:$\leq$ 1 week (1)	Access	1 + 1	Major
		Expertise	: Proficient (2)	Network	1 + 1	Critical
		Knowledge	: Restricted (1)	Facility	2 + 1	Critical
		Accessibility	: Easy (1)	Application	2 + 1	Critical
		Equipment	: Standard (0)	Security	3 + 2	Critical
		Difficulty	: Basic (5)	Management	1 + 1	Critical
		Likelihood	: Likely (3)			
	Counterfeit insecure ITS-S components	Time	:$\leq$ 3 months (13)	Access	1 + 1	Minor
		Expertise	: Expert (5)	Network	3 + 1	Major
		Knowledge	: Sensitive (4)	Facility	3 + 2	Major
		Accessibility	: Moderate (4)	Application	3 + 2	Major
		Equipment	: Bespoke (7)	Security	3 + 2	Major
		Difficulty	: $>$ High (41)	Management	3 + 1	Major
		Likelihood	: Unlikely (1)			

Threat group	**Threat agent**	**Attack**				**Risk**
		Factor	**Range**	**Attack Location**	**Imp.+Int.**	
	Physical environment monitoring or influencing equipment	Time	:> 6 months(999)	Access	1 + 1	Minor
		Expertise	: Proficient (2)	Network	3 + 1	Major
		Knowledge	: Restricted (1)	Facility	3 + 2	Major
		Accessibility	: Difficult (12)	Application	3 + 2	Major
		Equipment	: Bespoke (7)	Security	3 + 2	Major
		Difficulty	: > High (1016)	Management	3 + 1	Major
		Likelihood	: Unlikely (1)			
Complete, selective or random DoS to (non-)ITS messages	Radio jammer	Time	:≤ 1 week (1)	Access	2 + 2	Critical
		Expertise	: Laymen (0)	Network	N/A	N/A
		Knowledge	: Public (0)	Facility	N/A	N/A
		Accessibility	: Easy (1)	Application	N/A	N/A
		Equipment	: Standard (0)	Security	N/A	N/A
		Difficulty	: No rating (2)	Management	N/A	N/A
		Likelihood	: Likely (3)			
	Radio transceiver (saturating the radio channel(s)	Time	:≤ 1 week (1)	Access	2 + 2	Critical
		Expertise	: Proficient (2)	Network	N/A	N/A
		Knowledge	: Restricted (1)	Facility	N/A	N/A
		Accessibility	: Easy (1)	Application	N/A	N/A
		Equipment	: Specialised (3)	Security	N/A	N/A
		Difficulty	: Moderate (8)	Management	N/A	N/A
		Likelihood	: Possible (2)			
	Malicious or insecure firmware	Time	:≤ 6 months (26)	Access	3 + 2	Major
		Expertise	: Expert (5)	Network	3 + 2	Major
		Knowledge	: Sensitive (4)	Facility	3 + 2	Major
		Accessibility	: Moderate (4)	Application	3 + 2	Major
		Equipment	: Bespoke (7)	Security	3 + 2	Major
		Difficulty	: > High (43)	Management	3 + 2	Major
		Likelihood	: Unlikely (1)			
	MITM equipment	Time	:≤ 1 week (1)	Access	3 + 2	Critical
		Expertise	: Laymen (0)	Network	3 + 2	Critical
		Knowledge	: Restricted (1)	Facility	3 + 2	Critical
		Accessibility	: Easy (1)	Application	3 + 2	Critical
		Equipment	: Standard (0)	Security	3 + 2	Critical
		Difficulty	: Basic (3)	Management	3 + 2	Critical
		Likelihood	: Likely (3)			

Threat group	Threat agent	Attack				Risk
		Factor	Range	Attack Location	Imp.+Int.	
	Malicious or incorrect applications	Time	:≤ 3 months (13)	Access	N/A	N/A
		Expertise	: Expert (5)	Network	N/A	N/A
		Knowledge	: Sensitive (4)	Facility	N/A	N/A
		Accessibility	: Easy (1)	Application	3 + 2	Major
		Equipment	: Bespoke (7)	Security	N/A	N/A
		Difficulty	: > High (30)	Management	N/A	N/A
		Likelihood	: Unlikely (1)			
	Stolen firmware components	Time	:≤ 1 week (1)	Access	1 + 1	Critical
		Expertise	: Proficient (2)	Network	2 + 1	Critical
		Knowledge	: Restricted (1)	Facility	2 + 1	Critical
		Accessibility	: Easy (1)	Application	2 + 1	Critical
		Equipment	: Standard (0)	Security	3 + 2	Critical
		Difficulty	: Basic (5)	Management	2 + 1	Critical
		Likelihood	: Likely (3)			
	Counterfeit insecure ITS-S components	Time	:≤ 3 months (13)	Access	1 + 1	Minor
		Expertise	: Expert (5)	Network	3 + 1	Major
		Knowledge	: Sensitive (4)	Facility	3 + 2	Major
		Accessibility	: Moderate (4)	Application	3 + 2	Major
		Equipment	: Bespoke (7)	Security	3 + 2	Major
		Difficulty	: > High (33)	Management	3 + 1	Major
		Likelihood	: Unlikely (1)			
	Physical environment monitoring or influencing equipment	Time	:≤ 1 week (1)	Access	2 + 2	Critical
		Expertise	: Proficient (2)	Network	2 + 2	Critical
		Knowledge	: Public (0)	Facility	3 + 2	Critical
		Accessibility	: Easy (1)	Application	3 + 2	Critical
		Equipment	: Specialised (3)	Security	3 + 2	Critical
		Difficulty	: Moderate (7)	Management	2 + 2	Critical
		Likelihood	: Possible (2)			
Eavesdropping	MITM equipment	Time	:≤ 1 week (1)	Access	1 + 0	Minor
		Expertise	: Proficient (2)	Network	2 + 0	Major
		Knowledge	: Restricted (1)	Facility	2 + 0	Major
		Accessibility	: Easy (1)	Application	2 + 0	Major
		Equipment	: Specialised (3)	Security	3 + 2	Major
		Difficulty	: Moderate (8)	Management	2 + 0	Major
		Likelihood	: Possible (2)			

