PREparing SEcuRe VEhicle-to-X Communication Systems

Deliverable 5.1

Deployment Issues Report V1

Project:	**PRESERVE**
Project Number:	**IST-269994**
Deliverable:	**D5.1**
Title:	**Deployment Issues Report V1**
Version:	**1.1**
Confidentiality:	**Public**
Editor:	**Panos Papadimitratos**
Cont. Authors:	**C. Jouvray, F. Kargl, J. Petit, P. Papadimitratos, M. Sall**
Date:	**2012-08-30**

Part of the Seventh Framework Program
Funded by the EC-DG INFSO

Document History

Version	Date	Main author	Summary of changes
v0.1	2011-11-04	P. Papadimitratos (KTH)	Initial version
v0.2	2011-12-21	J. Petit (UTwente)	Writing section Pseudonymous Authentication
v0.3	2012-01-15	C. Jouvray (Trialog), M. Sall (Trialog)	Writing section Life-cycle and operation issues
v0.5	2012-01-20	P. Papadimitratos (KTH)	Writing sections 5 and 6
v0.7	2012-01-20	P. Papadimitratos (KTH)	Writing introduction, integrating technical reports (Sections 1, 3, 7)
v0.8	2012-01-20	P. Papadimitratos (KTH)	Editing
v0.9	2012-02-06	J. Petit (UT)	Writing section Other research results
v1.0	2012-02-10	P. Papadimitratos (KTH)	Intra-project review input integration
v1.1	2012-08-10	P. Papadimitratos (KTH), N. Bissmeyer (SIT), R. Moalla (Renault), N. Alexiou (KTH)	Internal revision and revision based on external reviews. External reviewers: B. Glass, H. Stuebing, A Weimerskirch, and W. White

Approval		
	Name	Date
Prepared	Panos Papadimitratos	2012-02-06
Reviewed	All Project Partners	2012-08-10
Authorized	Frank Kargl	2012-08-30

Circulation	
Recipient	Date of submission
Project Partners	2012-09-06
European Commission	2012-09-06

Contents

List of Figures

List of Tables

1 Glossary

Abbrev	Synonyms	Description	Details
API		Application Programming Interface	An API is a particular set of specifications that software programs can follow to communicate with each other.
AU		Application Unit	Hardware unit in an ITS station running the ITS applications
ASN.1		Abstract Syntax Notation One	ASN.1 is a standard and flexible notation that describes data structures for representing, encoding, transmitting, and decoding data.
CA		Certificate Authority	A CA is an entity that issues digital certificates.
CAM		Cooperative Awareness Message	CAMs are sent by vehicles multiple times a second (typically up to 10 Hz), they are broadcasted unencrypted over a single-hop and thus receivable by any receiver within range. They contain the vehicle's current position and speed, along with information such as steering wheel orientation, brake state, and vehicle length and width.
CAN		Controller Area Network	A CAN is a vehicle bus standard designed to allow microcontrollers and on-board devices to communicate with each other.
CCM		Communication Control Module	Module responsible for protecting on-board communication. Originates from the EVITA project.
CCU		Communication & Control Unit	Hardware unit in an ITS station running the communication stack
CE		Consumer Electronics	Electronic devices like smartphone or MP3 player of the vehicle driver or a passenger

Abbrev	Synonyms	Description	Details
CL		Convergence Layer	Module that connects the external on-board entities (e.g. communication stack or applications) to the PRESERVE Vehicle Security Subsystem (VSS)
CPU		Central Processing Unit	
CRC		Cyclic Redundancy Code	Is used to produce a checksum in order to detect errors in data storage or transmission.
CRS		Cryptographic Services	Module acting as proxy for accessing different cryptographic algorithm implementations. Originates from the EVITA project
DoS		Denial of Service	A DoS is a form of attack on a computer system or networks.
DENM	DNM	Decentralized Environmental Notification Message	A DENM transmission is triggered by a cooperative road hazard warning application, providing information to other ITS stations about a specific driving environment event or traffic event. The ITS station that receives the DENM is able to provide appropriate HMI information to the end user, who makes use of these information or takes actions in its driving and traveling. Fehler: Referenz nicht gefunden
EAM		Entity Authentication Module	Module responsible for ensuring entity authentication of in-vehicle components. Originates from the EVITA project
ECC		Elliptic Curve Cryptography	ECC is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields.
ECU		Electronic Control Unit	
ECR		ECU configuration register	Register used for secure boot and authenticated boot inside the HSM (similar to platform configuration register inside a TPM)
FOT		Field Operational Test	

Abbrev	Synonyms	Description	Details
G5A		ITS road safety communication (802.11p)	Frequency band between 5.875 GHz and 5.905 GHz - reserved for ITS road safety communication
G5B		ITS non-safety communication (802.11p)	Frequency band between 5.855 GHz and 5.875 GHz - reserved for ITS road non-safety communication
G5C	C-WLAN	5GHz WLAN communication (802.11a)	
GNSS	GPS	Global Navigation Satellite System	Generic term for an Global navigation satellite system (GPS, GLONAS, Galileo)
HMI		Human-Machine Interface	
HSM		Hardware Security Module	
HU		Head-Unit	
I2V	I2C	Infrastructure-to-Vehicle	Communication between infrastructure components like roadside units and vehicles
I2I		Infrastructure-to-Infrastructure	Communication between multiple infrastructure components like roadside units
ICS		ITS Central Station	ITS station in a central ITS subsystem
ILP		Inter Layer Proxy	Component introduced by the SeVeCom project, that captures and allows modification of messages between different layers of a communication stack
IDK	Module Authentication Key	Device Identity Key	The Device Identity Key is introduced by EVITA and is used for HSM identification. The IDK can also be certified by a manufacturer authentication key.
IMT	GSM, GPRS, UMTS	Public cellular services (2G, 3G, ...)	
IPR		Intellectual Property Right	

Abbrev	Synonyms	Description	Details
ITS		Intelligent Transportation Systems	Intelligent Transport Systems (ITS) are systems to support transportation of goods and humans with information and communication technologies in order to efficiently and safely use the transport infrastructure and transport means (cars, trains, planes, ships).
ITS-S		ITS Station	Generic term for any ITS station like vehicle station, roadside unit, ...
IDM		ID & Trust Management Module	Module responsible for ID management originating from SeVeCom project.
IVC	ITSC, ITS Communications	Inter-Vehicle Communication	Combination of V2V and V2I
IVS	OBU	ITS Vehicle Station	The term "vehicle" can also be used within PRESERVE
LDM	Environment Table	Local Dynamic Map	Local geo-referenced database containing a V2X-relevant image of the real world
LTC		Long-Term Certificate	PRESERVE realization of an ETSI Enrolment Credential. The long-term certificate authenticates a stations within the PKI, e.g., for PC refill and may contain identification data and properties.
LTCA		Long-Term Certificate Authority	PRESERVE realization of an ETSI Enrollment Credential Authority that is part of the PKI and responsible for issuing long-term certificates.
MAC		Media Access Control	The MAC data communication protocol sub-layer is a sublayer of the Data Link Layer specified in the seven-layer OSI model.
OBD		On-Board Diagnosis	OBD is a generic term referring to a vehicle's self-diagnostic and reporting capability that can be used by a repair technician to access the vehicles sub-systems.

Abbrev	Synonyms	Description	Details
OEM		Original Equipment Manufacturer	Refers to an generic car manufacturer
OBU	IVS	On-Board Unit	An OBU is part of the V2X communication system at an ITS station. In different implementations different devices are used (e.g. CCU and AU)
PAP		Policy Administration Point	Module related to the PDM originating from EVITA project
PC	Short Term Certificate	Pseudonym Certificate	A short term certificate authenticates stations in G5A communication and contains data reduced to a minimum.
PCA		Pseudonym Certificate Authority	Certificate authority entity in the PKI that issues pseudonym certificates
PDM		Policy Decision Module	Module responsible for enforcing the use of policies originating from EVITA project
PDP		Policy Decision Point	Module related to the Policy Decision Module originating from EVITA project
PeRA		Privacy-enforcing Runtime Architecture	Module responsible for enforcing privacy protection policies originating from PRECIOSA project
PEP		Policy Enforcement Point	Module related to the Policy Decision Module originating from EVITA project
PIM		Platform Integrity Module	Module responsible for ensuring in-vehicle component integrity originating from EVITA project
PKI		Public Key Infrastructure	A PKI is a set of hardware, software, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.
PMM		Pseudonym Management Module	Module responsible for management of the station's pseudonym certificates originating from SeVeCom project
RSU	IRS, ITS Roadside Station	Roadside Unit	A RSU is a stationary or mobile ITS station at the roadside acting as access point to the infrastructure.

Abbrev	Synonyms	Description	Details
SAP		Service Access Point	Informative functional specification that enables the interconnection of different component implementations.
SM		Security Manager	Module responsible for securing the V2X communication with external ITS stations originating from SeVeCom project
SCM		Secure Communication Module	A generic name for the complete secure communication stack
SEP		Security Event Processor	Module responsible for security event management (e.g. checking message plausibility, station reputation calculation)
TPM		Trusted Platform Module	A TPM is both, the name of a published specification detailing a secure crypto-processor that can store cryptographic keys, as well as the general name of implementations of that specification, often called the "TPM chip" or "TPM Security Device".
UML		Unified Modeling Language	UML is an object modeling and specification language used in software engineering.
UTC		Coordinated Universal Time	UTC is the primary time standard by which the world regulates clocks and time.
V2I	C2I	Vehicle-to-Infrastructure	Direct vehicle to roadside infrastructure communication using a wireless local area network
V2V	C2C	Vehicle-to-Vehicle	Direct vehicle(s) to vehicle(s) communication using a wireless local area network
V2X	C2X	Vehicle-to-Vehicle (V2V) and/or Vehicle-to-Infrastructure (V2I)	Direct vehicle(s) to vehicle(s) or vehicle(s) to infrastructure communication using a wireless local area network
VIN		Vehicle Identification Number	Unique serial number of a vehicle
VSA		Vehicle Security Architecture	General outcome of PRESERVE work package 1

Abbrev	Synonyms	Description	Details
VSS		V2X Security Subsystem	Close-to-market implementation of the PRESERVE VSA that is the outcome of PRESERVE work package 2
WLAN		Wireless Local Area Network	
XML		Extensible Markup Language	XML is a set of rules for encoding documents in machine-readable form.

2 Introduction

The Work Package 5 (WP5) is concerned with security and privacy related issues, notably those that pertain to key matters for later deployment and standardization of secure and privacy protecting V2X communication systems.

The focus of the first version of this deliverable is on privacy enhancing technologies and issues during operation. The first part of the document (Sec. 3 - Sec. 7) provides a survey of pseudonymous authentication schemes, and discusses in further depth specific protocols that on the one hand provide security for V2X and on the other hand enhance privacy. Moreover, a brief presentation of other related publications that appeared during the first year of the project. The second part (Sec. 8) elaborates a number of use cases, to illustrate operational issues, notably relating with the life-cycle management of secure V2X.

The two parts of the document correspond to the subtasks 5230 and 5120 respectively. A great deal of related current state of the art, published prior to PRESERVE, notably by PRESERVE partners as parts or follow-ups of precursor projects, is not included here in detail. Rather, the survey covers that work. On the other hand, the material in the WP2 deliverables of PRESERVE reflect the evolved understanding and provide a solid basis for protocols for deployment and maintenance.

The focus in this report is on forward looking issues, beyond the developing PRESERVE architecture and security subsystem. In addition, the WP5 reports provide a track record of all related research output. Accordingly, the V2X Security Subsystem (VSS) does not integrate all schemes presented in this deliverable: the details regarding VSS, notably its first version, are available in deliverables of WP2 and WP4, and the field trial related material in deliverables of WP3. It is expected that the second and third versions of the VSS will integrate some schemes and elements that are results of the ongoing WP5 work.

In the rest of this document, a number of aspects and technical approaches are discussed. First, a discussion on privacy and level of protection including non-technical considerations is given. Then, pseudonymous authentication schemes, as one broadly investigated method to provide security and enhance privacy, are surveyed. A more detailed exposition of secure and privacy enhancing schemes is given next, setting the stage for ongoing and upcoming investigations. The first part of the deliverable continues with a concise discussion of other research results obtained during the first year of the project, relating to various security aspects. It concludes with a discussion around the appropriate placement of security within the communication/networking protocol stack. The second part of the deliverable, regarding life-cycle and operation issues, unfolds a set of use cases and relates them to security matters.

3 V2X Privacy Protection Broader Considerations

There is still an on-going discussion about the need and extent of required privacy protection in cooperative Intelligent Transportation Systems that are based on DSRC-type V2X communication. In these systems, periodic and sporadic broadcast messages like CAM and DENM are sent that include position information about vehicles. During the FP6 and FP7 research projects SeVeCom, PRECIOSA, and PRESERVE, the authors of this paper dealt with the question whether this is relevant with respect to data protection and privacy laws. With this discussion, we want to share insights and conclusions in a condensed form to make it accessible for discussion in on-going harmonization and standardization.

3.1 Personal Data Processing

Do cooperative ITS process Personal Data? Both standards by ETSI and IEEE foresee that periodic broadcast messages sent the position of the vehicle together with a unique identifier (MAC address and ITS station-identifier) and a cryptographic signature and certificate to all neighboring stations in reception range (or if re-broadcasted to an even larger set of recipients). Pseudonym certificates are issued by a trusted authority, the so-called pseudonym provider.

One naive position on privacy could be that the involved information does not constitute personal information, as there is no link to a person but only to a vehicle. Vehicles might be driven by many different drivers and therefore knowing the location of a vehicle does not reveal the position of an individual. Furthermore, the MAC address or ITS station-identifier is only an indirect reference from which no vehicle (and thus also no specific driver) can be identified. Whether the cryptographic certificate would include a direct identifier for a vehicle (i.e., a VID or license-plate number) is debatable and technically not required.

In this case, data protection and privacy laws like the European Data Protection Directive 95/46/EC [2] would not apply and no further actions to protect privacy would be required. However, this is a too simplistic view. First of all, it is known that vehicles are on average only driven by a small number of drivers and therefore knowing the position of a vehicle reveals the whereabouts of drivers. As an example, [3] indicates that a typical German household owns one car that is driven by two to three persons. Furthermore, as shown, e.g., in [4], knowing the itineraries of a vehicle one can deduce the owners homeplace with a strong correlation.

We constitute that in our opinion location information communicated in V2X messages has a sufficiently clear link to individuals and therefore constitutes personal information and consequently data protection laws apply.

This is in line with the position of the European Data Protection Supervisor, Peter Hustinx, who constitutes in his opinion on the European Commission's ITS Directive [5]:

> Some of the information that will be processed through ITS is aggregated — such as on traffic, accidents, and opportunities — and does not relate to any individual, while other information is related to identified or identifiable individuals and therefore qualifies as personal data within the meaning of Article 2(a) of Directive 95/46/EC.

Note that legal position in the U.S. might be different, as indicated, e.g., in the Supreme court ruling "US v. Knotts" [6] that stated that

> A person traveling in an automobile on public thoroughfares has no reasonable expectation of privacy in his movements.

3.2 Discussion on Pseudonyms as the Solution

Research projects like SeVeCom and PRECIOSA proposed the use of changing pseudonyms for providing privacy protection for position information in V2X messages [7, 8]. The basic idea is that security credentials (pseudonym certificates) and identifiers (MAC addresses, ITS station identifiers) used, do not relate directly to a vehicle or a person.

Given the nature of the credentials and identifiers in question, this is trivially the case. However, as indicated, e.g., by [4], this is not sufficient. If a vehicle would use a permanent pseudonymous identifier, an attacker could still track the itineraries of a vehicle and knowing where a vehicle typically parks at night and travels to at day it becomes almost trivial to conclude which person it belongs to. Therefore, the mentioned pseudonym concepts foresee that a vehicle is equipped with a set of changeable pseudonyms and regularly modifies all identifiers it uses [7, 8]. That way reconstructing itineraries and linking to specific persons becomes much harder.

In essence, we are dealing here with data that can only be linked back to specific persons with a certain probability and the privacy protection measures taken aim at reducing this probability down to an acceptable level. It is not completely clear how legal frameworks deal with this notion of data that is linked statistically to a certain person. A straightforward question would be what probability would be acceptable to declare data as non-personal or anonymized data and that would make data protection laws inapplicable.

It also needs to be noted that some research [9] indicates that a strong attacker that has a complete coverage of an area and captures all packets that are sent is able to effectively track vehicles. Other work indicates similar results [4, 10, 11].

There is also work that suggests additions, e.g., for protecting pseudonym changes in cryptographical mix-zones [12]. However, those approaches often raise further questions of practicability and effectiveness. We therefore stick to the discussion of the basic scheme as also proposed by the C2C-CC [13].

Another important issue is tracking via information contained as message content or in certificates that might allow correlation of packets over pseudonym changes. An example are vehicle dimensions that are included in CAM messages. Even when changing a pseudonym, having exact vehicle dimensions in those packets will trivially allow an attacker to correlate packets sent from the same vehicle before and after a pseudonym change. The extent of this problem has not been investigated in detail so far, however, a similar problem can be shown for data in cryptographic certificates [14].

This raises the question whether changing pseudonyms provide a sufficient privacy protection for V2X.

From our point of view (and missing stronger privacy protection schemes for V2X), we consider changing pseudonyms as a best available technique (BAT), a term introduced in the so called Sevilla Process, to describe a process where a technical solution is applied that is generally considered by experts in the field as the best solution that is currently available. Introducing such an approach for privacy protection in ITS has been discussed in the eSecurity Working Group [15]. Therefore, application of changing pseudonyms can be seen as both sufficient and required for privacy protection in V2X.

3.3 Pseudonym Resolution

When applying changing pseudonyms, one open issue is whether those pseudonyms should provide an unconditional anonymity, i.e., whether nobody should be able to identify vehicles (within the constraints outlined above)?

The pseudonym solutions currently under discussion [13] in principal include the option that the provider of pseudonyms retains a mapping between an issued pseudonymous certificate and the identity of the holder of this pseudonym.

If this is stored, the pseudonym provider could be required or forced to reveal the identify of a pseudonym holder in cases like legal disputes or crime investigations.

Basically, pseudonym systems can be categorized with respect to their pseudonym resolution characteristics as follows:

a) Technically possible and implemented: the system supports the solution and it is actively used.

b) Technically possible, but not implemented: while it would be possible to implement a pseudonym resolution, it is not actively used.

c) Technically possible, but constraint: while it is possible to resolve pseudonyms, there are technical constraints that prevent misuse of this mechanism.

d) Technically impossible: the system is constructed in a way that effectively prevents pseudonym resolution.