Threat group	Threat agent	Attack				Risk
		Factor	Range	Attack Location	Imp.+Int.	
	Malicious or insecure firmware	Time	:≤ 6 months (26)	Access	1 + 0	Minor
		Expertise	: Expert (5)	Network	2 + 1	Major
		Knowledge	: Sensitive (4)	Facility	2 + 1	Major
		Accessibility	: Moderate (4)	Application	2 + 1	Major
		Equipment	: Bespoke (7)	Security	3 + 2	Major
		Difficulty	: > High (46)	Management	2 + 1	Major
		Likelihood	: Unlikely (1)			
	Malicious or incorrect applications	Time	:≤ 3 months (13)	Access	N/A	N/A
		Expertise	: Expert (5)	Network	N/A	N/A
		Knowledge	: Sensitive (4)	Facility	N/A	N/A
		Accessibility	: Easy (1)	Application	2 + 1	Major
		Equipment	: Bespoke (7)	Security	N/A	N/A
		Difficulty	: > High (30)	Management	N/A	N/A
		Likelihood	: Unlikely (1)			
	Stolen firmware components	Time	:≤ 1 week (1)	Access	1 + 0	Major
		Expertise	: Proficient (2)	Network	1 + 0	Major
		Knowledge	: Restricted (1)	Facility	1 + 0	Major
		Accessibility	: Easy (1)	Application	2 + 0	Critical
		Equipment	: Standard (0)	Security	3 + 2	Critical
		Difficulty	: Basic (5)	Management	1 + 0	Major
		Likelihood	: Likely (3)			
	Counterfeit insecure ITS-S components	Time	:≤ 3 months (13)	Access	1 + 0	Minor
		Expertise	: Expert (5)	Network	1 + 0	Minor
		Knowledge	: Sensitive (4)	Facility	2 + 1	Major
		Accessibility	: Moderate (4)	Application	2 + 1	Major
		Equipment	: Bespoke (7)	Security	3 + 2	Major
		Difficulty	: > High (33)	Management	1 + 0	Minor
		Likelihood	: Unlikely (1)			
	Physical environment monitoring or influencing equipment	Time	:> 6 months(999)	Access	1 + 0	Minor
		Expertise	: Proficient (2)	Network	2 + 0	Minor
		Knowledge	: Public (0)	Facility	2 + 0	Minor
		Accessibility	: Difficult (12)	Application	2 + 0	Minor
		Equipment	: Specialised (3)	Security	3 + 2	Critical
		Difficulty	: > High (1016)	Management	2 + 0	Minor
		Likelihood	: Unlikely (1)			

Threat group	Threat agent	Attack				Risk
		Factor	**Range**	**Attack Location**	**Imp.+Int.**	
Denial of message transmission or reception	Inexistence of auditable log	Time	:≤ 1 day (0)	Access	1 + 0	Major
		Expertise	: Laymen (0)	Network	1 + 0	Major
		Knowledge	: Public (0)	Facility	1 + 1	Critical
		Accessibility	: Unnecessary (0)	Application	1 + 1	Critical
		Equipment	: Standard (0)	Security	1 + 1	Critical
		Difficulty	: No rating (0)	Management	1 + 0	Major
		Likelihood	: Likely (3)			
	MITM equipment	Time	:≤ 1 month (4)	Access	1 + 0	Minor
		Expertise	: Proficient (2)	Network	1 + 0	Minor
		Knowledge	: Restricted (1)	Facility	1 + 0	Minor
		Accessibility	: Easy (1)	Application	1 + 0	Minor
		Equipment	: Specialised (3)	Security	1 + 0	Minor
		Difficulty	: Moderate (11)	Management	1 + 0	Minor
		Likelihood	: Possible (2)			
	Malicious firmware	Time	:≤ 6 months (26)	Access	1 + 0	Minor
		Expertise	: Expert (5)	Network	2 + 0	Minor
		Knowledge	: Sensitive (4)	Facility	2 + 1	Major
		Accessibility	: Easy (1)	Application	2 + 1	Major
		Equipment	: Bespoke (7)	Security	3 + 2	Critical
		Difficulty	: > High (43)	Management	2 + 0	Minor
		Likelihood	: Unlikely (1)			
	Malicious applications	Time	:≤ 3 months (13)	Access	N/A	N/A
		Expertise	: Expert (5)	Network	N/A	N/A
		Knowledge	: Sensitive (4)	Facility	N/A	N/A
		Accessibility	: Easy (1)	Application	2 + 1	Major
		Equipment	: Bespoke (7)	Security	N/A	N/A
		Difficulty	: > High (30)	Management	N/A	N/A
		Likelihood	: Unlikely (1)			
	Stolen firmware components	Time	:≤ 1 week (1)	Access	1 + 0	Major
		Expertise	: Proficient (2)	Network	1 + 0	Major
		Knowledge	: Restricted (1)	Facility	1 + 0	Major
		Accessibility	: Easy (1)	Application	1 + 0	Major
		Equipment	: Standard (0)	Security	3 + 2	Critical
		Difficulty	: Basic (5)	Management	1 + 0	Major
		Likelihood	: Likely (3)			

Threat group	**Threat agent**	**Attack**				**Risk**
		Factor	**Range**	**Attack Location**	**Imp.+Int.**	
	Counterfeit insecure ITS-S components	Time	:≤ 3 months (13)	Access	$1 + 0$	Minor
		Expertise	: Expert (5)	Network	$2 + 0$	Minor
		Knowledge	: Sensitive (4)	Facility	$2 + 1$	Major
		Accessibility	: Moderate (4)	Application	$2 + 1$	Major
		Equipment	: Bespoke (7)	Security	$3 + 2$	Major
		Difficulty	: > High (33)	Management	$2 + 0$	Minor
		Likelihood	: Unlikely (1)			
	Physical environment monitoring or influencing equipment	Time	:≤ 3 months (13)	Access	$1 + 0$	Minor
		Expertise	: Proficient (2)	Network	$1 + 0$	Minor
		Knowledge	: Public (0)	Facility	$2 + 1$	Major
		Accessibility	: Difficult (12)	Application	$2 + 1$	Major
		Equipment	: Specialised (3)	Security	$2 + 2$	Major
		Difficulty	: > High (34)	Management	$1 + 0$	Minor
		Likelihood	: Unlikely (1)			

Table A.1: Risk Evaluations

APPENDIX B

Evaluation of Countermeasures

B.1 Add a Message Authentication Code to each in-vehicular message

A Message Authentication Code (MAC) enables verifiability of message integrity and authenticity. A MAC combines a cryptographic hash function with symmetric encryption. The two communicating parties must have access to a shared secret symmetric key and a common hashing mechanism. On message reception, successfully decrypting the message and verifying the message hash value provides assurance on the message's authenticity and integrity. More details on MACs can be found in [31].

- Countermeasure strategy:Asset hardening
- Advantages
  - Erroneous and malicious message modifications during transmission can be detected whenever a MAC is used, enabling reliable message delivery.
- Disadvantages
  - Message size increases, by addition of the checksum.
  - Messsage processing is delayed because of the integrity and authenticity validations (encryption/decryption and hashing).
  - If no probabilistic encryption [31] is used, message replay is still possible.
- Implications on the ITS architecture
  - For enabling verifiability of message integrity between two communicating parties, both parties must use the same MAC algorithms (which can be implemented in software easily). However, the MAC algorithm implies sharing and storing a secret key in a secure manner.

  - A MAC on itself does not prevent from replay attacks. Hence, probabilistic encryption is required.

- Ability to remove relevant threats
  - Assures integrity and authenticity of messages, hence mitigates the risk of message injection and modification.

B.2 Use a unique identifier for each message

Even though messages with corresponding MAC can be verified on their integrity and authenticity, malicious message replay can not be detected if non-probabilistic encryption is used in the MAC mechanism. Non-non-probabilistic encryption can be achieved by including a message identifier—which will change every message—to each message. Basically, there are three types of message identifiers: *counter*, *nonce* (number used once) and a *time stamp*, each of them have their advantages and disadvantages.

- Countermeasure strategy: Asset hardening
- Advantages
  Nonce: Messages including a nonce (assumingly an unpredictable number), equal to another message nonce received before, will be considered as a message replay and can be rejected. So, within a given context, a nonce may not be reused.. Consequently, communicating ITS (component(s)) need to maintain a list of consumed nonces, and, morover, require a (pseudo-)random number generator ((P)RNG) for generation of the nonces.
  Counter: Each ITS (component(s)) maintain a (synchronised) counter to which a message's nonce may not deviate (up to a certain threshold). Massages including a counter value lower than than the synchronised counter value can be rejected. Consequently, communicating ITS (component(s)) need to synchronise their counter states.
  Time stamp: Messages with a time stamp outside the current time window can be rejected.
- Disadvantages
  Nonce: Ideally, a nonce should be generated with a (true) random bit generator. However, random bit generation is an inefficient procedure in most practical environments [31]. As the ITS Station will have to process (non-)messagesat a high frequency, (true) random number generators are less applicable. Therefore, a *pseudorandom bit generator* (PRBG) will be more applicable. Hardware implementations of a RNG often are relatively slow (limited number of random bits per second), and therefore are often used to "seed" a faster—but less secure—pseudorandom number generator. Moreover, because of the highly dynamic traffic topology, synchronising the list of used nonces among communicating ITS (component(s)) will be impractical. Furthermore, the number of possible nonce values (nonce space) require to be verly large in case of high message frequency, as nonce collisions will occur more often, leading to message discards. Consequently, nonce values will have an increase byte size, affecting the available bandwidth. Moreover, because of the 'Birthday', collisions will occur with higher probability than $1/n$ (with n equal to the number of possible nonce values), hereby requiring a larger nonce space.