While current solutions fall in a) or b) – depending on whether the pseudonym-identity mappings are stored or not one can also envision mechanisms that fall into the categories c) or d). [16] discusses an approach that falls in category c). Here, pseudonyms can only be resolved under well-defined conditions, requiring the collaboration of a dedicated group of resolution authorities. Misuse by the pseudonym provider is thereby prevented. With small modifications, the solution can also be made a category d) solution that offers complete anonymity.

Whether pseudonym resolution is a relevant option depends on a number of decisions that relevant stakeholders need to make. This might also require explicit laws to be passed that clarify whether, e.g., interests of law enforcement outweigh data protection requirements. This matter has to be discussed considering and weighting up interests of

a) Law Enforcement

b) Data Protection

c) Customer Protection

d) Driver Interests

e) Security Requirements

It will likely be the case that this trade-off will be seen very differently in different countries of the world. Therefore, a technical solution that could support categories a) – d) would be clearly preferable.

The technical solution should be flexible and fulfill the following requirements if possible:

- Resolution of pseudonyms should only be possible by authorized authorities. Better, one authority alone should not be able to request resolution information. May a data protection agency could be involved in the resolution process.
- Resolution of pseudonyms should be possible by different authorities that may not be involved in the enrollment process of vehicles and maybe not in the process of pseudonym acquisition. Possibly, the authority that desires the pseudonym resolution is not available in point of time when pseudonyms are issued.
- The technical solution for pseudonym resolution should not affect message and certificate formats as well as communication via G5A. Pseudonyms created in different PKI domains that follow possibly different resolution strategies a) - d) should be compatible.
- Different types of pseudonym resolution should be possible
  - Full identity resolution → mapping between pseudonym ID and long-term ID
  - Linking of pseudonyms → misbehavior evaluation may need only to known that pseudonym A belongs to the same ITS-S as pseudonym B)

3.4 Summary

From our point of view, cooperative ITS and V2X communication systems will clearly process personal data and it is evident and generally agreed that privacy precautions need to be taken. Pseudonym schemes should be considered as a Best Available Technique, however, a broad review of existing schemes and proposals is still missing. The question whether a scheme should allow some sort of pseudonym resolution or should provide complete anonymity is still open, but should be addressed and solved in a broad discussion among stakeholders. If a pseudonym resolution is to be included, it should be clearly defined who can resolve pseudonyms under what circumstances. In such a case, a technical solution should be defined that prevents pseudonym resolution in all other cases. Finally, more research is needed to investigate the exact level of privacy protection that a pseudonym scheme can provide.

4 Pseudonymous Authentication - Survey

4.1 Introduction & Motivation

A future challenge for vehicle manufacturers is to develop smart vehicles in order to enhance the driving environment (safer, optimized and fun). To achieve this goal, wireless vehicular communications are being developed actively since the past few years. Standards describing vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communications are investigated. Due to cost constraints, the deployment of V2I is slowed down compared to V2V, where no infrastructure is mandatory to ensure the service. Among the envisioned services, the safety-related applications have caught a lot of attention because of their implication on the reduction of fatalities, a top priority of governments. Examples of safety-related applications are Local Danger Warning (LDW), Electronic Emergency Braking Light (EEBL) or Cooperative Collision Avoidance (CCA). Exchange of information between vehicles are mandatory to make these applications work properly. Indeed, On-Board Units (OBU) need information like position, speed, heading from their neighborhood to create a local view of the road around them. Such kind of information are available by the frequent broadcast of beacons.

As these applications are safety-related, they need to be secured. More especially, beacons have to be authenticated in order to ensure that the sender is a valid vehicle. Indeed, in EEBL application for example, a vehicle receiving an alert will brake in response to avoid the collision. That is why it needs to be sure that the sender is a valid vehicle. Therefore, without proper security mechanisms the safety-related application could be jeopardized, provoking accidents and consequently, users will not trust the V2X system anymore. To provide authentication (identity, message and attribute) a signature and a corresponding certificate are appended. The signature reveals the identity of the vehicle while the certificate proves the validity of the signature used.

At the same time, privacy needs to be ensured in order to avoid tracking attacks (thanks to the location provided by the beacons) or worse revealing the identity of the vehicle or the driver. A basic approach is to remove any vehicle/driver identifier from the message. So, we are facing a conflict between authentication and privacy. To solve this issue, an idea is to use a permanent (long-term) pseudonym. Indeed, a certified pseudonym will ensure the identity authentication while preserving the anonymity of the vehicle/driver [17].

A permanent pseudonym is not the solution, because tracking attacks are still possible and some works [9, 18] have proved that it is easy to identify or track the driver. For example, if an attacker wants to follow a vehicle V and the pseudonym is never changed, it is trivial

to follow V (if non overlapping pseudonyms are assumed). The need of a pseudonym change scheme is clearly identified.

In this chapter, we first detail the pseudonym lifecycle and use it to investigate the main approaches: asymmetric cryptography, identity-based cryptography, group-based cryptography and symmetric cryptography.

4.2 Pseudonymity and the Pseudonym Lifecycle

Digital pseudonyms were originally introduced by Chaum in the context of providing anonymity for electronic transactions as a "a public key used to verify signatures made by the anonymous holder of the corresponding private key" [19]. Pfitzmann and Hansen generalized this notion. They characterize a digital pseudonym as "a bit string which [...] is unique as identifier (at least with very high probability) and suitable to be used to authenticate the holder's items of interest relatively to his/her digital pseudonym, e.g., to authenticate his/her messages sent." [17]

Taking these two definitions together, it follows that a pseudonym, or pseudonymous credential, should be usable for authentication but must not contain any personal identifiable information that could link to the pseudonym holder's long-term identity. However, all actions authenticated with the same pseudonym are linkable because a pseudonym constitutes a unique, albeit short-lived, identifier. Short-term linkability of vehicular messages may be forced when the vehicle establishes a communication session with roadside units or it may be desired in order to facilitate and render safety applications (e.g., collision warnings/avoidance) more effective [20, 21]. Nonetheless, long-term linkability of pseudonymous actions is typically not desired [22]. Unlinkability of pseudonymous actions can be achieved by either changing pseudonyms over time or by using different pseudonyms for different contexts [17].

If non-repudiation is a desired characteristic, for example, to achieve accountability and audit-abilty [22], the secret part of the pseudonym must only be known to the pseudonym holder and sharing of secret credentials between users must be de-incentivized. If non-repudiation is guaranteed, accountability or traceability can be achieved with conditional pseudonymity. The basic idea here is that in normal operation peers only learn the pseudonyms of a node, but that privileged authorities have the ability to resolve a given pseudonym to the respective identity of the pseudonym holder under specific conditions. This is usually achieved with an identity escrow scheme in which an authority acts as a mediator for pseudonym generation. After authenticating a node's unique identity, the authority issues pseudonyms to that node and retains the capability to map issued pseudonyms to the pseudonym holder's identity. Approaches exists to enhance privacy in conditional pseudonymity by requiring multi-party cooperation for pseudonym-identity resolution, as will be discussed later on.

In vehicular networks, pseudonyms are employed as a mechanism to balance basic system requirements, security requirements, and privacy requirements [23–25]. The safety-critical nature of the road environment requires message authentication and accountability

is seen as an important deterrent against system misuse. While anonymity and untracability are essential to protect the privacy of individual drivers; privacy schemes must protect against linking of pseudonyms or network identifiers to the driver's identity and against tracking of specific nodes [26]. Pseudonym approaches for vehicular networks have to balance these seemingly contradicting sets of requirements without compromising the functionality of the vehicular network. Thus, pseudonym mechanisms must adhere to real-time or near real-time constraints of safety applications, support VANET-specific communication patterns, such as beaconing, multi-hop communication and geocast [27], and provide robustness and scalability [28].

As a result of the tension between these requirements, a multitude of pseudonymity mechanisms have been proposed since the inception of the field. Pseudonym schemes can be broadly categorized into approaches based on asymmetric cryptography (see Section 4.3), identity-based cryptography (see Section 4.4), group-based cryptography (see Section 4.5), and symmetric cryptography (see Section 4.6). While this categorization is useful to group similar approaches, it does not readily facilitates comparison of these categories or their schemes. But due to the requirements imposed by vehicular networks, an abstract pseudonym lifecycle can be identified which is similar for almost all analysed pseudonym approaches for vehicular networks.

Figure 4.1 gives an overview of the pseudonym lifecycle and its steps. Many of the lifecycle steps directly affect each other. Pseudonym issuance must already take pseudonym resolution and pseudonym revocation into account, and these steps inherently depend on the measures taken in the pseudonym issuance process to be effective. Pseudonym use and pseudonym change influence each other and also depend on how pseudonyms are issued or obtained by vehicles.

Each step is defined in the following subsections and specific challenges are pointed out and discussed. Subsequently, the categories named above will be used to structure the remainder of this survey, and the pseudonym lifecycle will be used in the discussion and analysis of different schemes to structure and compare them. The pseudonym lifecycle also allows us to put publications dealing only with specific aspects of the lifecycle in better relation to other work. Thus, we achieve a coherent overview of the current state of the art research on pseudonyms in vehicular networks, despite the variety of approaches that have been applied to the topic.

4.2.1 Pseudonym Issuance

Schemes commonly presume that a vehicle has a unique digital identifier, VID, and it can be authenticated as such. Similar to the vehicle identification number (VIN), which is embossed onto the vehicle chassis by the manufacturer, the VID is assumed to be preinstalled in a vehicle's OBU. A vehicle's long-term identity could also be issued by a vehicle registration authority, such as the department of motor vehicles (DMV), and is therefore sometimes also referred to as an electronic license-plate (ELP) [29].

In the pseudonym issuance process, the unique VID is required to authenticate the vehicle's OBU as an actual vehicle OBU to ensure that no other entities than vehicles can

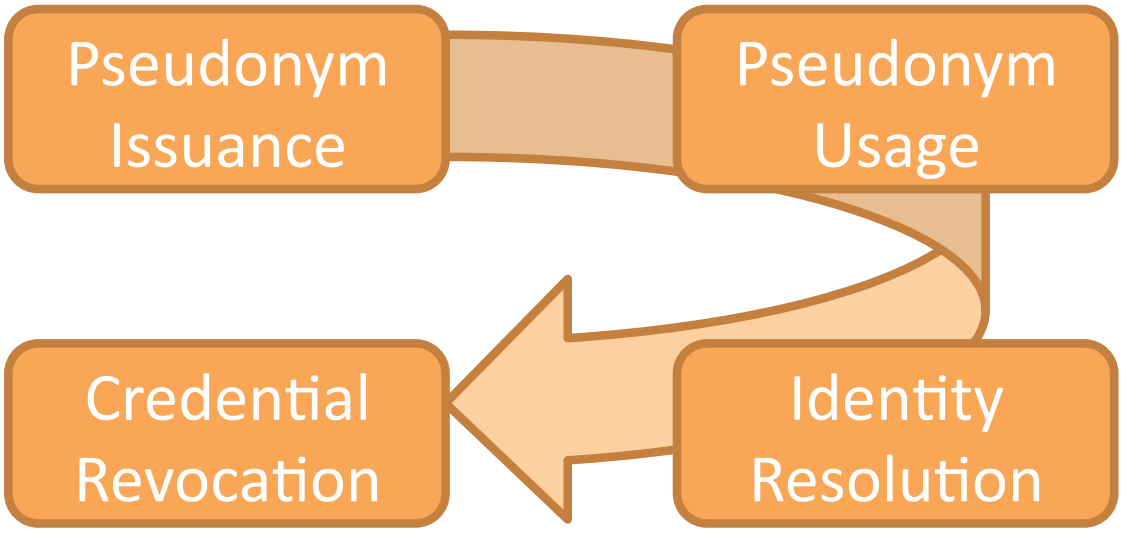


Figure 4.1: Abstract pseudonym lifecycle for vehicular networks.

obtain pseudonyms and thus join the vehicular network. For pseudonym issuance, two major approaches for certifying the authenticity of a pseudonym need to be distinguished: *third-party issuance* and *self issuance*.

The majority of approaches relies on third-party issuance, whereby pseudonyms are certified by authoritative entities. Depending on the scheme, these entities are referred to as *certificate authorities (CA)*, *pseudonym providers (PP)* or just *trusted authorities (TA)*.

The authoritative entity authenticates the vehicle with its VID, verifies the vehicle's eligibility to obtain pseudonyms (i.e., the vehicle's VID is valid and has not been revoked), and then issues certified pseudonymous credentials corresponding to the authentication scheme employed in vehicular communication (see Sec. 4.2.2). Depending on the scheme, a request-reply pattern is used to issue certified credentials or credentials are jointly computed.

The authoritative entity may retain escrow information to enable pseudonym-identity resolution later on. The authoritative entity gains the ability to revoke privacy of individual vehicles by linking pseudonyms back to VIDs. This either requires considerable trust in the pseudonym issuing authority or mechanisms to restrict pseudonym resolution capabilities. Furthermore, resolution information must be well protected to ensure that this information cannot be compromised by attacks against the authority's infrastructure [26]. See Sections 4.2.4 and 4.2.5 for detailed discussions on resolution and revocation.

Pseudonyms are typically assigned an expiry date or validity period to limit the number of pseudonyms available to a vehicle at any given time in order to prevent Sybil attacks [30]. In a Sybil attack, a single adversary poses as multiple legitimate vehicles by sending authenticated messages under multiple pseudonyms or identities simultaneously. The unlinkability property of pseudonyms prevents receivers from determining that theses messages originated from a single node, without further plausibility checks. Thus, the adver-

sary could try to propagate a specific viewpoint in the network to obtain an advantage on the road. For example, a greedy driver could simulate congestion on a stretch of road in order to persuade others to avoid it and gain a clear path to the destination [31].

Due to pseudonym changes as well as expiry of pseudonyms, many schemes require vehicle to obtain new pseudonyms occasionally. Whereby the frequency of these pseudonym refills depends on pseudonym change rate or pseudonym validity periods. Different strategies for pseudonym issuance have been proposed to address pseudonym refill which will be discussed within the later chapters where applicable.

In contrast to third-party issuance, pseudonym self-issuance has the advantage that issuance and generation of pseudonyms can be performed autonomously by the vehicle without requiring further interaction with authoritative entities once the vehicle's OBU has been initialized. However, Sybil attacks are generally harder to prevent in these schemes due to this autonomy.

4.2.2 Pseudonym Use

Once a vehicle has obtained pseudonyms it can engage in vehicular communication with other vehicles or infrastructure nodes. Pseudonym use entails two types of pseudonymous authentication: (1) authentication of messages to be send and (2) authentication verification of received messages.

The authentication of the vehicle's own messages allows other network entities to authenticate the sender as a vehicle with valid credentials. Message integrity must be protected to prevent modification of messages in transit. The message authentication scheme must also provide replay protection. Sender authentication, message integrity, and replay protection essentially corroborate the reliability of received information, which may then be used for safety critical decision making [28].

Typically, pseudonymous authentication schemes employ either digital signatures or message authentication codes to achieve this. On the receiver side, sender authentication entails verification of the validity of the employed pseudonym. A pseudonym must have been issued by a trusted authority or through verifiable self-issuance and must not be expired or revoked. Online verification with the support of back-end services is assumed to be infeasible due to intermittent connectivity with road-side infrastructure and real-time requirements of cooperative safety applications. Thus, all required verification information must be available locally. For example, schemes based on asymmetric cryptography need to attach pseudonym certificates to messages in order to enable signature verification by receivers (see Sec. 4.3). At the same time, communication overhead for security functions must be kept as low as possible to facilitate efficient and scalable use of the wireless medium.

Another challenge in pseudonym use is the inherent asymmetry between creating authenticating information for own messages to be send and verifying the authenticity of received messages. Typically, a vehicle must verify considerably more messages than it sends [32]. For example, in periodic beaconing vehicles may send beacon messages with frequency

r Hz, but assuming n neighboring vehicles in reception range it must verify approximately $n \cdot r$ msg/s. Thus, verification of messages and pseudonym credentials must be highly efficient in order to support applications with real-time requirements.

Pseudonyms can only be meaningful credentials for governing participation in vehicular networks, if private or secret keys are securely stored inside vehicle OBUs. For this reason, the integration of hardware security modules (HSM) or tamper-proof devices (TPD) in OBUs for key protection and management has been proposed [33, 34]. Hardware protection of credentials is also seen as an approach to prevent Sybil attacks by making only a limited set of pseudonym credentials available for use in parallel.

4.2.3 Pseudonym Change

Actions performed under one pseudonym can be linked to each other, due to the mentioned characteristics of pseudonyms. Thus, to prevent linkability of actions, actions must be performed under different pseudonyms, i.e., a vehicle must change its pseudonym sporadically. An adversary could then only link a limited number of messages.

In order to be effective, pseudonym changes must encompass all network layers [35]. When changing to a new pseudonymous authentication credential, application, protocol, and network identifiers, such as IP or MAC addresses, must all be changed as well to avoid trivial linking between old and new pseudonym.

The frequency of pseudonym changes depends on the desired level of privacy, i.e., what change rate is considered sufficient to prevent adversaries from deriving driving and movement patterns of individuals.

Topics of active research are also how, where and in what kind of situations pseudonyms should be changed in order to be effective. An example comes from mix-zones, discussed in 3.2. Pseudonym changes must not interfere with safety applications but must also be effective to prevent tracking based on vehicle trajectories and coordinates in beacon messages [36] or radio fingerprinting [37]. Proposed schemes vary between the different categories discussed later on, with a major focus of research on pseudonym change mechanisms and strategies for asymmetric schemes.

4.2.4 Pseudonym Resolution

While the previous steps concern all participants of a vehicular network, pseudonym resolution is only of relevance to hold misbehaving nodes accountable. Law enforcement representatives might capture pseudonyms from misbehaving nodes and pose a pseudonym resolution request to the issuing authority or pseudonym provider to obtain the VID of the pseudonym holder. The authority verifies the legibility of the request and could divulge the pseudonym holder, if respective pseudonym-identity mapping information has been retain upon pseudonym issuance.

While in the simplest case pseudonym resolution could be realized as a database look-up, more advanced resolution schemes have been proposed to enhance individual privacy by restricting resolution capabilities. Proposals include the separation of pseudonym issuing and pseudonym resolution authorities, thus limiting this conditional linkability to the single pseudonym and the pseudonym holder. Rather than facilitating the linking of all messages sent by a vehicle, and the use of threshold cryptography or secret sharing schemes, co-operation linking information is accessible only if all parties agree on the necessity of pseudonym resolution for a given misbehaving case.

Interestingly, while many pseudonym schemes foresee resolution capabilities on a technical level, the legal and societal implications of conditional pseudonymity in vehicle communications systems are not clear. Especially in Europe, the legality and requirement for conditional pseudonymity has been highly debated in recent years. It remains unclear if future vehicular networks will need to support pseudonym resolution or not.

4.2.5 Pseudonym Revocation

Misbehaving or faulty nodes may need to be revoked from the vehicular network, to ensure proper performance, security and correct operation of the network. Commonly, node revocation entails revocation of the node's authentication credentials, such as the pseudonyms, VID, or both. If specific pseudonyms are revoked, one must accept the possibility that the corresponding vehicle may have further pseudonyms to continue communication with, if all pseudonyms should be revoked they must be somehow linkable with some additional revocation information to determine that they all belong to the same pseudonym holder, thus weakening the privacy provided by pseudonyms.

The decentralized nature and large scale of vehicular networks makes distribution of up-to-date revocation information a major challenge for effective pseudonym and node revocation [38]. Thus, instead of distributing revocation information, e.g., certificate revocation lists (CRLs), some schemes rely on passive revocation. Pseudonyms are issued with very short lifetimes requiring frequent pseudonym refills with pseudonym providers. If a node should be revoked, the node's long-term identity (e.g., the VID) is revoked and subsequent pseudonym refill requests are then denied. In this case, a revoked vehicle may still participate in the network until it runs out of valid pseudonyms. Typical approaches for pseudonym and node revocation will be discussed in each section.

4.3 Asymmetric Cryptography Schemes

Pseudonymous communication can be achieved with traditional public key cryptography schemes (PKI) by equipping vehicles with a set of public key certificates and corresponding key pairs. The public key certificates are used as unlinkable pseudonyms and thus cannot contain any identifying information. Vehicles sign messages with the secret key of the currently active pseudonym and attach the signature, as well as the corresponding

pseudonym certificate, to the message. Receivers can verify a message signature based on the pseudonym certificate, but are unable to determine the sender's identity.

The first propositions to ensure privacy in vehicular networks were based on asymmetric cryptography. This approach is followed by the SeVeCom project [39], [24], [20], [40], IEEE 1609.2v2 standard [41] and Car-to-Car Communication Consortium PKI Memo report [42]. When the scheme differs from the general approach we explain the difference in a specific paragraph.

- *Pseudonym issuance*: In asymmetric cryptographic-based schemes, the architecture used to issue pseudonym is similar to Public Key Infrastructure (PKI). A hierarchical CA structure is used where CAs manage and issue long-term certificates to vehicles. Pseudonyms are issued by one of the Pseudonym Certification Authorities (PCAs). They are only valid for a short period of time. When issuing pseudonyms, the security system, involving the PCA and the Long-Term CA (LTCA), authenticates a vehicle, establishing it is a legitimate vehicle and it keeps the pseudonyms-to-identity mapping in case of liability investigation. The secret keys of the pseudonyms are stored and managed by a Hardware Security Module (HSM), which is tamper-resistant to restrict the parallel usage of pseudonyms.
- *Pseudonym use*: The pseudonym is used to sign every outgoing packet. The pseudonym restriction schemes (lifetime, amount of pseudonyms in parallel, if any, etc.) should be done while considering the assurance level of the secure hardware in the OBU.
- *Pseudonym change*: A pseudonym has a lifetime. When it expires, the OBU loads a new one from its store or request a new one from the pseudonym provider.
- *Pseudonym resolution*: Identity resolution is performed by pseudonymity resolution authorities, which either keep mappings between long-term identity and pseudonyms or have access to such mappings kept by pseudonym providers or CAs.
- *Pseudonym revocation*: Certificate revocation can be performed on both pseudonyms and the long-term certificate. If limited to the long-term certificate, better efficiency is achieved while trading off protection. If the long-term credential is revoked, no new pseudonyms can be obtained. A Certificate Revocation List (CRL) would only have to be distributed to pseudonym providers. But a vehicle would remain capable of participating pseudonymously in the network until all its pseudonyms are expired.

4.3.1 Pseudonymous Public-Key Infrastructure

In the traditional PKI an issue is that vehicles have to acquire new certified pseudonyms periodically. Zeng proposed a Pseudonymous PKI (PPKI) approach that enables users to generate CA-certified pseudonyms themselves, thus reducing the communication overhead [43]. This approach was applied to the VANET domain by Armknecht et al. [44] and differs from the general approach in the pseudonym issuance and revocation cycles.

Concerning the pseudonym issuance, PPKI does not distribute pseudonyms for the vehicles. Instead, vehicles generate their own pseudonyms according to their master keys, which are chosen by themselves and certified by the certificate authority. PPKI utilizes advanced cryptography, such as bilinear paring and zero-knowledge, to realize pseudonym and message authentication without originator verification. Since PPKI asks vehicles to issue their own pseudonyms, there is no PPs in this system.

If a user has to be revoked, only the CA can reconstruct the owner of a pseudonym certificate. Whenever a key has to be revoked the CA publishes some data depending on which the nodes have to update their keys. For this purpose the data is chosen such that it cannot be used by the excluded node, thus impeding it from updating its master key.

4.3.2 V-token

Based on SeVeCom, V-tokens [45] further enhances the privacy protection by separating the roles of certificate authorities (CAs), PPs, and resolution authorities (RAs). CAs issue credentials of v-tokens for vehicles. V-tokens are randomized ciphertexts which hide the identities of the vehicles and which can reveal the identities of the vehicles only by the RAs. A vehicle uses a credential of v-token to request a pseudonym from a PP. Then the PP checks the credential and leaves the v-token in the issued pseudonym. The broadcast authentication process is more or less the same with SeVeCom, while the identity resolution process incorporates more than one RAs to engage in a homomorphic threshold decryption scheme (e.g. ElGamal [46]). This scheme focuses on issuance and resolution cycles.

4.4 Identity-based Cryptography Schemes

The identity-based cryptography (IBC) is close to the asymmetric-based cryptography approach. Indeed, IBC is based on the idea of deriving public keys from identifiers. Presented with a signature, a verifier can check its validity merely by knowing the sender's identifier. Public keys or additional certificates are not required, because authenticity is implicitly guaranteed due to the fact that only authorized entities receive a secret key corresponding to an identifier. The secret keys have to be generated and assigned by a centralized trusted authority to prevent that anyone with knowledge of an identifier can derive a corresponding private key.

Compared to conventional PKI, IBC infrastructure avoids the use of certificates for public key verification and the exchange of public keys (and associated certificates) greatly improving the computation and communication efficiency.

- *Pseudonym issuance*: A vehicle requests pseudonyms from a RSU by sending its identifier and the corresponding certificate (encrypted with the RSU's public key). The RSU authenticates the vehicle based on the certificate and checks that the

identity certificate has not been revoked. Then, the RSU encrypts the vehicle identifier and a time stamp with its symmetric key SK_i. The result is concatenated with the RSU identifier, a time stamp and a string denoting the pseudonym holder as a vehicle; thus forming the pseudonym identifier PID_i. A pseudonym private key PSK_i is extracted from the pseudonym identifier. The pseudonym key pair (PID_i, PSK_i) is then encrypted with the vehicle's public key and send to the vehicle.

- *Pseudonym use*: The vehicle uses PID_i as sender address and signs messages with PSK_i. Receiver verifies the signature based on PID_i and the published system parameters.
- *Pseudonym change*: Vehicles have to request new pseudonyms periodically, similar to public key schemes discussed in Section 4.3, but less storage space is required because only the pseudonym identifier and the corresponding secret key have to be stored rather than an additional public key certificate. When a vehicle requests new pseudonyms it authenticates itself with a unique identifier, and before generating pseudonyms the issuing authority checks that the vehicle is not listed on a certificate revocation list.
- *Pseudonym resolution*: Identity resolution can be performed by the trusted authority by looking up the secret key SK_i of the RSU, specified in the PID_i, in a secret key database. The vehicles identity can then be decrypted with SK_i. A problem of the scheme is the reliance on symmetric keys shared between trusted authority and RSUs—a user is not protected against abuse of the authority conferred to RSUs. Furthermore, a centralized secret key database is a worthwhile target for adversaries.
- *Pseudonym revocation*: The revocation problem has been recently recognized as a great concern for IBC [47].

4.4.1 Secure revocable anonymous authenticated inter-vehicle communication

Fisher et al. [48] introduced a pseudonym issuance protocol that makes use of blind signatures and secret sharing to ensure that several authorities are required to cooperate in order to resolve a pseudonym. This protocol is named SRAAC (Secure revocable anonymous authenticated intervehicle communication) and involves multiple servers to issue pseudonyms to vehicles. Hence the resolution of anonymity also requires multiple servers. The common feature of these schemes is that a temporary (short-lived) public key is used as a pseudonym of the vehicle. In that way the temporary (short-lived) public key has two roles: a temporal id of the vehicle, and a public key for signature verification. Based on blind signature and secret sharing in the pseudonym issuance protocol to enforce distributed pseudonym resolution.

- *Pseudonym issuance*: In the pseudonym issuance process, a user blinds the public key to be signed and presents shares of it to a number of certification authorities. Each authority holds a partial secret of a secret key, which is shared between

all servers in a secret sharing scheme. Each authority performs a signature with its partial secret key on the presented blinded key part, returns it to the user, and stores a corresponding partial resolution tag in a database. The user can unblind and combine the received results, yielding a certificate which can be verified with a public key common to all authorities. The certificate is only valid if k of n servers participated in the issuance process, because otherwise the threshold of the secret sharing scheme is not reached, thus resulting in an incomplete signature.

- *Pseudonym resolution*: All partial resolution tags stored by the certification authorities can be combined for pseudonym resolution. To resolve a pseudonym, more than t servers have to cooperate in a second secret sharing scheme in order to link a pseudonym certificate to a tag in the database. They compute a joint tag for the presented pseudonym which then has to be compared to all tags in the database. Although the approach effectively prevents misuse of resolution authority, it also incurs considerable overhead by requiring a number of servers to take part in the certification of a single pseudonym. Furthermore, pseudonym resolution requires comparisons with all tags stored in the revocation database, and therefore, does not scale well with the number of users.

4.4.2 AnonymSign

AnonySign [49] is also based on IBC on bilinear maps but enables signature verification without the need to disclose the signer's identifier. A trusted authority assigns unique identifiers ID_i to vehicles, and computes corresponding private keys (D_i, S_i). A vehicle A computes a signature on a message m with D_A and S_A. A receiver B only needs its own identity ID_B, as well as secret keys D_B and S_B, to verify that the signature originated from someone with valid private keys from the same IBC scheme. This is possible due to properties of bilinear mappings, whereby two expressions constructed from D_B, S_B, and the signature components hold true if, and only if, all involved private keys have been created under the same secret system parameter t. Therefore, the scheme does not require periodic pseudonym changes, because no identifying information is included in any signatures. Identity resolution can only be performed by the trusted authority with knowledge of t, but requires computations with the secret keys of each registered user until an equality is fulfilled.

4.5 Group-based Schemes

Group signature-based schemes are proposed in [50, 51], where signer privacy is conditional on the group manager. As a result, all these schemes have the problem of identity escrow, as a group manager who possesses the group master key can arbitrarily reveal the identity of any group member. In addition, due to the limitation of group formation in VANETs (e.g., too few cars in the vicinity to establish the group), the group-based

schemes may not be applied properly. The election of the group leader could encounter some difficulties since the trusted entity cannot be found amongst peer vehicles.

- *Pseudonym issuance*: Group-oriented signature schemes enable an entity of a group to produce a signature on behalf of the group. There are two major paradigms in anonymous group-oriented signature schemes: group signature and ring signature. Ring signature scheme provides a similar feature. It does not support anonymity revocation mechanism, but no setup stage is needed to produce and distribute a group secret explicitly. Hence it enables any individual to spontaneously conscript arbitrarily $n-1$ entities and generate a publicly verifiable 1-out-of-n signature on behalf of the whole group, yet the actual signer remains unconditionally anonymous. Threshold ring signature is the t-out-of-n threshold version where t or more entities can jointly generate a valid signature but $t-1$ or fewer entities cannot [51].
- *Pseudonym use*: Each vehicle uses the group signature to sign messages.
- *Pseudonym change*: Not relevant.
- *Pseudonym resolution*: The group manager can resolve the identity of a group member.
- *Pseudonym revocation*: Not relevant. Revocation of the signer.

4.5.1 Efficient Conditional Privacy Preservation

The Efficient Conditional Privacy Preservation (ECPP) protocol deals with the growing revocation list while achieving conditional traceability by the authorities. ECPP [50] is also a pseudonym based system, which uses the PPs to generate pseudonyms and pseudonym credentials for the vehicles. Like in SeVeCom the long-term identity is also verified by PPs before issuing the pseudonyms.

ECPP uses bilinear maps to achieve conditional privacy. In ECPP, a vehicle uses multiple anonymous keys obtained from an RSU to prevent its communication from being traced. In addition to the provided anonymity features, the ECPP scheme suffers from three main drawbacks. First, it is not efficient due to two reasons: 1) it has fairly high latency for generation of pseudonym keys by the RSUs, and 2) it requires ubiquitous presence of RSUs to assist vehicles to derive their pseudonyms and corresponding keys at any given road location. Second, ECPP requires that the issued pseudonyms are known to the issuing authorities (i.e. RSUs) beforehand. Since RSUs are distributed in open areas along roads, they are usually vulnerable to physical attacks. Thus, they usually cannot be fully trusted. Third, there is no clear revocation mechanism of using ECPP. Since vehicles can derive their pseudonyms from every RSU, even a compromised one, malicious vehicles cannot be revoked [52].

4.6 Symmetric Cryptography Schemes

While symmetric cryptography is less flexible than asymmetric cryptography, it is well-known to have less computational overhead. In Inter-Vehicle Communication (IVC) exchange of information must be performed within a short period of time, which limits both the possible message generation time and the available effective bandwidth. This suggest that there are benefits associated with using symmetric cryptographic techniques, as these typically result in smaller transcripts [53]. At the same time, one can argue that the cost of deployment and maintenance of certification infrastructure is high, and that the availability of CA or CRL is not fully ensured. All these arguments explain why the symmetric cryptographic-based schemes could be an option for VANET deployment. The feasibility of this approach to build vehicular networks with balanced privacy and auditability was conducted by Choi et al. in 2005 [53]. Their solution is based on escrow mechanisms to enforce anonymity and resolution if needed.

In this type of scheme, vehicles share a secret key that it used for signing and verify message. To ensure privacy against peers, short-lived pseudonyms are used.

- *Pseudonym issuance*: The ombudsman generates the identification number, a seed value and registers the key of the vehicle. Each node can generate new pseudonym by computing a value which hides the pseudonym and the session key.
- *Pseudonym use*: Each packet (except for beacons) is tagged with the node's pseudonym. Since a pseudonym is coupled with a particular session key, a base station needs to find the session key to allow MAC verification. In case of V2V, the receiver will send the data to a base station that can verify the MAC.
- *Pseudonym change*: At the end of each short time interval, the corresponding pseudonym is updated. Keys could be periodically loaded, with periods up to once per year [54].
- *Pseudonym resolution*: The ombudsman escrows associations between identities and pseudonyms. It may collaborate with a base stations to reveal identities for given pseudonyms after the fulfillment of specific conditions such as law enforcement.
- *Pseudonym revocation*: The ombudsman knows the link between pseudonym and real identity, and thus, can revoke vehicle.

But even in symmetric cryptography schemes, infrastructure can be used. Indeed, [55] propose an approach that depends on RSUs in order to generate symmetric keys. When an RSU is detected a vehicle attempts to associate with it. The RSU assigns a unique shared symmetric secret key and a pseudo ID which can be released to other vehicles. To ensure anonymity this pseudo ID is associated with k vehicles. Utilizing the symmetric key and pseudo ID, the vehicle can generate a symmetric MAC code for any message that it sends to other vehicles (together with the RSU). Upon receiving a message the receiver must buffer the message until the RSU verifies the message, the MAC and notifies its authenticity through the periodic broadcasts of an aggregate of the hashes of authenticated

messages. This approach, however, heavily depends on the existence of RSUs which may not be possible at all times.

5 Privacy Enhancing Protocols

Vehicular communication (VC) systems are being developed primarily to enhance transportation safety and efficiency. Vehicle-to-vehicle communication, in particular, frequent cooperative awareness messages or safety beacons, has been considered over the past years as a main approach. Meanwhile, the need to provide security and to safeguard users' privacy is well understood, and security architectures for VC systems have been proposed. Although technical approaches to secure VC have several commonalities and a consensus has formed, there are critical questions that have remained largely unanswered: Are the proposed security and privacy schemes practical? Can the secured VC systems support the VC-enabled applications as effectively as unsecured VC would? How should security be designed so that its integration into a VC system has a limited effect on the system's performance?

This deliverable, as it will develop over the years of the project, will seek to address these questions. In the rest of this chapter, we first recap background regarding the current understanding of security and privacy enhancing technologies for VC systems. Then, we elaborate elements of different approaches that extends this basic approach towards further strength and flexibility, and discuss practical considerations in the context of harmonization towards standardization.

5.1 Background

Vehicular communication (VC) systems will comprise vehicles and fixed road-side equipment (RSU) with wireless transceivers, and sensing and processing units. Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I), that is, V2X, communication will enable a range of applications, with transportation safety playing a predominant role. Almost all research and development efforts converge to safety applications based on V2V communication, with vehicles frequently *beaconing* their status (e.g., position, speed, direction), along with warnings about potential dangers.

Nonetheless, VC systems can be vulnerable to attacks and jeopardize users' privacy: An adversary could, for example, inject beacons with false information, or collect vehicles' messages to track their locations and infer sensitive user data. Industry, academia, and authorities have recently understood that security and privacy protection are prerequisites for the deployment of VC systems. Security architectures were developed by the IEEE 1609.2 working group [25], the SeVeCom project [23,24], following the earlier NoW project [56] and now the Car-to-Car Communication Consortium (C2C-CC) [57] and the eSafety eSecurity WG activities [58].

Across projects and working groups, secure VC systems rely on public key cryptography and digital signatures to protect V2V and V2I messages; *Certification Authorities* (CAs) manage credentials for legitimate participants (vehicles and RSUs). Pseudonymous authentication, with vehicles using short-lived credentials and public-private key pairs, provides protection of privacy along with security (authentication, integrity and non-repudiation as primary requirements). Security mechanisms protect all traffic sent across the 802.11p data link [59], including the safety beacons each vehicle transmits, typically every 100 to 1000 ms, as per the ETSI CAM specification.