Counter: Because of the highly dynamic traffic topology, it will be impractical to synchronise the counter among communicating ITS (component(s)). Furthermore, the counter will be reset (overflow) after a fixed number of incrementations (or decrementations, depending on the implementation). An attacker having a record of one or more messages for every possible sequence number can replay them whenever the current sequence number reaches the corresponding sequence numbers again. Moreover, as safety related (non-)messagesare just broadcast—meaning, the communication protocol is stateless, thus conclusions can only be made on basis of individual message contents—the receiving ITS (component(s)) can not detect the replay by only checking the sequence number. For non-safety related communication—which can use stateful communication protocols such as TCP over IPv6—sequence numbers may be practical, but for safety related `V2V` communication, the sequence number is impractical [7]
Time stamp: Message replay within the same time window as it has been recorded still is possible. Moreover, communicating ITS (component(s)) need to synchronise their clocks, as otherwise valid and genuine (non-)messagesmay be discarded. Clock synchronisation could be achieved by the use of GNSS time. Furthermore, time representations must be non-recurring—e.g., such as POSIX time—otherwise validity time windows will recur as well, leaving open the possibility to message replay.

- Implications on the ITS(S) architecture

  - *Nonce*: A (pseudo)random number generator for nonce generation, secure storage and maintenance of list of consumed nonces are required.

  - *Counter*: Secure storage and maintenance of the synchronised counter (synchronised among all intercommunicating parties) is required. This counter synchronisation introduces additional management communication.

  - *Time stamp*: A global (in-vehicle) clock—securely accessible for all intercommunicating parties—for maintaining a synchronised time window. Provisioning of a global clock may introduce additional management communication.

  - *All methods*: For all methods, the communication protocol and message formats are affected.

  - Access layer and Network layer
    As safety-related messagesare time stamped and signed, these do not require an additional timestamp for communicating ITS messages between the Access Layer and Network Layer. However, non-ITS messages may not include a timestamp, thus require one of the three discussed message identifier methods.

- Ability to remove relevant threats

  - Message injections (e.g., by replay), outside the message identifier validity window, will be detectable.

    - Whenever the message identifiers are used to create probabilistic encryption for communication, eavesdropping on (probabilistically encrypted) messages doe not reveal any information on the message's content.

B.3 Authenticate ITS Station user

ITS Station users must be authenticated to prevent unauthorised users using a stolen ITS Station. For example, to prevent a stolen ITS Station from broadcasting messages. Current car keys could be extended for authentication.

- Countermeasure strategy: Asset hardening
- Advantages
    - An ITS Station will not grant access to a user, not possessing the corresponding vehicle key, e.g., a hijacker of an ITS Station.
- Disadvantages
    - Feature extension of current vehicle keys may be costly.
- Implications on the ITS architecture
    - ITS Station components must include a mechanism to block access to privileged resources.
    - Care must be taken to prevent from replay attacks.
- Ability to remove relevant threats
    - An ITS Station hijacker can not perform abuse masquerading attacks by using that ITS Station, whenever the hijacker does not possess the corresponding ITS Station key.

B.4 Authenticate ITS Station processes

A process may be restricted to certain ITS Station users, ITS Station components, ITS Station software or (non-)ITS messages, hereby requiring authentication and authorisation of those. In order to ensure only privileged ITS Station components or processes gain access to restricted processes, they must be authenticated and authorised by the use of an authentication and authorisation ticket. Digital certificates, including a list of permissions the certificate holder has, could represent such a ticket. Nevertheless, at runtime, verification of digital signatures probably requires too much resources. To overcome this, the first time an ITS Station component or process requests access to restricted resources, its digital certificate and corresponding permissions are verified. After successful validation, a session key could be generated to enable verifiability of the proclaimed identity in further communication, e.g., by the use of a Message Authentication Code, which is much faster than verifying digital signatures.

For example, 3rd-party applications are not allowed to access the Vehicle Control System. Authentication and authorisation is a method to grant access to restricted processes for authorised processes or entities only.

- Countermeasure strategy: Asset hardening
- Advantages
  - ITS Station processes will be able to verify authorisation of ITS Station components and processes.
- Disadvantages
  - Messages require additional information, such as an authorisation ticket (e.g., digital signature or MAC), hereby increasing the required communication bandwidth or storage capacity. Whenever authorisation tickets are included in a digital certificate, signed by a TTP, changes in permissions require certificates to be revoked and regenerated with updated permissions. Moreover, restricted ITS Station processes require capabilities to verify digital signatures.
- Implications on the ITS architecture
  - ITS Station manufacturers must implement authentication and authorisation mechanisms to prevent unauthorised components or processes from accessing restricted ITS Station processes.
  - In case permissions are included in digital certificates, signature verification mechanisms must be implemented as well.
- Ability to remove relevant threats
  - Counterfeit ITS components (or malicious processes), not possessing an authentic authorisation ticket, will not be able to access restricted ITS Station processes.
  - Processes have a limited set of privileges.

B.5 Authenticate ITS Station hardware

To enable detectability of unauthentic hardware, a digital certificate could be attached to ITS Station components.