Adding security for this high-rate communication would incur high overhead, both in terms of communication and processing. Consider, for example, a vehicle receiving digitally signed safety beacons from a hundred vehicles within range; it would need to validate a high percentage or almost all of those within a short delay in the order of a hundred milliseconds [59]. Even if VC is effective under such dense network conditions, the additional security overhead could cause failure in meeting the delay and reliability requirements of safety applications. This is especially so because the VC environment lacks abundant resources (bandwidth, computational power).

The following question naturally follows: Can secure VC systems be practical? Given the current system constraints and design approaches, could the addition of security and privacy mechanisms make VC systems ineffective? Based on broadly accepted approaches for secure and privacy-enhancing VC [23–25, 56], we first outline how pseudonymous authentication is possible without repeated interactions with the CAs [60, 61]. Then, we present a proposal for reducing the security overhead without harming the effectiveness of the VC system, and we investigate how variants of secure VC instantiations affect the system performance. In particular, what we are after in the long run is a comprehensive evaluation of secure VC operations: (i) We evaluate the communication reliability, and then (ii) we determine if and how VC nodes can sustain the incurred processing load, providing an approximate analytical evaluation and closely matching simulation results. Having determined if VC nodes have sufficient processing power, (iii) we consider the overall system performance with respect to transportation safety and (iv) transportation efficiency, evaluating secure VC-enabled applications for a broad range of system configurations. Essentially, appropriately designed security and privacy-enhancing VC systems should be able to support applications (notably safety ones, as they are the most stringent in terms of requirements) as effectively as unsecured VC systems can. Moreover, (v) it is important to investigate revocation and have a practical method for anonymous authentication schemes in VC, and (vi) discuss additional technical issues and model and assess which processing resources will be needed for future systems.

5.2 Hybrid Authentication

5.3 Problem and Approach Overview

We want to determine whether the broadly accepted state of the art of secure VC is viable, especially considering how challenging VC environments are; because heavy-traffic scenarios (thus, dense network topologies) - with tens, one hundred or more vehicles (nodes) within range - can often occur. The traditional approach has been to analyze the protocol overhead and the network performance. However, in VC systems the objective is not to have a well-performing network *per se*, but to effectively support VC-specific applications. This is why we investigate the overall system performance, considering five dimensions: *(i) communication technology, (ii) system resources, (iii) network configuration and environmental factors, (iv) security protocols, and (v) supported applications.*

The technology commonly accepted for V2V and V2I communication is the IEEE 802.11p [62], which is incorporated in the Dedicated Short Range Communication (DSRC) - Wireless Access in a Vehicular Environment (WAVE) [63] and the Communication Access for Land Mobiles (CALM) [64] standards. Vehicles transmit periodic *safety beacons* on one dedicated channel, at a system-selectable beaconing rate. *Bandwidth*, one of the primary system resources, is determined by the standards, and it is considered fixed for this investigation. The second primary resource, *processing power*, can be adapted. Here, we take into consideration platforms that are currently used in VC prototypes, but any system should have sufficient processing power for its designated tasks. Thus, the system designer can always increase the processing power at the expense of increased cost.

The use of specific *cryptographic primitives* and other *protocol functionalities* determine the processing load for each node (vehicle). Here we consider the basic pseudonymous authentication approach, which has gained broad acceptance: It provides message authentication, integrity, non-repudiation and it makes it hard for two or more messages from the same sender to be linked[1]. Given the large number of temporary identities (pseudonyms) in the system, pseudonymous authentication can become cumbersome to manage. Therefore, we consider here a novel scheme, first presented in [60, 61], to alleviate this constraint, thanks to a more powerful but also more expensive anonymous authentication primitive. We describe these security protocols in Sec. 5.4.

We consider *transportation safety and efficiency applications* as they are distinctive features of VC systems (compared to other mobile computing systems) and two main driving forces for the VC systems deployment. Moreover, they are, especially the safety ones, the most challenging among VC-enabled applications; their stringent time constraints and their critical nature can affect the well-being of the vehicle passengers. We focus here on

[1] More precisely, it allows that messages produced by a node over a protocol-selectable period of time, τ, be linked. But messages m_1, m_2 generated at times t_1, t_2 respectively, such that $t_2 > t_1 + \tau$, should not be linkable. The shorter τ is the fewer the linkable messages are and the harder tracking a node becomes.

one safety application, *Emergency Braking Notification (EBN)*, and one efficiency application, *Decentralized Floating Car Data (DFCD)*.[2]

In order that the appropriate processing power can be determined and provisioned, we provide a framework to analyze the effect of a given *processing load* on the node performance. Then, we consider a system for which processing is not a bottleneck (otherwise, the system would certainly fail) and we evaluate the effectiveness of the applications. Conversely, given appropriate design choices (i.e., equipment with sufficient power), our investigation reveals the effect of other parameters and their interdependencies.

5.4 Secure Communication

Each node (vehicle) has a long-term, unique identity and corresponding credentials managed by a *Certification Authority (CA)*; without loss of generality, we assume there is a single CA, even though in reality a CA hierarchy would be present [65]. Instead of using their long-term credentials, vehicles obtain from the CA and use a set of short-lived certified public keys that do not identify the vehicle; then, they digitally sign messages with the corresponding private keys. As this is the widely used approach of *pseudonymous authentication* [23–25, 56, 57], we refer to it as the *Baseline Pseudonym (BP)* scheme, and define its operation in Sec. 5.4.1. We consider only the vehicles, as the privacy of RSUs or other infrastructure does not need to be protected.

As the BP scheme requires numerous short-lived certificates and keys per vehicle, the stronger the protection of privacy is sought the higher the number of identities would be. For large-scale systems, this and the cost of periodically pre-loading vehicles with temporary keys and credentials can become a significant burden. To reduce the key management complexity and enhance the system usability and efficiency, we propose that nodes self-generate, i.e., self-certify, their own pseudonyms. With this method, first described in [60, 61], vehicles do not need to be side-lined or to compromise their user's privacy if a "fresh" pseudonym is no longer available; no "over-provisioning" in the supply of pseudonyms is necessary; and the cost of obtaining new pseudonyms over an "out-of-band" channel is avoided.[3]

This can be achieved with the use of *anonymous authentication* primitives, notably *Group Signatures (GS)* we describe in Sec. 5.4.2. As GS is hard to use for all VC messages, because of the GS processing and communication overhead, in Sec. 5.4.3 we propose our *Hybrid Pseudonym (HP)* scheme that allows vehicles to generate on-the-fly their pseudonyms, by combining the BP and GS approaches. HP alleviates the management overhead of the BP, but in principle it is more costly than BP (due to HP's use of GS). To reduce the cost of HP to equal roughly that of BP and to increase the robustness of any pseudonymous approach, we propose a set of optimizations (Sec. 5.4.4).

[2] The terminology for the former in the ETSI Basic Set of Applications document (ETSI TS 102 637-1 V1.1.1 (2010-09)) is "Emergency electronic brake lights".

[3] Recall that VSS v.1 does not implement this approach but rather relies on traditional public key cryptography.

Concerning revocation, all the approaches make use of *Revocation Lists* (RL), generated by the CA and distributed to vehicles primarily via the infrastructure [23, 65]. When a node validates a certificate, it checks whether the sender is revoked; if successful (i.e. the sender is not revoked), it proceeds with validating the message (signature(s)).

5.4.1 Baseline Pseudonym (BP) Scheme

Each node V is equipped with a set of *pseudonyms* that are certified *public keys* without any information that identifies V. More specifically, for the i-th pseudonym K_V^i for node V, the CA provides a certificate $Cert_{CA}(K_V^i)$, which is simply a CA signature on the public key K_V^i (unlike the common notion of certificate, for example the X.509 certificate). The node uses the private key k_V^i for the pseudonym K_V^i to digitally sign messages. To enable message validation, the pseudonym and the certificate of the signer are attached in each message. With $\sigma_{k_V^i}()$ denoting V's signature under its i-th pseudonym and m the signed message payload, the message format is:

$$M1 : m, \sigma_{k_V^i}(m), K_V^i, Cert_{CA}(K_V^i)$$

Upon receipt of *M*1, a node, with the public key of the CA available, validates $Cert_{CA}(K_V^i)$, and then verifies the signature using K_V^i.

Each pseudonym is used at most for a period τ (referenced in the rest of the paper as the *pseudonym lifetime*) and then discarded. We abstract away a number of possible implementation aspects, such as (i) the dynamic adaptation of the period of pseudonym usage, (ii) the number of pseudonyms (K_V^i and the corresponding k_V^i, $Cert_{CA}(K_V^i)$) that are pre-loaded to V, (iii) the frequency of pseudonym refills, and (iv) policies for pseudonym change, such as factors rendering a pseudonym change unnecessary (e.g., a TCP connection to an access point), and interactions of pseudonym changes with the network stack [66]. All these are important yet largely orthogonal to this investigation. The CA maintains a map from the long-term identity of V to the $\{K_V^i\}$ set of pseudonyms provided to a node. If presented with a message *M*1, the CA can perform the inverse mapping and identify the signer.

5.4.2 Group Signature (GS) Scheme

Each node V is equipped with a secret *group signing key* gsk_V, with the *group* members comprising all vehicles registered with the CA. A *group public key* GPK_{CA} allows for the validation (by any node) of any *group signature* $\Sigma_{CA,V}$ generated by a group member. Intuitively, a group signature scheme allows any node V to sign a message on behalf of the group, *without* V's identity being revealed to the signature verifier. Moreover, it is impossible to link any two signatures of a legitimate group member. Note that no public key or other credentials need to be attached to an anonymously authenticated message; the format is:

$$M2 : m, \Sigma_{CA,V}(m)$$

Group signatures, introduced by Chaum [67], are revisited in numerous works, e.g., [68–71], with formal definitions in [72, 73]. For the rest of the discussion, we assume and use the group signature scheme proposed in [74]. If the identification of a signer is necessary, the CA can perform an *Open* operation [72, 73] and reveal the signer's identity.

5.4.3 Hybrid Pseudonym (HP) Scheme

The combination of the BP and GS schemes is the basic element of our proposal [60, 61]. Each node V is equipped with a group signing key gsk_V and the group public key GPK_{CA} (recall that the group is the total of vehicles registered with the CA). Rather than generating group signatures to protect messages, a node generates its own set of pseudonyms $\{K_V^i\}$ (according to the BP public key cryptosystem). As for the BP scheme (Sec. 5.4.1), a pseudonym is a public key without identification information, and $\{k_V^i\}$ is the set of corresponding private keys. For HP, the CA does not provide a certificate on K_V^i; instead, V uses gsk_V to generate a group signature $\Sigma_{CA,V}()$ on each pseudonym K_V^i instead. In other words, it generates and "self-certifies" K_V^i on-the-fly, by producing $\Sigma_{CA,V}(K_V^i)$. Similarly to *M*1, V attaches $\Sigma_{CA,V}(K_V^i)$ to each message, and signs with the corresponding k_V^i:

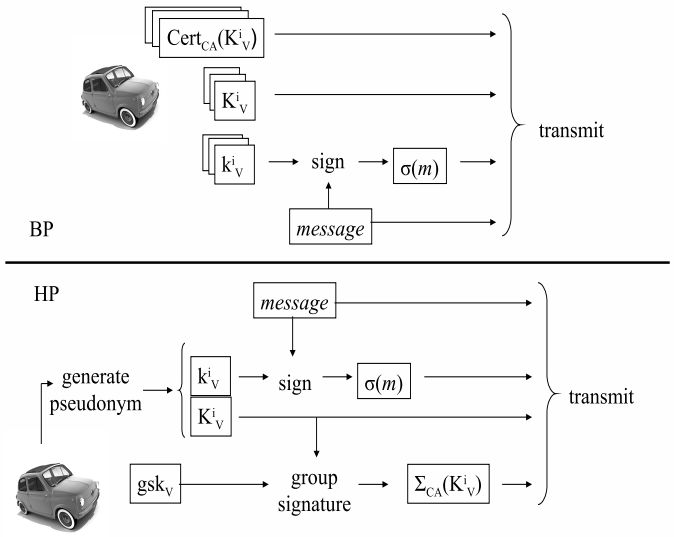
$$M3 : m, \sigma_{k_V^i}(m), K_V^i, \Sigma_{CA,V}(K_V^i)$$

When a node receives a message *M*3, the group signature $\Sigma_{CA,V}(K_V^i)$ is verified, using GPK_{CA}. If successful, the receiver infers that a legitimate system (group) member generated pseudonym K_V^i. We emphasize that, as per the properties of group signatures, the receiver/verifier of the certificate *cannot* identify V and *cannot* link this certificate and pseudonym to any prior pseudonym used by V. Once the legitimacy of the pseudonym is established, the validation of $\sigma_{k_V^i}(m)$ is identical to that for *M*1. To identify the message signer, an *Open* on the $\Sigma_{CA,V}(K_V^i)$ group signature is necessary; message m is bound to K_V^i via $\sigma_{k_V^i}(m)$, and K_V^i is bound to V via $\Sigma_{CA,V}(K_V^i)$. Fig. 5.1(a) compares the BP and HP.

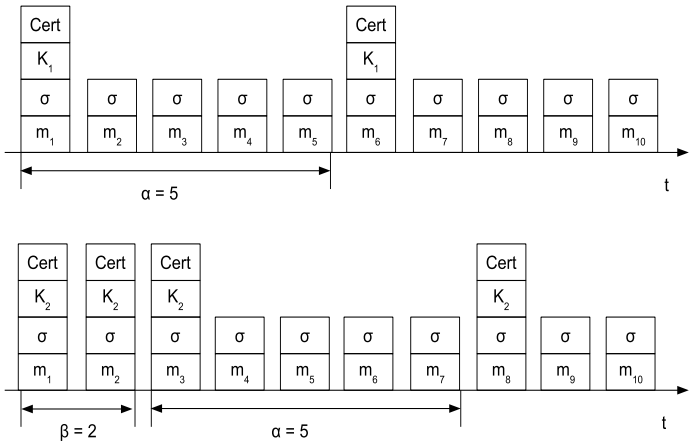
5.4.4 Optimizations for the BP and HP Schemes

We describe optimizations to reduce overhead (Optimizations 1 and 2) and enhance robustness (Optimization 3). We employ the notation of the HP scheme, but the same considerations hold for BP too. Fig. 5.1(b) summarizes Optimizations 2 and 3.

Optimization 1 On the sender's side, $\Sigma_{CA,V}(K_V^i)$ is computed only once per K_V^i, because $\Sigma_{CA,V}(K_V^i)$ remains unchanged throughout the pseudonym lifetime τ. For the same reason, on the verifier's side the $\Sigma_{CA,V}(K_V^i)$ is validated upon the first reception and stored, even though the sender appends it to multiple (all) messages. For all subsequent receptions, if $\Sigma_{CA,V}(K_V^i)$ has already been seen, the verifier skips its validation. This optimization is useful because in practice $\tau \gg \gamma^{-1}$, where γ is defined as the *beacon frequency*.



(a) Illustration of the BP and HP schemes.



(b) Illustration of Optimizations 2 and 3, with $\alpha = 5$ and $\beta = 2$. One *LONG* message is sent every 5, and repeated 2 times after a pseudonym renewal.

Figure 5.1: Illustration of the BP and HP security schemes and related optimizations.

Optimization 2 The sender appends its signature $\sigma_{k_V^i}(m)$ to all messages, but it appends the corresponding $K_V^i, \Sigma_{CA,V}(K_V^i)$ only once every α messages. We term such messages (*M*1 and *M*3) as *LONG*. *M*4 is defined as follows:

$$M4 : m, \sigma_{k_V^i}(m)$$

	Sign (ms)	Verify (ms)	Overhead (bytes)
BP *LONG*	1.3	7.2	141
HP *LONG*	54.2	52.3	302
SHORT	0.5	3	52

Table 5.1: Processing delay (in ms) and communication overhead (in bytes) for different packet types.

	Packets per beacon period γ^{-1}
BP *LONG*	13.9
HP *LONG*	1.9
SHORT	33.3

Table 5.2: Maximum number of verifiable packets per γ^{-1} s, for $\gamma = 10$.

We denote *M4* as *SHORT*, and α as the *Certificate Period*. $\alpha \in [1, \tau\gamma]$, where $\tau\gamma$ is the total number of transmissions during the pseudonym lifetime τ. To allow the user to choose the right K_V^i to verify an incoming *SHORT* message, all messages will carry a randomly generated 4-byte *keyID* field. This does not affect privacy as all *SHORT* messages signed under the same K_V^i can be trivially linked.

When a pseudonym change occurs, the new triplet $\sigma_{k_V^{i+1}}(m), K_V^{i+1}, \Sigma_{CA,V}(K_V^{i+1})$ must be computed and transmitted. V will sign messages with the new k_V^{i+1} corresponding to K_V^{i+1} from then on.

Optimization 2 can affect the protocol robustness, if the message that carries $K_V^{i+1}, \Sigma_{CA,V}(K_V^{i+1})$ is not received. Then, nodes in range of V must wait for α messages for the next pseudonym transmission, while being unable to validate *any* message from V. This can be dangerous if vehicles are close to each other and/or are moving at high relative speeds. Thus, we propose the following scheme to mitigate this problem.

Optimization 3 V repeats the transmission of $K_V^{i+1}, \Sigma_{CA,V}(K_V^{i+1})$ for β consecutive messages when K_V^{i+1} is issued, with β denoted as the *Push Counter*. After the β repetitions, with $\beta \in [0, \alpha - 1]$, the normal sequence 1 *LONG*, $\alpha - 1$ *SHORT* starts again.

5.5 Cryptographic Overhead

We use EC-DSA as the basic signature algorithm [75], the group signature algorithm proposed by [74], and security level of $t = 96$ bits for message signatures and $t = 128$ bits for CA certificates in BP and for group signatures used in GS and HP. High security might not be necessary for the short-lived K_V^i, but it is required for the long-term keys and CA certificates.

Overhead The $K_V^i, Cert_{CA}(K_V^i)$ is 89 bytes for BP, and with $\sigma_{k_V^i}(m)$ and *KeyID* the overhead is 141 bytes per message. For GS, the overhead is $\Sigma_{CA,V}(m)$, thus 225 bytes per message. For HP, the overhead is $\sigma_{k_V^i}(m), K_V^i, \Sigma_{CA,V}(K_V^i)$, *KeyID*, in total 302 bytes per message. For the $\alpha - 1$ *SHORT* messages, the overhead is $\sigma_{k_V^i}(m)$, *KeyID*, thus 52 bytes. The effective overhead reduction depends on the value of α.

Computation We make use of a Centrino machine with the clock speed set at 1.5 GHz, which is close to the CVIS (Cooperative Vehicle-Infrastructure System) vehicle PC, a rather powerful platform (compared to generally available embedded processors) adopted for the development of future VANET applications [76]. We obtain an EC-DSA benchmark on the platform through the OpenSSL standard test suite [77]. As for group signatures, a well-established implementation of the chosen algorithm [74] is not yet available. Thus, to estimate the processing delay, we calculate the number of 32-bit word scalar multiplications required for GS signing and verifying; we extract the relevant data from [78] and [79] and we benchmark the scalar multiplication operation.