- Countermeasure strategy: Asset hardening
- Advantages
  - Enables detectability of unauthentic hardware components.
- Disadvantages
  - See countermeasure *Digitally sign data*

- Implications on the ITS architecture
  - See countermeasure *Digitally sign data*
- Ability to remove relevant threats
  - Counterfeit ITS components (or malicious processes) can be detected.

B.6 Compartmentalise software executions

Protocol stack layers must not be able to influence of monitor other protocol stack layers. Hence, they must be executed compartmentalised. When a protocol stack layer "breaks out" of its compartment, it may gain permissions on CPU firmware "level", thus the ability to affect other compartmentalised protocol stack layers. Hence, the choice of compartmentalisation mechanism is dependant for the actual security is provides. The following non-exhaustive list indicates the compartmentalisation mechanisms that are considered in this analysis. The advantages and disadvantages are discussed briefly for each mechanism.

- Hardware separation. Separating protocol stack layer(s) by implementing those on different physical units, e.g., CPUs.
  - *Advantage.* Protocol stack layers can only influence other protocol stack layers—implemented on the other physical unit—via the communication interface between the units.
  - *Disadvantage.* Communication between the physical units may require integrity and authenticity assurance. This may imply that the physical units have to establish and securely store a shared key.
- Trusted Execution Environment. CPUs that offer Trusted Execution Environments [2] (TEE), can execute software either in a "normal world" or in a "secure world". The "secure world" provides secure execution of software and secure storage, whereas the "normal world" typically executes (third-party) applications. For example, ARM TrustZone® supports TEE.
  - *Advantage.* Software, executed in the "normal world ", can only interact with software that is executed in the "secure world" via strict and highly secure TEE kernel.
  - *Disadvantage.* Due to the complexity of CPUs with TEE, development of software for this CPU is more difficult, hence more subject to faulty (insecure) implementation. Furthermore, TEE creates "only" two compartments: the "secure world" and the "normal world".
- Virtualisation. Virtualised software is executed in an isolated user-space, which is executed "on top" of the kernel of the operating system. Virtualised software can only communicate via the kernel.
  - *Advantage.* Does not require a (probably more expensive) TEE-enabled CPU. Furthermore, in general, an unlimited number of virtualised compartments can be created. Hence, virtual compartmentalisation allow more flexible software design than TEE for example.

  - *Disadvantage*. Virtual compartmentalisation is less secure than TEE and hardware separation. Hence, the risk of virtualised software "breaking out" of its compartment is higher than with the other two methods.

- Countermeasure strategy: Asset hardening
- Advantages
  - Software executions are separated from other software executions. The actual security of the compartmentalisation is dependent on the used compartmentalisation method.
- Disadvantages
  - May require additional hardware components or more expensive hardware components.
  - May make software development more difficult.
  - May have a negative impact on the computational resources.
- Implications on the ITS architecture
  - May require additional or more expensive hardware components.
- Ability to remove relevant threats
  - Compartmentalised software executions can not maliciously affect other software executions.

B.7 Comply with NIST firmware bootstrap and update guidelines

Comply to NIST guidelines for securing firmware (see [20]). That is:

a) The firmware must be protected from malicious modification.

b) Firmware updates must be signed.

c) Firmware protection cannot be bypassed.

d) A user must be present for all firmware updates.

e) There must be anti-rollback protection.

- Countermeasure strategy: Asset hardening
- Advantages
  - Firmware integrity and authenticity is protected before installation and after.
- Disadvantages
  - See countermeasure *Digitally sign data*

- Implications on the ITS architecture
  - See countermeasure *Digitally sign data*
- Ability to remove relevant threats
  - Unauthentic software can not be installed and (authentic) firmware can not be modified maliciously.

B.8 Digitally sign data

Digitally signing data enables verifiability of integrity and authenticity of the signed data. In general, all data can be digitally signed, but, for the purpose of this thesis, only messages and software (i.e., firmware and applications) are considered.

B.8.1 Digitally sign messages

Digital message signature enable verifiability of message integrity and authenticity. ITS messages will be digitally signed, as agreed on by the Car-2-Car Communications Consortium (C2C-CC). However, in-vehicular messages (non-ITS messages or ITS messages that have been verified) require verifiability of its integrity and authenticity as well.

Nevertheless, as processing messages is subject to stringent timing or computational requirements, digital signatures must not be applied excessively. Therefore, for in-vehicular communication, it is recommended to enabling verifiability of message integrity and authenticity by the use of, for example, a MAC.

- Countermeasure strategy Asset hardening
- Advantages
  - Message integrity and authenticity are verifiable on reception, if corresponding certificate chain—up to a trusted level—is available.
  - Enables detectability of counterfeit ITS Station component(s), if intercommunicating ITS Station components possess a digital certificate, signed by a TTP. For simplification of certificate management, ITS Station components from the same manufacturer could share the same certificate (a “group” certificate). Nevertheless, this will decrease the security in the sense that data origin authenticity can not be verified anymore and, moreover, whenever a group certificate is compromised, all corresponding ITS Station components require a new (common) certificate.
- Disadvantages
  - Every ITS Station component requires a unique (or “group”) certificate, signed by a TTP, common to all intercommunicating ITS Station components. Although the defined PKI for message signatures could be extended for the purpose of communication internal to in-vehicular communication, the TTP still has to sign certificates for ITS Station components. This may have a severe impact on the complexity of the certificate management.

    - To ensure secrecy of the secret key, corresponding a certificate, each certificate holder requires secure—and, in the case of personal (i.e., non-“group”) certificate, exclusive—access to this key. Consequently, a tamper-resistant memory unit is required.
    - Certificates of misbehaving or compromised ITS Station components should be revocable, hereby requiring a so-called Certificate Revocation List (CRL), generated, maintained and distributed by a TTP. Although a revoked certificate can not be trusted anymore, signatures generated with revoked certificates may still validate at ITS Station components with an outdated CRL. CRL maintenance thus is very important.

- Implications on the ITS architecture
    - A TTP, common to all intercommunicating ITS Station components, must sign corresponding certificates and generate, maintain and distribute a Certificate Revocation List (CRL).
    - Every ITS Station component requires secure access to its certificate and corresponding secret key, as well as to the certificate of the TTP. This will require a tamper-resistant is required.
    - Every (intercommunicating) ITS Station component must be capable of generating and verifying digital signatures. If such capabilities are shared, e.g., by the Security Layer sharing its resources, secure separated access sessions must be established.

- Ability to remove relevant threats
    - Maliciously injected or modified messages (by MITM equipment) will not go undetected. Note, malicious firmware could still inject or modify messages (whenever it is capable of generating valid digital signatures).
    - If ITS Station components will contain unique digital certificates, counterfeit ITS Station components will be detected.

B.8.2 Digitally sign software

On contrary to digital signing messages, digitally signing software is less bounded by stringent timing or computational requirements. Software’s integrity and authenticity can be verified after being downloaded and during software bootstrapping.

- Countermeasure strategy
    - Asset hardening

- Advantages
    - Software integrity and authenticity are verifiable on reception and before execution, if corresponding (part of) certificate chain is available.
    - Enables detectability of malicious software injection or modifications before execution.

- Disadvantages
    - A TTP, generating, maintaining and distributing certificates and Certificate Revocation Lists is required.

  - Manufacturers may need to generate and maintain certificates—comprising more privileges—for maintenance of the ITS Station.
  - ITS Station components that will verify software's integrity and authenticity will require (additional) secure resources such as secure memory for key and certificate storage. Furthermore, they will require mechanisms to verify digital signatures.
  - Software will be appended with a digital signature, hereby increasing the required bandwidth (for when it is downloaded) and memory (for installation).
  - On-the-fly verification of over-the-air downloaded software may affect safety-related ITS Station processes.

- Implications on the ITS architecture

  - A TTP, common to all intercommunicating ITS Station components, must sign corresponding certificates and generate, maintain and distribute a Certificate Revocation List (CRL).
  - Every ITS Station component requires secure access to its certificate and corresponding secret key, as well as to the certificate chain.
  - Every ITS Station component, verifying software integrity and authenticity, must have access to a digital signature verification mechanism. If such mechanisms are not available to one or more ITS Station components, available mechanisms must be shared through secure session. This may problematic.

- Ability to remove relevant threats

  - Maliciously modified or injected software (over-the-air or in-vehicle) will be detectable.

B.9 Encrypt data

Encryption can be used to hide confidential information—while being transmitted or stored—from readers, unauthorised to read that information. However, symmetric and asymmetric encryption implementations have different implications on the ITS architecture. Therefore, both types of encryption are analysed on their implications on the ITS architecture. Basically, asymmetric encryption is faster but requires the communicating parties to establish a shared key (e.g., at manufacturer or at runtime by using asymmetric encryption). Symmetric encryption is much slower than symmetric encryption, but does not require to have the communicating parties knowing a shared secret at runtime. Mostly, asymmetric encryption is used for establishing a shared secret key in the initialisation phase of the communication and symmetric is used to encrypt the rest of the communication.

B.9.1 Symmetric encryption

Symmetric encryption requires a secret shared session key, common to all intercommunication parties, authorised to read the confidential information. Establishment of the shared secret key can be done by pre-sharing (e.g., at manufacturer) or by the use of a key-establishment protocol. However, the latter requires asymmetric encryption. Key establishment over a secured channel may be impractical, as ITS Station components may originate from different manufacturers. Moreover, every additional party, sharing the same secret key, increases the risk of leaking that key.

- Countermeasure strategy: Asset hardening
- Advantages
  - Symmetric encryption prevents from eavesdropping on confidential information, if implemented correctly. Note that, this advantage applies to asymmetric encryption as well.
  - Requires less computational power than asymmetric encryption (in general), thus more applicable for environments with stringent timing or computational requirements.
- Disadvantages
  - Pre-sharing secret session keys may be impractible because different ITS Station components may have different manufacturers.
  - Symmetric encryption does not (necessarily) enable verifiability of message integrity, nor verifiability of message authenticity. Note that, this disadvantage applies to asymmetric encryption as well.
  - Although significantly less than with asymmetric encryption, symmetric encryption affects the performance of the ITS Station.
  - Symmetric encryption does not enable non-repudiation, as the message could have originated from any party, having access to the shared symmetric key.
- Implications on the ITS architecture
  - All ITS layers (for ITS Station components that use symmetric encryption)
    - Secure storage for the secret session key(s). Note that this implications applies to asymmetric encryption as well.
    - The secret session key must be shared over a secured channel—preserving confiden, other than channel which is used for the encrypted communication. This implies having a secure environment in which session keys are shared, possibly residing at the manufacturer, or elsewhere.
    - Establishment of shared keys by the use of hybrid solution (using asymmetric encryption for session key establishment and symmetric encryption for following communication) introduces a delay in the initialisation phase (i.e., bootstrapping) of an ITS Station.
  - Network and Access layer
- Ability to remove relevant threats
  - En route message injection and modification (when integrity and authenticity preservering encryption is used)
  - En route message eavesdrop will be infeasible, whenever all security aspects are correctly taken into account.
  - To, create probabilistic encryption, one could include a message identifier in the symmetrically encrypted data.