Table 5.1 shows the costs for signature, verification and overhead for the chosen algorithms. To obtain individual processing delays for a given type of message, it suffices to take the sum of the corresponding cryptographic primitive delays (M1, M3 and M4). Security levels are $t = 96$ for $\sigma_{k_V^i}(m)$, and $t = 128$ for $Cert_{CA}(K_V^i)$, $\Sigma_{CA,V}(m)$ and thus $\Sigma_{CA,V}(K_V^i)$; we summarize results per message in Table 5.2 on page 35.

6 Other Research Results

In this section, we provide a concise discussion of related obtained research results, devoting one section per paper.

First, we are concerned with the need to authenticate traffic from remote nodes, notably for applications that "float" or disseminate data across multiple hops, e.g., destined for a specific area/location (Sec. 6.1). The problem addressed is how to process and validate, in terms of security, such traffic without degrading own operation, due to excessive processing load, and without reducing the level of security protection.

In Sec. 6.2, we are concerned with the correctness of position information provided by peer vehicles. This is critical for several aspects of vehicular communication-based functionality. We show how any node can validate positions advertised by its neighbors, operating independently and based on the inputs from its candidate neighbors. The benefit is twofold, allowing to verify or reject neighboring nodes (that can communicate directly with the verifier) based on the correctness of the position information they provide.

Unlike the volume of works in the predecessor projects of PRESERVE, which focused strictly on 802.11p based communications, in Sec. 6.3 we consider cellular communication (2/3G). In particular, the collection of traffic information data over such links. We address exactly the problem of securing this data collection and at the same time ensuring the protection of the information contributing user (through her or his smartphone).

Along the same lines, that is, considering cellular and other (notably 802.11) wireless communication, is the work in Sec. 6.4. Focusing on location based services, we address the problem of reducing the exposure of the user. Essentially, we allow users (their devices) to leverage peers for useful, up-to-date and information (with verifiable integrity) and thus avoid revealing their location and related activities.

6.1 Adaptive Message Authentication

Although very convenient for exploitation in vehicular networks, public key cryptography is costly and introduces significant processing overhead. Recent benchmarks, such as those obtained within the framework of the European eCrypt project [80], show that signature verification on a wide range of computing platforms takes a significant amount of time, even for the fast elliptic curve algorithms proposed for use in vehicular networks [66,81,82]. Due to the on-board vehicle equipment cost constraints, the currently envisioned automotive communication boxes face the same limitations: cryptographic message processing delays are typically in the order of several milliseconds [83]. More importantly, with tens of

nodes (vehicles) usually in proximity, each node has to handle and validate hundreds of messages per second.

We argue that the processing overhead in intermediate nodes can result in decreased network performance, due to the limited processing capabilities of the envisioned vehicular platforms. Our goal is to decrease the number of cryptographic operations performed by the nodes and to avoid a deterioration in performance due to processing power limitations. At the same time, we verify that this reduction of message verifications does not make a vehicular network more vulnerable to outside adversaries, nor to DoS attacks.

We focus on traditional approaches to identity management and secure inter-vehicle communication in vehicular networks, such as [84]. In terms of Inter-Vehicle Communication and multi-hop forwarding, these proposals recommend two extreme strategies. The first group of proposals requires intermediate nodes to verify that messages had been sent by legitimate senders and to check the integrity of the messages before resending them. We show that this approach to secure multi-hop forwarding tends to be too pessimistic and results in many unnecessary message verifications, degrading the network performance. On the other hand, the second approach, which advocates skipping message verification in intermediate nodes, neglects nodes' vulnerability to DoS attacks; although it performs well with few adversaries in the network, our simulations show that when no message verifications are performed, the goodput of legitimate nodes significantly drops as the number of adversarial nodes in the network increases.

The solution we propose [85] is an adaptive scheme which integrates the best features of the two aforementioned approaches. The aim is to make nodes perform only the necessary number of cryptographic operations while skipping the redundant message verifications and improving the overall performance of the network. The scheme takes advantage of the fact that nodes in different parts of a vehicular network face different security conditions at a given point in time. Essentially, we can view the vehicle finding itself in areas where (in the absence of misbehaving vehicles) there is little or no bogus messaging or in areas where there are frequent such messages. If we term the latter as "hostile" areas, then nodes in relatively less or not hostile areas can afford to be less cautious (check fewer messages) than others. On the other hand, given the dynamic nature of vehicular networks, the situation may change quickly and dramatically, so nodes should have the ability to adapt to changing circumstances.

Our contribution is twofold:

- We propose AMA (Adaptive Message Authentication), a scheme that probabilistically checks messages in intermediate nodes. Our scheme is reactive in that the checking rate increases to 100% only when forged messages are detected, and only for a limited period before returning to probabilistic checks. AMA is independent of the forwarding algorithm or the wireless standard that is used for communication and it can be easily integrated in the existing frameworks for secure communications in vehicular networks.
- We show through extensive simulations that the scheme guarantees substantial performance gains over the traditional proactive approach. The adaptiveness of the scheme brings increase in performance in cases with few adversaries as well as

in the cases when the adversarial nodes represent a significant percentage of the population.

6.1.1 Scheme Overview

The reasoning behind our scheme [85] is driven by the observation that the adversaries are limited in scope and that they cannot keep the whole network under attack at all times. Consequently, we designed a scheme that is shown in Figure 6.1. We call it AMA (Adaptive Message Authentication).

AMA has two modes of operation. We call them “check-all” and “relaxed”. The “relaxed” mode allows nodes to pay less attention to defensive measures. All the legitimate nodes are initially in the “relaxed” mode. It is this mode that is expected to bring performance gain to the scheme, as only a fraction of received messages are checked by a node in the “relaxed mode”. Nodes distinguish between the messages that have the current location of the node as the destination zone and those that only have to be relayed to others. Each message in the first group is checked with probability 1 and each message in the second group with probability p. If they happen to check a forged message, the forgery is always detected and it forces the node to switch its mode of operation to “check-all”.

“Check-all” mode is conservative and it mandates checking each received message. A legitimate node is expected to be in this mode when there are adversarial nodes nearby. A node stays in “check-all” mode until it receives c consecutive legitimate messages. Then, it switches back to “relaxed” mode.

The rationale is that if a node senses that there no adversaries in the neighborhood, that is, it receives few or no messages that do not pass the verification, a node can relay most of the messages without prior authentication and integrity checking. It can keep checking only a small fraction of these messages in order to ensure a timely detection of security threats. While the selected messages are being checked, the other messages that need to be relayed do not have to wait before being forwarded.

It is possible, of course, to use a different function for the checking rate increase, not just a step function. We show that even this simple scheme guarantees significant performance gains, for an appropriate choice of the parameters p and c, under very realistic assumptions (the scheme and both parameters p and c are known to the adversary).

6.1.2 Summary

Strict security requirements for vehicular communications led to several proposals that consider authentication and integrity check of each relayed message as necessary conditions for secure multihop inter-vehicle communication. This default approach brings considerable security overhead. Complex cryptographic operations, such as signature verification, introduce non-negligible processing delays. We show that the measured processing times on low-end platforms can result in degradation of network performance. On

the other hand, ignoring security can lead to DoS attacks and even more severe decrease in network performance.

With this scheme, we demonstrate that a simple, yet adaptive, filtering scheme that allows nodes to judiciously decide when to check the received message that requires further relaying, and when to simply forward it without any delay, brings significant performance gain. The scheme, which we term AMA, treats multihop messages in reactive rather than proactive way and requires checking of relayed messages only in the presence of a threat. Our simulations with the state of the art in vehicular routing algorithms show that, as a result of security overhead reduction, the goodput of legitimate nodes increases up to 33%. We believe that, because of the significant gains possible, this approach is worthy of further investigation.

6.2 Secure Neighbor Position Verification

In vehicular ad hoc networks, knowledge of neighbor positions is a requirement in a number of important tasks. However, distributed techniques to perform secure neighbor position discovery, suitable for highly mobile ad hoc environments, are missing. With this scheme, we address this need by proposing a lightweight distributed protocol that relies only on information exchange among neighbors, without any need of a-priori trustworthy nodes. We present a detailed security analysis of our protocol in presence of one or multiple adversaries, and we evaluate its performance in a realistic vehicular environment.

Privacy, traceability of misbehaving nodes, unauthorized message insertion, fraudulent relaying and insecure neighbor discovery are all aspects of primary importance that need to be addressed. Here, we tackle the latter issue, and, specifically, secure verification not only of the presence of neighbors but also of their exact location [86]. Indeed, VANETs are among the most likely candidates to benefit from Secure Neighbor Position Discovery (SNPD) when fine-grained location identification is required, e.g., in the case of congestion charging, traffic monitoring, traffic light prioritization for special vehicles, etc. Critical routing tasks, especially those based on geographic routing, also require that neighboring nodes are reliably identified and localized.

The challenges that an SNPD system must address are multi-faceted: (i) devices running an SNPD need to be able to track their own position and relate it to a common, reliable time reference; (ii) on-demand, real-time knowledge of neighbor positions and identities is needed; (iii) neighboring devices can be faulty or under the control or influence of an adversary, and must be properly detected. While we will assume that the devices are compliant with the first requirement, we focus on designing an SNPD mechanism that addresses the latter two requirements and allows nodes to validate the positions of neighbors within their communication range in a distributed manner.

Note that most of previous work [87–90] has focused on secure neighbor discovery, which represents a subset of the position discovery problem we target, since our goal is to assess not only the authenticity of would-be neighbors, but also the correctness of their position. Our same objective, i.e., secure neighbor position discovery and verification,

has been tackled before in the generic field of ad hoc networks, where nodes mobility is limited or absent. In addition, the solutions proposed for such environments rely on the presence of dedicated mobile or hidden base stations [91], or on the availability of a number of collaborating and trustworthy devices [92]. Finally, we remark that the SNPD problem significantly differs from that of location proving [93], whose goal is to allow a centralized authority to verify the position announced by one specific mobile user.

In this work [86], instead, we envision a system where nodes act individually but cooperate and leverage the contribution of neighbors to weed out wrong-doers. In such a scenario, we propose a lightweight, distributed, and efficient protocol that enables each node to discover and verify the position of its neighbors. The protocol can be executed by any node, at any point in time, without prior knowledge or assumed trustworthiness of the other nodes that participate. Also, our protocol can sustain high-speed nodes and leverages RF transmissions, since other types of communication require line-of-sight (e.g., infra-red) or have short ranges (e.g., ultra-sound) that make them inappropriate for VANETs.

6.2.1 Secure neighbor position discovery protocol

The SNPD protocol we propose allows any node in the network to discover and verify the position of its *communication neighbors* that participate in the protocol message exchange. The procedure is performed in a reactive manner, i.e., it can be run by any node at any time instant, by initiating the message exchange. Such node will be referred to as the *verifier*.

Our solution is based on a best effort, cooperative approach. It aims at verifying the position only of the neighbors with which the message exchange takes place successfully. It therefore disregards nodes for which the protocol exchange prematurely ends, e.g., due to message losses on the channel, or communication neighbors that refuse to take part in the protocol. The scheme assumes that the node position does not vary significantly during the message exchange, as confirmed by simulation results[1].

The SNPD protocol leverages the information collected by neighboring nodes thanks to the broadcast nature of the wireless medium. Such information, fed back to the verifier, is used to compute, via Time of Flight (ToF)-based ranging, the distance between pairs of neighbors. Based on this knowledge, the verifier performs security tests to tag its communication neighbors as:

- *verified*, i.e., nodes the verifier deems to be trustworthy;
- *faulty*, i.e., nodes the verifier deems to have announced an incorrect position;
- *unverifiable*, i.e., nodes the verifier cannot prove to be either correct or faulty – this may happen due to lack of sufficient information on these nodes or because the verifier cannot form a clear opinion on their behavior.

[1] In an overcrowded scenario featuring 50 neighbors moving at an average speed of about 30 km/h, the average duration of the message exchange spanned over 150 ms, resulting in an average of 10% of colliding nodes; such a time interval corresponds to an average position shift of 1.2 m.

Clearly, the objective of our SNPD protocol is to be robust to adversarial nodes in that it minimizes the number of unverifiable nodes and the number of positive/negative falses. By the latter we mean correct nodes declared as faulty and adversaries tagged as verified.

Below, we detail the message exchange between the verifier and its communication neighbors, followed by a description of the security tests run by the verifier.

6.2.1.1 Message exchange

Let t_X be the time at which a node X starts a broadcast transmission, and t_{XY} the time at which a node Y starts receiving that same transmission; p_X is the current position of X, and $\mathbb{N}_X$ is the current set of its communication neighbors.

Consider a verifier S that initiates the SNPD protocol. The message exchange procedure is outlined in Algorithm 1 for S, and in Algorithm 2 for any of S's communication neighbors.

node S **do**
  $S \rightarrow *$: $\langle$POLL, $K'_S\rangle$;
  S : store t_S;
  when *receive* REPLY *from* $Y \in \mathbb{N}_S$ **do**
    S : store $t_{YS}, \mathbb{c}_Y$;
  end
  after $T_{max} + \Delta + T_{jitter}$ **do**
    $S \rightarrow *$: $\langle$REVEAL, $E_{k'_S}\{h_{K'_S}\}, K_S, Sig_S\rangle$;
  end
end

Algorithm 1: Message exchange protocol: verifier node

The verifier starts the protocol by broadcasting a POLL whose transmission time t_S is stored locally (Alg. 1, lines 2-3). Such message is anonymous, since (i) it does not contain the verifier's identity, (ii) it is transmitted employing a fresh MAC address, and (iii) it contains a public key K'_S from a one-time use private/public key pair k'_S, K'_S, taken from a pool of anonymous keys which do not allow neighbors to map them onto a specific node. Note that including a one-time key in the the POLL also ensures that the message is fresh. Furthermore, including the public key in broadcast messages makes the protocol self-contained, as it does not have to rely on a separate asymmetric key exchange; as for key management, it can exploit one of the architectures proposed in the literature, e.g., [94].

A generic communication neighbor $X \in \mathbb{N}_S$ that receives the POLL stores its reception time t_{SX}, and extracts a random wait interval $T_X \in [0, T_{max}]$ (Alg. 2, lines 2-5). After T_X has elapsed, X broadcasts a REPLY message using a fresh MAC address, and records the corresponding transmission time t_X (Alg. 2, lines 6-10). The REPLY contains encrypted information for S, namely the signed neighbor identity, Sig_X, and the POLL reception time: we refer to these data as X's *commitment* and tag it as $\mathbb{c}_X$. The hash $h_{K'_S}$, derived from the verifier's public key, K'_S, is also included to bind POLL and REPLY belonging to the same message exchange.

forall the $X \in \mathbb{N}_S$ **do**
  when *receive* POLL *by* S **do**
    X : store t_{SX};
    X : extract T_X uniform r.v. $\in [0, T_{max}]$
  end
  after T_X **do**
    $X : \mathbb{c}_X = E_{K'_S}\{t_{SX}, K_X, Sig_X\}$;
    $X \rightarrow * : \langle \text{REPLY}, \mathbb{c}_X, h_{K'_S} \rangle$;
    X : store t_X;
  end
  when *receive* REPLY *from* $Y \in \mathbb{N}_S \cap \mathbb{N}_X$ **do**
    X : store $t_{YX}, \mathbb{c}_Y$;
  end
  when *receive* REVEAL *from* S **do**
    $X : \mathbb{t}_X = \{(t_{YX}, \mathbb{c}_Y)\ \forall Y \in \mathbb{N}_S \cap \mathbb{N}_X\}$;
    $X \rightarrow S : \langle \text{REPORT}, E_{K_S}\{p_X, t_X, \mathbb{t}_X, Sig_X\} \rangle$;
  end
end

Algorithm 2: Message exchange protocol: neighbor node

Upon reception of a REPLY message from a communication neighbor Y, the verifier S stores the reception time t_{YS} and the commitment $\mathbb{c}_Y$ (Alg. 1, lines 4-6). A different communication neighbor X receives the REPLY message broadcast by Y, if Y is a communication neighbor of both S and X, i.e., $Y \in \mathbb{N}_S \cap \mathbb{N}_X$. In such case, X too stores the reception time t_{YX} and the commitment $\mathbb{c}_Y$ (Alg. 2, lines 11-13). Note that also REPLY messages are anonymous, hence a node records all commitments it receives without knowing their origin.

After a time $T_{max} + \Delta + T_{jitter}$, S broadcasts a REVEAL message; Δ accounts for the propagation and contention lag of REPLY messages scheduled at time T_{max}, and T_{jitter} is a random time added to thwart jamming efforts on this message. Through the REVEAL, (i) S unveils its identity by including its signature and its public key to decrypt it, and (ii) it proves to be the author of the original POLL. The latter is achieved by attaching the encrypted hash $E_{k'_S}\{h_{K'_S}\}$ (Alg. 1, lines 7-9).

Once the identity of the verifier is known, each neighbor X, which received S's original POLL, unicasts to S an encrypted and signed REPORT message containing its own position, the transmission time of its REPLY, and the list of pairs of reception times and commitments referring to the REPLY broadcasts it received (Alg. 2, lines 14-17). Commitments are included 'as they are', since only S can decrypt them and match the identity of the nodes that created the commitments with the reported reception times. We also point out that, by transmitting its own position only after the reception of the REVEAL, a neighbor prevents a verifier from exploiting anonymity to run a flooding attack.

6.2.1.2 Summary

This is a lightweight, distributed scheme for securely discovering the position of communication neighbors in vehicular ad hoc networks. Our solution does not require the use of a-priori trustworthy nodes, but it leverages the information exchange between neighbors. Although simple, our analysis showed the scheme to be very effective in identifying adversarial nodes. Results derived using realistic vehicular traces confirmed such ability and highlighted the good performance of our solution in terms of both false negatives/positives and uncertain neighbor classifications.

6.3 Secure and Privacy Protecting Contributory Traffic Information Systems

Smartphones and portable personal devices, especially those integrating GPS receivers, have become common practice nowadays. Moreover, cellular networks offer very broad coverage. As a result, scores of new location-based applications and services have emerged. In fact, leveraging the smartphone capabilities and the dense infrastructure can be highly advantageous for Intelligent Transportation Systems (ITS) and traffic management applications: each smartphone could provide location samples to a traffic management server, and then provide its user with traffic information.

Smartphone-based ITS can have dramatically lower cost than traditional ones: they have no need for special in-car hardware, and they could reach fast high penetration rates. Major application platforms, Apple's iPhone and Google's Android, provide friendly development environments for prototyping. Moreover, features such as online digital maps or phonetic route guidance could be easily added. Finally, any driver with a relatively modern smartphone would be able to join the system. However, there are significant challenges to meet before deploying such a solution. On the one hand, obtaining location samples and traffic information must be secure. Otherwise, the traffic management server could receive forged location samples. Or, the mobile client could get corrupted traffic information responses. At the same time, the privacy of the system users cannot be at stake: No one would like to have information on his/her whereabouts, exactly what the mobile clients regularly send to the traffic management server, disclosed. Tracing an individual could lead to identification and even damages (e.g., PleaseRobMe [95]).

Existing commercial solutions [96, 97] rely on password based authentication and they provide a statement that they remove the user's identification from all contributed location samples; they pledge no private information disclosure unless this is required by the authorities. Our goal is to provide strong security, authenticating the individual contributions of the clients. We also want to provide privacy by design, notably by making location updates anonymous and unlinkable. In particular, we want to deprive the traffic management server from any chance to trace and identify users. In the rest of the discussion, for simplicity, we term this the ITS server.

The contribution of our work [98] is a practical approach to achieve this goal. The novelty of our proposal lies in that: we leverage traditional authentication services by cellular infrastructures, we augment those with anonymous authentication, and we keep the ITS service separate from the mobile operator. Users contribute encrypted, in an end-to-end manner, data to the ITS server, being anonymously authenticated. This keeps their data (location updates) confidential and unlinkable. Thus, the mobile operator cannot access such detailed location information; that is, more detailed than what mobile operators are already entrusted to maintain (they can determine roughly the whereabouts of any handheld in their network). More specifically, our proposed architecture augments the Generic Bootstrapping Architecture (GBA), standardized by the 3GPP [99], by anonymous authentication.

The security of the system and the privacy of its participants are two major challenges towards smartphone-based ITS. Our approach addresses these challenges by separating the authentication from the location data gathering system. Authentication for each user leverages the GBA architecture of the IMS. Then, anonymous authentication is used to access and provide data to the ITS server.

One drawback of the implemented group signature scheme is that when a user is revoked the legitimate users have to recalculate their keys. Group signatures schemes with verifier local revocation can be an alternative [100]. Furthermore, anonymous authentication in general introduces the problem of Sybil attacks against the ITS server: a misbehaving device could produce and sign multiple spurious location updates and send them to the ITS server. Due to their unlinkability, the server cannot link them to the misbehaving client and detect the abuse. Traditional approaches with pseudonyms can overcome this threat, using with certificates with non-overlapping validity [7] and one pseudonym used for each location update/sample. This variant of the pseudonym solution may have increase management cost (e.g. for preloading sufficiently many pseudonyms), but it may very well be practical due to low computational costs and the very low rates of updates for the considered traffic management application (e.g., compared to safety applications). Alternatively, the signing procedure could be controlled by a secure hardware module (e.g., the SIM card) [7] or group signatures with limited number of valid signing actions [101] could be used. Our approach enables the integration of these different cryptographic primitives to provide anonymous authentication and it leverages the mobile operator as a trusted third party. While traditional public key cryptography is easy to use on smartphones, we implemented a specific group signatures scheme, as a proof of concept to ascertain the feasibility of this type of anonymous authentication on smartphones.

6.4 Collaborative Location Privacy

Smart phones, among other increasingly powerful mobile computing devices, offer various methods of localization. Integrated GPS receivers or positioning services based on nearby communication infrastructure allow users to position themselves fairly accurately. This gives rise to a range of *Location-Based Services* (LBSs): users can query an LBS server and obtain information relevant to their current location and surroundings, that is, context

data about specific points of interest. The value of LBSs is exactly in obtaining accurate and up-to-date information on the fly.

The flip-side of getting on-site high-quality on-demand information is the loss of users' privacy: Each time an LBS query is submitted, private information is revealed. The user can be linked to her location, and multiple pieces of such information can be linked together; thus, the profiling of users becomes possible. Clearly, the user could forgo the LBS benefits; e.g., she could download a large data volume and then search locally about specific context information. But this would be cumbersome, if not impractical, for the user and it would be inefficient for obtaining information that changes dynamically over time.

LBS users are most often identified explicitly by their service, e.g., through the process of creating an account and logging in (e.g., [97,102]) to access the service. Thus, it becomes trivial for the LBS server (or anyone that can get access to the LBS logs) to link the user, even her real identity, to her location (where the queries were submitted from). But even if the LBS does not perform any explicit user identification, it is still possible to fingerprint users of specific applications [103], or de-anonymize them (i.e., infer their identity) by using their location [104], and then trace their whereabouts.

More important, independently of whether the user is identified or not, placing too much trust in LBS providers is undesirable. Indeed, the LBS operators may be tempted to misuse the rich data they collect. Or they may, as opposed to cellular operators (who have a contract with their users), share the data with third-party companies that offer, for example, targeted advertisements. Or the LBS data repositories may be targeted by attackers, who break into the LBS servers and obtain logs of user queries. The result in all cases is the same: user-sensitive data fall in the hands of untrusted parties.

Tracking the user over time and space, and then identifying her, implies not only loss of privacy for the user but possibly other dire consequences such as *absence disclosure*: learning that a user is away from her home could allow a house break-in or blackmail [95]. As a result, the need to enhance privacy for LBS users has been understood and several solutions have been proposed. One approach could be to blur location information, e.g., by having the user's client submit inaccurate samples to the LBS server. However, obfuscation approaches (e.g., spatial/temporal cloaking introduced in [105]) which can protect user location-privacy, degrade the user experience if users need high privacy: e.g., LBS responses would be inaccurate or untimely. Moreover, obfuscation cannot be effective against absence disclosure [106]. Another approach could be to introduce a third party in the system, acting between the user and the LBS: its role would be to protect the users' privacy. Such an intermediary server, between the user and the LBS, could anonymize (and obfuscate) queries by removing any information that identifies the user or her device [107, 108]. Or it could blend one query with those of other users, so that the LBS server always sees a group of queries [109]. However, such approaches only shift the problem: the threat of an untrustworthy LBS server is addressed by the introduction of a new third-party server. Some other approaches require the LBS to change its operation, for example, by mandating it needs to process modified queries (submitted by the intermediary in different forms than actual queries of the user), or that it needs to store data differently (e.g., encrypted or encoded, to allow private access [110]).

Any such centralized intervention or any substantial changes to the LBS operation would be hard to adopt, simply because the LBS providers would have little incentive to fundamentally change their operation. Misaligned incentives have been identified as the root of many security problems [111]. Additionally, new proxy servers become as attractive for attackers as centralized LBSs. Hence, the lack of incentives and guarantees for protecting the users' location information, make these approaches infeasible in practice.

In order to enhance the location privacy of LBS users without any of the above-mentioned limitations, we propose here a new *user-centric* scheme. Mobile users concerned about their location privacy are indeed the most motivated entities to engage in protecting themselves. Our solution, called MobiCrowd, takes advantage of this fact, making the privacy-sensitive users responsible for their own privacy protection. Our approach requires no change of the LBS server architecture and its normal operation, it makes no assumption on the trustworthiness of the LBS or any other third-party server, and it enhances the privacy of mobile users in terms of both presence and absence disclosure.

MobiCrowd [112] achieves this improvement thanks to a novel *collaborative privacy-protection mechanism*: basically, a user can avoid disclosing her location information if her device can have its LBS queries answered by nearby peers (i.e., other reachable user devices) that happen to have the sought data. We analyze our scheme experimentally and analytically, proposing an epidemic model for the dynamics of information sharing among users. The model captures the effect of many users clustering at the same place, and it can be used to test various "what-if" scenarios about MobiCrowd. This is a novel approach to evaluate a location-privacy preserving mechanism for mobile networks: it acts on the *parameters of their mobility model* rather than on some specific location traces. Thus, we can study the effects of a mixture of parameters and we can also identify the causes of high or low location privacy in various settings. We then perform a simulation on real mobility traces, and we show that the conclusions from the experimental evaluation verify the results derived from our model.

The threat of local observers sniffing the wireless channel trying to infer users' private information, is out of the scope of this scheme; such a threat could exist with or without MobiCrowd and it can be alleviated by frequently changing device identifiers (e.g., MAC addresses [113] for WiFi as it is done for GSM by changing TMSI [114]). More important, local observers would have a tedious task and still be ineffective in collecting information: they would need to be physically present next to any given victim user, over long periods and across different locations. In contrast, a centralized LBS can by default observe *all* the queries of a user, which is why we focus on this much greater threat here. However, in order to secure the scheme against untrustworthy users who might disseminate invalid or outdated information, the LBS information package (e.g., the set of points of interest) is proposed to be self-verifiable (i.e., be digitally signed by the server). In fact, this is the only change that MobiCrowd imposes on the LBS operation.

Our scheme [112] leverages capabilities of contemporary smart phones: They can establish ad hoc and infrastructure connections (e.g., cellular base stations and Wi-Fi access points). We build a *mobile transparent proxy* in each device that protects the users' location-privacy. Our proxy, transparently located on-board the user's device and between the LBS client and the network, maintains a buffer with location context information. This

buffer is checked for available data when the user submits a query. If the valid and up-to-date data is not available, our mobile proxy broadcasts the query (i.e., the type of required information) to other nearby devices. If and only if none of those neighbors can provide the requested information, the LBS is queried. We have implemented our scheme on the Nokia N800, N810 and N900 mobile devices, and demonstrated it with the Maemo Mapper (a geographical mapping software for points of interest) [115]. Note that our approach can be ported to the upcoming technologies that enable mobile devices to directly communicate to each other via (potentially more energy-efficient) Wi-Fi-based technologies [116–118] that aim at constructing a mobile social network between mobile users.

6.5 Dynamic Consensus for Secured Vehicular Ad hoc Networks

Safety related applications such as cooperative collision avoidance, local danger warning and road hazard notification could save lives. In fact, alerts from these applications enable the drivers to react to dangerous situations such as obstacles or bad road conditions, hence reducing the risk of an accident. It is crucial to make sure that the life critical information in these applications cannot be forged or modified by an attacker. Vehicular networks are especially vulnerable to *fake attacks* where misbehaving vehicles inject erroneous information into the network to affect the behavior of the other drivers for their selfish objectives. For example, in traffic congestion optimization, honest drivers may be misled and driven to congested area by falsely injected information, while the attacker vehicle can enjoy less traffic on its own path. More dangerously, the drivers may be misled into potential accidents.

From a security point of view, the decision whether or not such an application should rely on reported hazard, is a crucial issue, which cannot be completely protected by conventional security mechanisms. Conventional solutions, such as digital signatures, focus on securing the communication network. In this way, attackers are prevented from manipulating the network. But cryptographic protection mechanisms cannot verify information itself. In other words, manipulating sensor readings to simulate a false message may still result in a perfectly signed and certified message. Therefore, an additional application-level approach is required. A technique is to evaluate the plausibility of information received during the decision process. Thus, hardening the decision process against attacks.

To provide trust into these warnings and avoid inappropriate reactions, a simple way is redundancy. The consensus mechanism provides such property. Indeed, a vehicle—that implements the consensus mechanism—needs to receive X times the same warning from its neighborhood before making a decision—react or warn the driver [119]. A main issue is to define the decision method that sets X. This could be done in two ways: static or dynamic. In the static case, X is set at the manufacturing of the vehicle and could be changed only with human intervention—during annual vehicle inspection for example. In the dynamic case, X will change sporadically. We focus on a dynamic *threshold-based*

scheme which sets the minimum messages needed before reaction.

It is worth noticing that, depending on the decision method used to set X, the technique has an impact on the guarantees of real-time constraints of the application because of:

- the number of messages generated on the network.
- the delay to transmit one message.
- the processing time (because each message needs to be verified [120]).
- the delay to make a decision.

So the choice of the decision method should be done carefully. Moreover, each technique should be deeply investigated to assess its performance. In this work, we investigate the consensus mechanism to increase trust in local danger warning application. More especially, we focus on the decision method because it sets the consensus parameter and has an impact on the vehicle reaction. First, we propose a generic model that defines decision methods. As the vehicular network topology changes quickly, we aim at setting the consensus parameter dynamically. So, we propose a decision method that sets X and *Threshold* in function of the network density and the criticalness of the warning. We analyze the impact of these parameters on the decision delay and the braking distance. More especially, we compare our threshold-based decision method to the majority method. Our simulations show that the dynamic method has a higher decision delay than in the majority method. But the decision is still made before the braking time. We conclude that the dynamic decision method permits to increase the trust into the warning by collecting more messages than in the majority method without jeopardizing the braking distance. As of future work, we first intend to optimize the decision delay formula. Indeed, it is assumed that the X warnings are received continuously without considering the background traffic, the competition between warnings (for different events), or the queuing time. Our model should take into account these considerations.

6.6 Spoofed Data Detection in VANETs using Dynamic Thresholds

One of the primary motivations for research on inter-vehicular communication is deployment of safety applications such as cooperative collision avoidance, local danger warning, and road hazard notification. By wirelessly exchanging information on mutual positions, speed, and heading, the basic idea is to provide a better situational awareness for close-by vehicles so that local applications can decide whether there are potentially critical situations with a risk of collision or crash. These applications would then provide warnings to drivers in such critical driving situations. Expectations are that this will significantly reduce the numbers of traffic-related accidents and injuries.

However, this promise can only be fulfilled if the system works with very high reliability and this, consequently, requires resilience against security attacks. It is crucial to ensure

that situation information exchanged between vehicles cannot be forged or modified by an attacker. If you would assume that an attacker can provide wrong position or speed information to other vehicles, this will very easily lead to wrong or suppressed warnings and thus to inappropriate behavior of drivers. For example, a driver that is warned about an immediate crash ahead will likely break sharply and might cause rear-end accidents as an effect.

Without proper security mechanisms in place, inter-vehicular networks are especially vulnerable to such false data injection attacks where misbehaving vehicles inject erroneous information into the network for selfish or malicious reasons. Basic security mechanisms for Vehicular Ad hoc Networks (VANETs) suggest to use authentication and integrity protection mechanisms to ensure that only valid vehicles or road-side units participate in communication. This can be implemented using digital signatures and a Public-Key-Infrastructure (PKI) [121] as foreseen by all current standardization efforts [122]. But even in this case, one can still not trust that all (valid) vehicles report correct information [123].

Vehicles will typically receive information from multiple neighbors in their immediate surrounding. For this scheme, we assume that this happens by means of Wave Safety Messages (WSMs) as defined in [122]. Assuming that a certain fraction of vehicles is malicious and will report wrong data, this leads to the classical Byzantine Agreement (BA) problem [124] where some vehicles report a problem and some do not, but you do not know which are honest. A closely related sub-problem, the consensus problem, has been extensively studied in general distributed systems [125]. However, in contrast to general distributed systems, we are dealing with a very dynamic environment that requires near real-time decision making while at the same time facing bandwidth constrained communication channels.

Here, we are addressing the problem how to determine whether information about an event like icy road or an accident ahead is trustworthy or not. We assume that we receive information from multiple communication partners, some of which might be malicious. Applying a consensus mechanism allows the local On-Board Unit (OBU) to reach a decision before taking actions like warning the driver. Generally speaking, the OBU would require to receive a certain number of consistent reports about a specific event before a warning would be issued. Having such a consensus mechanism in place increases the trustworthiness of received warnings at the expense of additional delay as the OBU would first have to wait for reception of a certain number of messages to reach a certain *confidence threshold* [119].

A question that has been neglected by research so far is how to set this *threshold*. The chosen value will have an influence on a number of parameters like the required processing resources for checking the messages [120]. But most important, it influences the trade-off between the decision delay (and thus the delay until a driver gets warned) and the trustworthiness of the information (and thus the opportunity for an attacker to cheat). So a threshold must be chosen carefully.

There are a number of sub-problems that need to be addressed. First, an OBU needs to decide whether two event warnings received from neighboring vehicles relate to the same event and are subject to consensus checking or not. Assume, for example, that the OBU

receives warnings from three vehicles A, B and C, where A reports free road 40 meters ahead. However, B reports icy road 60 meters ahead and C reports dry road conditions 61 meters ahead (cf. Figure 6.2). Should those three reports be treated as one event — having a 2:1 majority for dry road ahead — or should we consider A separate and just look at the reports from B and C to check whether there is icy road or not. In the latter case, there is a 50% chance that the car will find icy road after 60 meters.

Current work on trust provisioning and consensus often assumes a unique event identifier and a perfect synchronization between vehicles to work around this problem. We think that such an assumption is not realistic. We will discuss how a consensus mechanism for VANETs can work without such identifiers.

A second issue is that a static threshold will not be sufficient. Depending on driving situations, type of event, previously received information, general context information, or possible reaction to warnings, a different level of trustworthiness might be required before reaching a consensus decision.

By analogy, when a driver drives down an unknown road, he will likely react immediately to any warnings he receives [126]. For example, when a driver A sees an upcoming vehicle flashing its headlights, A will assume that there is a problem or danger ahead and will likely react by slowing down. However, if A does not detect a hazard after a certain distance, he will conclude that it was a false warning or that the problem has disappeared. So, if later (at least after this certain distance), there is another vehicle flashing its headlights, then A might be less responsive and might only respond if two or more vehicles warn him. We use the same idea in our approach. Vehicles will constantly adjust their decision threshold by constantly computing the average “noise level” representing the probability of a false warning (implying the average level of attackers that might send wrong information).

By not assuming a unique event identifier and by having an adaptive threshold scheme, our approach has significant advantages over earlier proposals. It is more flexible and practical while still providing good detection capabilities for spoofed information. In this work, we investigate the problem of threshold establishment in VANET. We propose a dynamic threshold mechanism to increase trust in local danger warning by detecting spoofed data. More specifically, we model the threshold as a Kalman filter. We propose an algorithm similar to a learning scheme to dynamically adjust this threshold. Thus, the threshold estimates the current percentage of attackers in the VANET. We provide simulations and analyze the impact of the density and the percentage of attackers on the decision delay and the percentage of wrong decisions. Our method overestimates the presence of attackers but leads to protect vehicle from spoofed data injection. We conclude that the default threshold value should be chosen carefully to shorten the inevitable bootstrapping phase. Currently, we are working on further extensive simulations to assess the delay to achieve a best-suited threshold.

6.7 Privacy-by-design in ITS applications

The work in this and the next section relate to papers coauthored with partners of the PRECIOSA project. It is part of the liaison and dissemination activities related to privacy by ITS. It analyses how ITS applications may embrace a privacy-by-design approach. It takes a holistic viewpoint based on three founding principles: data minimization, enforcement and transparency. The impact on architecture and technology is presented. Three challenges for ITS deployment are then discussed: the intrinsic instability of the resulting engineering process; the impact on future ITS platforms; the difficulty to reach consensus. Finally, tangible steps are identified on how to go forward in terms of further research work as well as building further industry consensus.