B.9.2 Asymmetric encryption

Asymmetric encryption is usually used establishment of the shared key, as it does not require a secured channel (different to the one used for encrypted communication). The main difference with symmetric encryption is point in time where trust is involved; with symmetric encryption it lies in the trusted environment during establishment of the shared key, with asymmetric it lies in the trusted environment and the TTP during at the time of digitally signing the digital certificate.

- Countermeasure strategy: Asset hardening
- Advantages
  - Symmetric encryption prevents from eavesdropping on confidential information, if implemented correctly. Note that, this advantage applies to asymmetric encryption as well.
  - Intercommunicating parties can establish a shared symmetric key "on-the-fly", without having "met" in a secure environment to establish a secretly shared session key, as would be necessary with symmetric encryption. Note that, with a hybrid solution, "on-the-fly" key-establishment is possible as well.
  - Asymmetric encryption does not require communicating parties to have "met" before for establishing a shared session key, as would be the case with symmetric encryption. Nevertheless, with asymmetric encryption, one would require a TTP.
  - If the secret key, used for asymmetric encryption, of one communicating party is compromised, communication intended for that compromised component is compromised only.
- Disadvantages
- Symmetric encryption does not (necessarily) enable verifiability of message integrity, nor verifiability of message authenticity. Note that, this disadvantage applies to asymmetric encryption as well.
  - Requires less computational power than asymmetric encryption (in general), thus more applicable for environments with stringent timing or computational requirements.
  - Asymmetric encryption requires more computational power than symmetric encryption.
  - Asymmetric encryption requires knowledge of the public key of the intercommunicating ITS Station components.
  - Group communication can only be established when all members share the same public key pair (thus, the corresponding private key as well), or when they have established a shared symmetric key (e.g., by using a key-agreement protocol).
  - Masquerading attacks are still possible if the integrity and authenticity of a public key can not be verified. A TTP, digitally signing the public key, can provide verifiability of these. Note that this implies the intercommunicating parties being capable of generating and verifying digital signatures.
  - Encryption affects the performance of the ITS Station (more that symmetric encryption).

- Implications on the ITS architecture
  - Requires secure secret key storage for all intercommunicating ITS Station components (which use asymmetric encryption).
  - Requires all intercommunicating parties being capable of generating and verifying digital signatures.
  - Requires larger memory capacity for storage (compared to symmetric encryption) of personal public key pair, corresponding certificate, TTP's certificate and public keys and corresponding certificate of other parties.
  - Asymmetric encryption requires larger public and private keys for establishing the same security level as symmetric encryption. For example, to obtain a 256 bit security level, a 256 bit (symmetric) AES key is required or a asymmetric 15360 bit RSA key or 512 bit asymmetric ECC key is required.
  - Messages could become larger, depending on the used encryption method and encryption parameters. For example, RSA encrypted message will have the size of (at most) the size of the used *modulus*
  - Establishment of shared keys, by the use of a key-agreement protocol, introduces a delay of the initialisation phase (i.e., bootstrapping) of an ITS Station
- Ability to remove relevant threats
  - En route message injection and modification (when integrity and authenticity preserverving encryption is used)
  - En route message eavesdrop
  - To, create probabilistic encryption, one could include a message identifier in the symmetrically encrypted data.

B.10 Include a list of allowed applications in each ITS Station's certificate.

Some applications are restricted to a certain set of ITS Stations. For example, emergency vehicles contain applications that are authorised to send certain emergency messages. Non-emergency vehicles may not have such privileges. In case such applications can be copied or stolen, traffic safety and efficiency may be affected. To mitigate this problem, each ITS Station certificate should contain a list of authorised applications (and corresponding privileges). A malicious application that (unauthorised) initiates the broadcast of certain restricted messages, can be detected the Network Layer by checking the privileges of the messages with the digital certificate.

- Countermeasure strategy
  Asset hardening
- Advantages
  - Incorrect or malicious applications can not lead to a broadcast of messages that the ITS Station is not allowed to.

- Disadvantages
  - Whenever the set of applications and corresponding permissions change, the digital certificates need to be revoked or updated.
- Implications on the ITS architecture
  - Only additional check, in software, at the Network Layer.
- Ability to remove relevant threats
  - Incorrect or malicious applications can not lead to a broadcast of messages that the ITS Station is not allowed to.

B.11 Managing access of applications which require privileges outside their "container" by Management Layer

Applications that share resources, e.g., external memory, must not be able to affect other applications' data. The Management Layer should manage access to the shared resources.