6.8 Privacy Verification Using Ontologies

This work is also part of the liaison and dissemination activities related to privacy by ITS. It addresses the problem of privacy verification in a privacy-by-design process and extends current design methods by additional (formal) steps which take advantage of ontologies. The proposed extensions result in a systematic approach that better protects privacy in future information systems.

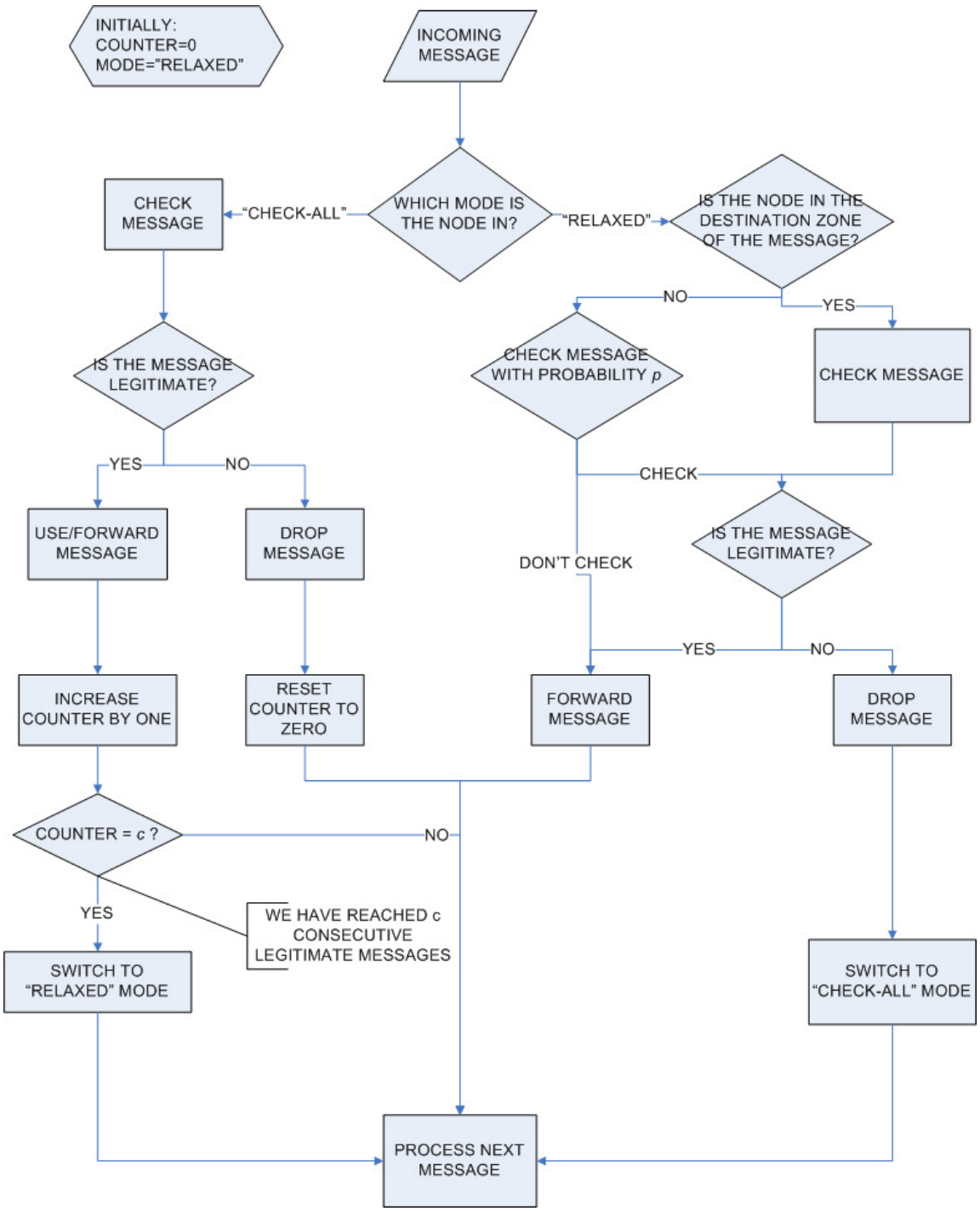


Figure 6.1: AMA - the scheme for adaptive authentication and integrity check of messages exchanged between vehicles. Briefly, an AMA node can be in one of two modes: "check-all" and "relaxed." A node starts in "relaxed" mode. In this mode, a node checks with probability 1 the messages destined for itself, but only with probability p the messages destined for other nodes. If it detects a forgery, the node switches to the "check-all" mode. In the "check-all" mode, a node checks all messages with probability 1, and switches to "relaxed" mode only if c consecutive legitimate messages are received.

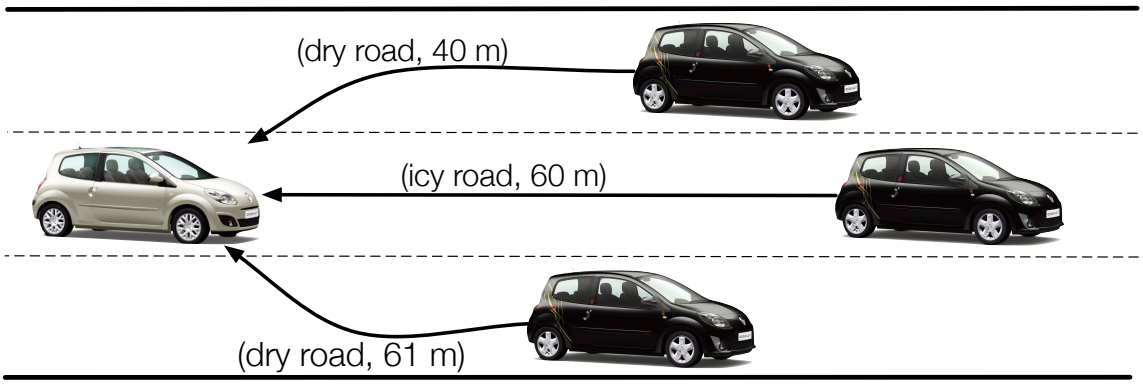


Figure 6.2: Problem of deciding on conflicting event notifications

7 Position of Security in the Protocol Stack

This chapter seeks to provide a clear statement, from the point of view of PRESERVE, on the position of the security processing into the communication stack. More precisely, the question is: Should the security processing be done at the network layer or the facilities layer?

PRESERVE partners are convinced that this question is important because directly impacts the Vehicular Security Architecture (VSA) implementation. Based on numerous discussions, this document tries to clarify the statement of PRESERVE. Section 7.1 details pros of the network layer. Section 7.2 details pros of the facilities layer. Section 7.3 discusses and states the final decision of PRESERVE, to position security at the network layer.

The main focus of PRESERVE is the message protection of V2X messages (i.e. CAM and DENM) that are specified by ETSI. Protection of other V2X communication protocols (e.g. IP) is not considered by this document.

7.1 Network Layer

This option avoids that security depends on the correct implementation of a certain application or could be undermined by changing or modifying applications. The security kernel and the core security processing (including checking/verifying signatures) should happen inside the stack independent of whether a certain application calls certain functions or not. So, if it's mandatory by a standard that CAMs have to be signed, then the communication system should ensure that this happens independently of a specific application using Local Dynamic Map (LDM) data. The network layer position is more generic than the application one. Indeed, it can cover other message types without need for re-defining the security payload. Moreover, the security will be implemented on the communication unit where, e.g., also IPsec could benefit from the availability of a hardware security module.

The data of the network header are protected by digital signature in order to avoid attacks on the routing. If the network stack would be able to pass metadata about a packet/message from the network layer to higher layers in the stack, then all cryptographic security processing could be performed on the network layer. Then the packet is tagged accordingly and these data are available to applications for decision processes (e.g. if an emergency vehicle contains this status in the certificate, this information becomes part of the metadata after verification of the certificate/signature).

The simTD project is performing security processing at the network layer and finds that metadata processing is a strongly desirable feature, as otherwise the data verified by the security system (e.g. attributes in the certificates) gets lost between layers. PRESERVE does not consider potential security compromise between layers as a major issue. Indeed, if an attack manages to manipulate the communication stack in a way that modifications of data between layers becomes possible, this attacker will likely also be able to directly manipulate facilities or applications.

Finally, this option fits with our testing plans where we might not have an application unit available all the time.

7.2 Facilities Layer

The application ultimately requires the assurance provided by the signature/certificate verification. If security is processed in this level then lower layer components may be changed without affecting the assurance on the application layer.

Facilities layer mechanisms can also be re-used by other applications/components without using the network layer security directly. Applying security on facilities layer avoids exchange of security meta-data between the layers of the communication stack. In some use cases, we need to authenticate the sending application of the message. So, applying a network-layer authentication is irrelevant.

The objective of PRESERVE is to secure applicative messages (such as CAM, DEMN). These messages are generated in the facilities Layer. A logical method is to apply the security services associated at the location of the message generation.

The security services to apply depend on the type of the message (based on the security requirements of the application that generates one or more types of messages). Therefore, for each type of message, we redefine the security functions necessary to apply. In the facilities layer, we do not need to know all the type of packet that could be generate and do not need to define multiple security policy as the application will decide itself.

We can secure the applicative data in layers below (network) but it is not the security of a message but the security of a packet that encapsulates an applicative payload (CAM). This requires more processing in the security layer. Indeed, the security layer has to interpret the packet, limit the fields and if necessary cancel the network header fields related to routing. And thus, it leads to more data streams exchanged between the network and facilities layers in both directions.

7.3 Discussion and decision

For reasons of information hiding if the application requires assurance of integrity then that means the application must provide the proof and the validation. If we can ensure that layer N-1 is able to give that assurance to layer N, then using the network layer (assuming

this is layer N-1) will be relevant. However, if the network layer is more than one layer away the level of assurance gets weaker and an attacker lying between the facilities and the network could modify the application's data and remain undetected with the network layer giving a false sense of security to the transmission.

But assuming that an internal attacker can modify things between the layers of communication stack, leads to trusted operating systems. Likewise, if communication layers are split between different physical entities, then it leads to trusted distributed systems. Ensuring that layers higher up than the network will check the integrity of the data is one of the goals of the OVERSEE project. Thus, this argument could be solved by the OVERSEE contribution.

Another aspect is that, in case of forwarding packets in multi-hop scenarios, security checking cannot be done at the facilities layer, as packets will likely be forwarded directly in the communication unit. Indeed, some data would be then unprotected at the network layer. Facilities layer processing has the disadvantage that network layer information cannot be protected by the same signature. Assuming that in the future more complex geonetworking protocols will be used and relevant information in the network header needs to be protected, this would lead to security on two different layers, which would double the processing and packet overhead.

This discussion also raised the issue of computational overhead. Indeed, CAMs and DENMs are used by different applications independently. Therefore, applying message signing and verification on facilities layer would lead to multiple verifications of the same message. PRESERVE also acknowledges that if different applications/facilities have different security requirements, this might complicate the network layer. There is a need to identify an interface mechanism by which the applications/facilities can signal the security requirements of a message or connection to the network/Security layer.

PRESERVE discussed about whether 1609.2 is really facilities or application layer security. While it clearly seems to be above network layer, it looks like 1609.2 is not caring too much about layers. Security processing is done at a layer in-between facilities and network layer. The security payload is specific per message type, however, the mechanisms are generic between messages. We argued that we should probably also be rather flexible with our mechanisms so that they can be applied at various layers.

To conclude, security processing at the network layer will permit the security to be transparent for the facilities layer. For outgoing messages, application just set if the message has to be signed or not. For incoming message, the network layer performs security processing (before applications can access the data) and then creating a certain "per packet" or "per session" security state that an application can access later on. Then, it forwards the data to the facilities layer with the meta-data needed. Indeed, attaching meta-data permits the application to have the possibility to check the meta-data from network layer (e.g. for consistency checks of location data) and security information from the respective layer. An option could also be to inform the facilities layer that the packet was not correctly signed and so leave to the facilities layer the final decision to discard it or not.

With the aforementioned arguments, the PRESERVE project states that the security processing will be done at the network layer. Moreover, thanks to the meta-data available for the facilities layer, the application could also apply its own security check.

8 Life-cycle and Operation Issues

The OBU will be integrated inside vehicles and will need specific operations during all along the life cycle. The objective of this chapter is to define all security issues which can happen. In Section 8.2, the different actions are listed. The corresponding events are associated to the actions. Finally, in Section 8.3, each action are described and, in particular, all security issues are identified.[1]

8.1 Actors and Physical Entities of the Life Cycle

Figure 8.1 describes the actors and physical entities that are used in the description of life-cycle use-cases. The description is based on the ITS architecture description of the U.S. Department of Transportation [1].

- **ITS Central Station**
  - **Installation Application Server** The Installation Application Server (IAS) provides software for the ITS-S (i.e. OBU and VSS). Possible operators of the server may be vehicle manufacturers or suppliers. The server is able to communicate with the ITS-S via wide area wireless communications (e.g. UMTS) or via fixed point entities (e.g. ITS Roadside Station, On-Board Diagnosis (OBD) at a garage).
  - The **Security Management** in the field is running a **Public Key Infrastructure (PKI)** that is used to provide certificates for secure V2X communication in the vehicle communication. Inside the PKI several Certificate Authorities are used to separate responsibilities. According to [127] the Root CA (RCA) acts as trust anchor in the ITS. The Long-term CA (LTCA) provides long-term certificates for all ITS-S that are allowed to be part of the ITS domain. With pseudonym certificates, provided by the Pseudonym CA (PCA), the ITS-S are able to cryptographically protect messages in the vehicle communication.
- **ITS Vehicle Station**
  - The **vehicle** consists of an On-Board Unit (OBU) that is running the ITS applications, the communication facilities (i.e. radio, communication stack, ...) and connects to the on-board network. The Vehicular Security Subsystem (VSS)

[1] Recall that the details on the FOT, notably which use cases applied, are available in deliverables of other WPs.

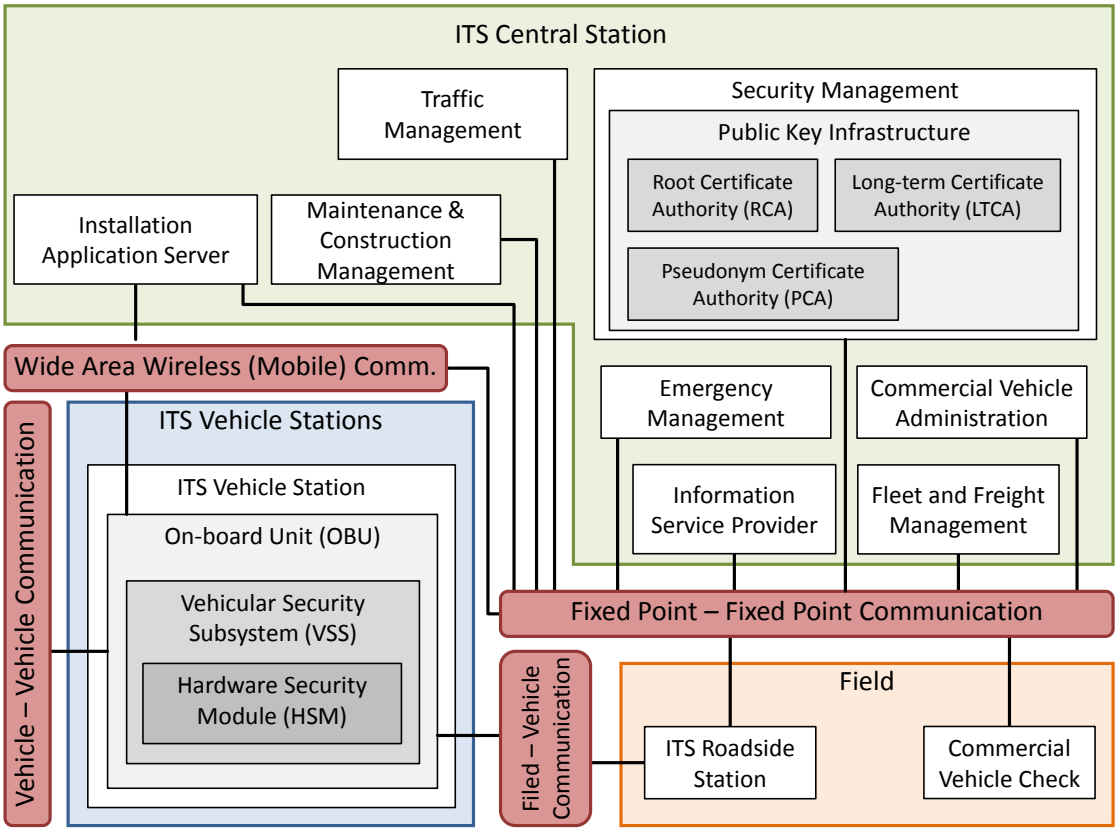


Figure 8.1: Actors and physical entities of the life cycle [1]

is connected to the OBU or is part of it. The VSS is providing security services to protect on-board communication and external V2X communication. A Hardware Security Module (HSM) is used in the VSS to store cryptographic credentials (i.e. private keys), accelerate cryptographic operations and acts as trust anchor inside the vehicle.

- **Field**
  - The **ITS Roadside Station**, also known as **Roadside Unit (RSU)**, consists of the same components as a ITS Vehicle Station (i.e. OBU, VSS, HSM). The RSU is able to act as gateway between the vehicle communication and fixed point communication. This enables multi-hop communication between vehicles and the central stations and entities of the field.
  - A **Commercial Vehicle Check** is a mobile or fixed point entity that is used to connect directly to a ITS-S. A garage or a dealer with an OBD tool for example is a commercial vehicle check that is able to establish a connection to ITS-S. A garage with additional authorizations may be able to process critical processes (e.g. secure software update, exchange of root certificate, etc.)

- The **buyer** or **owner** of an ITS-S is involved in the life-cycle process as initiator of actions. In case of problems, the owner of a vehicle contacts a commercial vehicle check that is able to detect the root cause and fixes the problem.