- Countermeasure strategy
  Asset hardening
- Advantages
  - Incorrect or malicious application can not affect data of other applications which is stored in shared memory.
- Disadvantages
  - May affect application performances as the Management Layer may be involved in their processes.
- Implications on the ITS architecture
  - Additional functionalities for the Management Layer are required, possibly affecting current ETSI standards.
- Ability to remove relevant threats
  - Incorrect or malicious application can not affect data of other applications which is stored in shared memory.

B.12 Perform plausibility checks on received messages

In order to detect spuriously generated messages, the message's content can be verified on its trustworthiness. For example, the Facility Layer could verify the trustworthiness of ITS message contents, but also the trustworthiness of traffic safety-related notifications from traffic safety applications. Which algorithms should be used is out of scope of this document.

- Countermeasure strategy
  Asset hardening
- Advantages
  - Plausibility of received messages can be verified on basis on the message contents. Hence, spurious messages may be detected.
- Disadvantages
  - Plausibility checks may not detect all spurious messages.
  - Plausibility checks affects the performance of the ITS Station.
- Implications on the ITS architecture
  - Perhaps, a faster CPU with more (internal/external) memory is required to implement the plausibility checks.
- Ability to remove relevant threats
  - The percentage of detectable spurious messages is dependent on the plausibility check algorithm. But, clearly, not all spurious messages will be detected.

B.13 Perform a Built-In Self-Test (BIST) during boot time and during run time of the ITS Station

A so-called Built-In Self-Test (BIST) could detect malfunctioning ITS Station components. However, this BIST may not detect all maliciously modifications to hardware components. The BIST should be performed on boot time, but also during run time.

- Countermeasure strategy: Asset hardening
- Advantages
  - Malfunctioning ITS Station components will be detectable.
- Disadvantages
  - Malicious modifications to ITS Station components may still go undetected.
- Implications on the ITS architecture
  - May delay boot time of the ITS Station.

- Ability to remove relevant threats
  - Malfunctioning hardware components will be detectable, but maliciously modified components may go undetected.

B.14 Use a secure element

A Trusted Platform Module (TPM), is capable of securely generating and storing cryptographic keys, as well as storing data in an encrypted form. Furthermore, since each TPM has a unique and secret RSA key burned into it at production time, it is capable of being authenticated as well. This TPM could be used to securely store certificates (personal and of TTPs) (as required by digital signature implementation, cryptographic keys for encryption and for secure storage of MAC keys and generation of message identifiers.

- Countermeasure strategy: Asset hardening
- Advantages
  - Intercommunicating ITS Stations, having access to a personal or (group) shared TPM, will be able to implement message identifiers (e.g., nonce generation and MAC), digital signatures processing and encryption/decryption, as well as secure storage of other secret information. All advantages, implied by these implementations, apply here as well (when implemented).
- Disadvantages
  - A TPM must be accessible over a secure channel, otherwise, an attacker could perform a MITM attack between the ITS Station component and the corresponding TPM. This may result is the requirement of having tamper-resistant hardware package.
  - TPM introduces a delay in processing speed, as before access to restricted ITS Station components or processes is granted, authentication and authorisation of the access requesting component or process has to be verified.
  - TPM are relatively expensive, i.e., has a financial impact on the cost of an ITS Station (component)
- Implications on the ITS architecture
  - Communication, among ITS Station components, require additional security information, hereby consuming more communication bandwidth.
  - ITS Station component manufacturers must implement authentication and authorisation mechanisms to prevent unauthorised user from gaining access to restricted ITS Station processes.
- Ability to remove relevant threats
  - An ITS Station hijacker can not perform masquerading attacks by using that ITS Station, whenever the hijacker does not possess the corresponding ITS Station authentication and authorisation ticker.

  - Unauthorised access to the ITS Station processes is infeasible without possession of a valid authentication and authorisation ticker.

B.15 Use hardware that is resistant to side-channel attacks

Covert channels may enable influencing and monitoring or protocol stack layers. Resistance to side-channel attacks is in particular important for protocol stack layers that store and use confidential information, such as, for example, the Security Layer when digitally signing ITS messages.

- Countermeasure strategy: Asset hardening
- Advantages
  - Side-channel attacks do not affect processes and confidential information does not leak.
- Disadvantages
  - High development costs.
- Implications on the ITS architecture
  - Typically, a secure element is considered resistant to side-channel attacks.
- Ability to remove relevant threats
  - Protocol stack firmware can not be influenced by side-channel attacks and confidential information does not leak via side-channels.

B.16 Use tamper-resistant hardware

Tampering is the abnormal (malicious) use of ITS Station components. Tampering may enable processes and data influencing and monitoring.

- Countermeasure strategy: Asset hardening
- Advantages
  - Hardware tampering does not affect processes and confidential information does not leak.
- Disadvantages
  - High development and tamper-resistance certification costs.
- Implications on the ITS architecture
  - Typically, a secure element is considered tamper-resistant.
- Ability to remove relevant threats
  - Protocol stack firmware can not be influenced and confidential information does not leak by tampering.

B.17 Use tamper-evident hardware

Tamper-evident hardware contains immutable evidence of a tampering event. Hence, this tamper-evidence may be used as input to the Buil-In Self-Test (see B.13.

- Countermeasure strategy: Asset hardening
- Advantages
  - Hardware tampering will not go undetected.
- Disadvantages
  - High development costs.
- Implications on the ITS architecture
  - Requires expensive tamper-evident hardware.
- Ability to remove relevant threats
  - Hardware tampering will not go undetected.