8.2 Actions during the Life Cycle

During the life cycle of an OBU, different use cases can raise security issues. The following list provides all the actions we have studied:

- **Installation Root CA.** The Root CA is acting as trust anchor for all entities of the ITS. As result, the Root CA has to be installed in one of the first steps of a new PKI domain. Table: 8.1.
- **Installation Long-term CA.** The Long-term CA is used by ITS-S to get long-term certificates. At the installation of a new PKI domain at least one LTCA has to be installed. Table: 8.2.
- **Installation Pseudonym CA.** The Pseudonym CA is used by ITS-S to get short-term pseudonym certificates. At the installation of a new PKI domain at least one PCA has to be installed. Table: 8.3.
- **First / Re-installation SW.** This use-case described the process when installing or reinstalling VSS software in the ITS-S. Table: 8.4.
- **First / Re-installation root certificate.** As the root certificate is used as trust anchor in the V2X communication security solution, the certificate of a commonly trusted RCA has to be installed before the ITS station can be registered the obtain further certificates. Table: 8.5.
- **ITS-S initialization and registration.** This use-case describes the bootstrapping process of new ITS stations with the PKI entities. After the station is registered, the initial certificates can be installed on the ITS-S. Table: 8.6. [2]
- **Secure software update of OBU.** The objective of this use-case is to download and update the software of the OBU except for the VSS software in order to keep the vehicle components up-to-date. Table: 8.7.
- **Secure software update of VSS.** The objective of this use-case is to download and update the VSS software in order to keep the vehicle components up-to-date. Table: 8.8.
- **Physical update or replacement of OBU.** There is a need for replacing the OBU together with the HSM, for example because the hardware is not functionning properly. Table: 8.9.

[2] It is likely that this use case unfolds asynchronously, in a sense: for example, at production and then continue with registration once a vehicle is sold, or later. We do dwell on this aspect here, as it will also depend on policy.

- **Physical update or replacement of HSM.** This use-case describes the process when the VSS HSM has to be physically updated (addition of component, update of physical component(s)) or replaced (change of the complete HSM). Table: 8.10.
- **Refill of Pseudonym Certificates.** The short-term pseudonym certificates are used to sign outgoing messages in the V2X communication between vehicles and between vehicles and the field. Due to the short lifetime of these certificates, a regular refill has to be done. Table: 8.11.
- **Update of Long Term Certificate.** The long-term certificate is used to identify a registered vehicle that requests short-term pseudonym certificates. Even though the long-term certificate is valid for a longer time period, the certificate expires and has to be renewed during the lifetime of a ITS-S. Table: 8.12.
- **Misbehavior Detection, Reporting, Evaluation, or Reaction.** The ITS stations detect on the one hand if the own system is manipulated. On the other hand misbehavior of other stations in the VANET is detected based on received messages. In both cases, a misbehavior report is sent to the infrastructure in order to identify possible attackers and exclude them from the network. Table: 8.13.
- **Revocation of ITS-S.** In case of misbehavior detection or compromision of ITS-S the long-term certificate is deactivated so that the station is not able to request pseudonym certificates. As long as the ITS-S is revoked/deactivated, active participation in V2X communication is not possible. Table: 8.14.
- **Revocation of Root CA.** If the private key of the Root CA is compromised or published then the RCA certificate becomes invalid. The detection of the RCA compromise can be done in different ways, which are out of scope of this document. Here, we assume that the event of the compromise is publicized by the RCA itself. In this case, all issued certificates of other CAs and all related long-term certificates and pseudonyms have to be renewed. Table: 8.15.
- **Revocation of Long-term CA.** If the private key of the LTCA is compromised or published then a new key pair has to be created for this LTCA. The Root CA issues the new LTCA certificate and the old certificate is added onto the CRL which is signed by the RCA. All related long-term certificates of ITS stations become invalid and new long-term certificates have to be requested. Table: 8.16.
- **Revocation of Pseudonym CA.** If the private key of the PCA is compromised or published then a new key pair has to be created for this PCA. The Root CA issues the new PCA certificate and the old certificate is added onto the CRL which is signed by the RCA. All related pseudonym certificates of ITS stations become invalid and new pseudonyms have to be requested. Table: 8.17.
- **Changing security format or protocol.** A new security protocol must be installed in order to be compliant with the standard or because some security vulnerabilities has been detected in the current one. Table: 8.18.

- **Changing certificate format.** The format of the certificates must be changed in order to be compliant with the standard and being able to communicate with other vehicles or infrastructure. Table: 8.19.
- **Changing crypto.** Another crypto system must be used instead of the current one which has revealed some security vulnerabilities. Table: 8.20.
- **End of lifetime of ITS-S** This use-case described the process to handle the end of lifetime of ITS-S. It is similar to "Revocation of ITS-S" use-case. However, it also deals with the destruction and recycling of the ITS-S. Table: 8.21.
- **End of lifetime of Root CA** If an operator of a Root CA goes out of business, the RCA-Cert cannot be used for the V2X communication as trust anchor. Affected LTCA, PCA and ITS-S have to be equipped with a new trustworthy RCA-Cert. Table: 8.22.
- **End of lifetime of LTCA** If an operator of a Long-term CA goes out of business, affected ITS-S have to be registered at another LTCA. Table: 8.23.
- **End of lifetime of PCA** If an operator of a Pseudonym CA goes out of business, affected ITS-S have to renew their PCs. Table: 8.24.
- **Revocation/Deletion of credentials** Valid vehicle credentials, may be needed to be revoked and deleted. Table: 8.25.
- **HSMFailure** If an HSM fails, eg because of an electronic component breakdown, then a new certified HSM has to be installed. The old credentials are revoked and new ones are issued and securely installed. Table: 8.26.

8.3 Operation Issues

Use cases defined in Section 8.2 are fully described in the next tables. In particular, the trigger raises the use cases are identified. According to each trigger, the process is described in details. Finally, security and business issues are discussed.

Use case label	5.1
Use case name	Installation of Root CA
Actors	RCA
Precondition	PKI domain not existing
Postcondition	New RCA certificate available
Trigger 0 (T0)	Creation of a new trust domain
Trigger 1 (T1)	RCA compromised
Trigger 2 (T2)	(Security) service provider goes out of business
Trigger 3 (T3)	Assembly of OBU
Trigger 4 (T4)	Manufacturing of HSM
Trigger 5 (T5)	Installation of OBU in car
Trigger 6 (T6)	New OEM/supplier enters the market
Trigger 7 (T7)	Backend connectivity
Process.RCA.RCABootstrapping	T0, T1, T2
	-Bootstrapping of RCA -Manual creation of CA key pair -Certificate creation and self signing of RCA-C -Storing private key of RCA in secure storage - RCA should not be accessible via network or internet
Process.RCA.RCACrossCertification	T0, T1, T2
	Optionally initiate cross-certification with foreign RCA
Process.RCA.ProvideRCACertificate	T3, T4, T5, T6, T7
	-Provide connectivity information of RCA via data service (e.g. IP address) -Provide Root-CA-Cert to requesting ITS-S via data service -Provide CRL containing revoked CA-Certs to requesting ITS-S via data service
Security discussion	RCA-CERT and CRL do not need to be handled confidential but it needs to be communicated in an authenticated and integrity protected way
	The private keys of the RCA (for signing and encryption) must be protected against misuse and arbitrary access.
Stakeholder/Business discussion	How many RCAs should be operated in Europe?

Table 8.1: Installation of the Root CA

Use case label	5.2
Use case name	Installation of Long-term CA
Actors	LTCA, RCA
Precondition	Long-term CA not existing
Postcondition	New LTCA certificate available
Trigger 1 (T0)	Creation of a new trust domain
Trigger 1 (T1)	PKI hacked
Trigger 2 (T2)	(Security) service provider goes out of business
Trigger 3 (T3)	Assembly of OBU
Trigger 4 (T4)	Manufacturing of HSM
Trigger 5 (T5)	Installation of OBU in car
Trigger 6 (T6)	New OEM/supplier enters the market
Trigger 7 (T7)	Backend connectivity
Process.LTCA.DownloadRCACert	T0, T1, T2
	-Get connectivity information of RCA (e.g. IP address) -Load Root-CA-Cert in a secure way -Make sure that Root-CA-Cert cannot be exchanged.
Process.LTCA.LTCABootstrapping	T0, T1, T2
	-Create new asymmetric key pairs for LTCA-Cert. -Provide LTCA-Cert request to RCA in a secure way (manual exchange of data as critical security process) -Receive LTCA-Cert issued by the RCA -Verify LTCA-Cert using Root-CA-Cert
Process.RCA.LTCABootstrapping	T0, T1, T2
	-Issue LTCA-Cert with private key of RCA (loading of RCA private key is a critical security process and should be done in a manual way) -Store Cert-ID of issued LTCA-Cert into a database (needed in case of revocation)

Use case label	5.2
Use case name	Installation of Long-term CA
Process.PCA.LTCABootstrapping	T0, T1, T2
	Notify PCAs that there is a new LTCA. Providing communication link information (e.g. IP address) and LTCA-Cert. If ITS-S request PCAs then the PCA has to know how the respective LTCA can be reached.
Process.ITS-S.LTCABootstrapping	T0, T1, T2
	ITS-S that use the new LTCA for registration store the LTCA-Cert into an internal storage
Process.LTCA.ProvideLTCACertificate	T3, T4, T5, T6, T7, T8
	-Provide connectivity information of LTCA (e.g. IP address) -Provide LTCA-Cert to requesting ITS-S or PCA
Security discussion	LTCA-Cert is public data and does not need be handled confidentially
	Private key of the LTCA must be protected against misuse and arbitrary access. The LTCA-Cert should contain a verification key that is only used to issue certificates and verify data. Additionally, the LTCA-Cert should contain an encryption key that is used to encrypt and decrypt data. The verification private key needs higher protection that the encryption private key.
Stakeholder/Business discussion	Who should operate the LTCA?
	Due to privacy issues, the PCA must not be operated by the same company / group of interests. If the LTPCA and PCA cooperate in a non defined way then pseudonyms can be linked to their long-term identity.

Table 8.2: Installation of the Long-term CA

Use case label	5.3
Use case name	Installation of Pseudonym CA
Actors	PCA, LTCA, RCA
Precondition	Pseudonym CA not existing
Postcondition	New PCA certificate available
Trigger 1 (T1)	PKI hacked
Trigger 2 (T2)	(Security) service provider goes out of business
Trigger 3 (T3)	Installation of OBU in car
Trigger 4 (T4)	Backend connectivity
Process.PCA.DownloadRCACert	T1, T2
	-Get connectivity information of RCA (e.g. IP address) -Load Root-CA-Cert in a secure way -Make sure that Root-CA-Cert cannot be changed.
Process.PCA.PCABootstrapping	T1, T2
	- Create new asymmetric key pairs for PCA-Cert. -Provide PCA-Cert request to RCA in a secure way (manual exchange of data as critical security process) -Receive PCA-Cert issued by the RCA (manual exchange of data as critical security process) -Verify PCA-Cert using Root-CA-Cert
Process.RCA.PCABootstrapping	T1, T2
	- Issue PCA-Cert with private key of RCA -Store Cert-ID of issued PCA-Cert (needed in case of revocation)
Process.ITS-S.PCABootstrapping	T1, T2
	Notify ITS-S that there is a new PCA. Providing communication link information (e.g. IP address) and PCA-Cert. If ITS-S would like to request PCs then the PCA has to be known.
Process.PCA.ProvidePCACertificate	T3, T4
	-Provide connectivity information of PCA (e.g. IP address) -Provide PCA-Cert to requesting ITS-S or LTCA

Use case label	5.3
Use case name	Installation of Pseudonym CA
Security discussion	PCA-Cert is public data and must not be handled confidentially
	Private key of the PCA must be protected against misuse and arbitrary access. The PCA-Cert should contain a verification key that is only used to issue certificates and verify data. Additionally, the PCA-Cert should contain an encryption key that is used to encrypt and decrypt data. The verification private key needs higher protection that the encryption private key.
Stakeholder/Business discussion	Who should operate the PCA?
	Due to privacy issues, the PCA must not be operated by the same company / group of interests as the one operating the LTCA. If the LTPCA and PCA cooperate in a non defined way then pseudonyms can be linked to their long-term identity. Conflicting interests should be revised over time.

Table 8.3: Installation of the Pseudonymous CA

Use case label	5.4
Use case name	First/Re-installation SW
Actors	OBU, installation-application server
Precondition	New-built vehicle OR after-market product
Postcondition	SW installed in the new OBU
Trigger 1 (T1)	Assembly of OBU
Process.ITS-S. SoftwareInstallation	T1
	-Software loaded into the OBU during the assembly -Installation Application Server provides Software that is installed on the OBU
Process.ITS-S. SoftwareReInstallation	Similar to use-case "Secure software update of OBU"
Security discussion	-The installation may be done by the supplier in an insecure way, but in a trusted environment? Is a secure SW installation needed (to protect malicious injection)? -If yes, trusted boot is needed? -What about the re-installation? Also secure or insecure, respectively?
StakeholderBusiness discussion	Supplier is responsible of this operation

Table 8.4: First/Re-Installation of the SW

Use case label	5.5
Use case name	First/Re-installation of the Root Certificate
Actors	HSM, RCA, Commercial Vehicle Check
Precondition	Station has no root certificate inside the HSM
Postcondition	New root certificate(s)
	Station is eqipped with a root certificate from a commonly trusted root CA
Trigger 1 (T1)	Assembly of OBU
Trigger 2 (T2)	PKI hacked
Trigger 3 (T3)	(Security) service provider goes out of business
Trigger 4 (T4)	HSM Failure / Broken
Process.LTCA. DownloadRCACert	T2, T3
	See Use-case “Installation Long-term CA” Table: 8.2.
Process.PCA. DownloadRCACert	T1, T2
	See Use-case “Installation Pseudonym CA” Table: 8.3
Process.ITS-S. DownloadRCACert	T1, T2, T3
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. HsmReplacement	T4
	See Use-case “Physical update - replacement of HSM” Table: 8.10
Security discussion	How to do it in the field? Is a credential beyond the RCA certificate needed, to validate authorization?
	If the root CA/key are compromised then a trustworthy channel is needed to update it otherwise the HSM has to be replace physically.
Stakeholder/Business discussion	Secure update of root certificates can be done with a OBD tool in a garage. Only an authorized station must be able to exchange the root certificate.

Table 8.5: First/Re-Installation of the Root Certificate

Use case label	5.6
Use case name	ITS-S initialization and registration
Actors	OBU, HSM, Installation Application Server, LTCA
Precondition	HSM is used in the VSS to store cryptographic credentials (i.e. private keys), accelerate cryptographic operations and acts as trust anchor inside the vehicle HSM not tampered
Postcondition	HSM registered at LTCA
	Pseudonyms loaded. Secure communication possible
Trigger 1 (T1)	Manufacturing of HSM
Trigger 2 (T2)	Assembly of OBU
Trigger 3 (T3)	Installation of OBU in car
Trigger 4 (T4)	HSM Failure / Broken
Process.ITS-S.DownloadRCACert	T1, T2, T3, T4
	-Get connectivity information of RCA (e.g. IP address) -Load Root-CA-Cert in a secure way. Possible mechanisms for a secure software update could be based on protocols specified by the Open Mobile Alliance (OMA) for the Device Management (DM). Further protocol proposals for remote firmware update are developed by Open Services Gateway initiative (OSGi), HIS Flash loader Specification, Secure Firmware Updates over the Air in Intelligent Vehicles (SFOTA) [2] and EVITA [3]. -Make sure that Root-CA-Cert cannot be changed. Optionally seal Root-CA-Cert with Platform Integrity Module (PIM) of VSS
Process.ITS-S.DownloadLTCACert	T1, T2, T3, T4
	-Load LTCA-Cert and information how LTCA can be connected (e.g. IP address) -Verify certificate with public key of Root-CA-Cert -Please consider that the certificate of the LTCA is updated due to change of permissions or foreseeable expiry of the current LTCA certificate.

Use case label	5.6
Use case name	ITS-S initialization and registration
Process.ITS-S.DownloadPCACert	T1, T2, T3, T4
	-Load PCA-Cert and information how PCA can be connected (e.g. IP address) -Verify certificate with public key of Root-CA-Cert -Please consider that the certificate of the PCA is updated due to change of permissions or foreseeable expiry of the current PCA certificate.
Process.ITS-S.LTCRequest	T1, T2, T3, T4
	-Register ITS Station at LTCA using a not further specified bootstrapping process. For example, the public key of the Devide Identity Key (IDK) of a Hardware Security Module (HSM) can be registered with a related IDK-ID (Canonical unique Id) at the LTCA. The cryptographic keys are generated by the HSM, and the the private key is never disclosed. -Using credentials from the bootstrapping process to sign a LTC request. Send encrypted LTC request to LTCA where ITS-S is registered. -Receive encrypted LTC response and decrypt it -Check that LTCA is not revoked using CRL requested from RCA -Verify received LTC using public key of LTCA-CA -Store LTC
Process.LTCA.LTCRequest	T1, T2, T3, T4
	- Receive encrypted LTC request from ITS-S. Use public key of LTCA-Cert to decrypt -Verifiy LTC request using credentials provided in the bootstapping process. For example, the Devide Identity Key (IDK) public key of a registered Hardware Security Module (HSM) can be used to verify the signed LTC request. -Sign LTC with private key of the LTCA -Encrypt response with private key of LTCA-Cert -Send encrypted LTC response to ITS-S

Use case label	5.6
Use case name	ITS-S initialization and registration
Process.ITS-S.PCRequest	T1, T2, T3, T4
	-Check whether enough pseudonym certificates are available for future time period and geographic location. -Create PC requests signed by the public key of the LTC. Add desired validity time and location validity to the request. -Encrypt PC request and send to PCA. Store locally related keys and credentials until response is received. -Receive encrypted PC response and decrypt it. If communication link was interrupted before pseudonym certificate response was received then retry request with stored credentials. In this case the PKI is able to response a previously processed request. -Verify received PCs using public key of PCA-CA -Check that PCA is not revoked using CRL requested from RCA -Store PCs in local storage. Create link between PC and private key handle. -Send an acknowledgment to the PCA
Process.PCA.PCRequest	T1, T2, T3, T4
	- Receive encrypted PC request from ITS-S -Decrypt PC request using public key of PCA-Cert -Create authorization request for PC and sent it to responsible LTCA. -Receive authorization response from LTCA for PC -Check that LTCA is not revoked using CRL requested from RCA -Create PC structure and add public key of requested PC into certificate structure - Sign PCs using the private key of PCA - Send encrypted pseudonym response to ITS-S - Store encrypted pseudonym response until an acknowledgment is received from the ITS-S

Use case label	5.6
Use case name	ITS-S initialization and registration
Process.LTCA.PCRequest	T1, T2, T3, T4
	- Receive authorization request from PCA -Check whether PCA is trusted and not revoked using CRL requested from RCA -Get public key of LTC from database -Verify authorization request using public key of LTC -Decide which number of PCs is allowed to be issued by the PCA -Decide which PC lifetime is allowed to be issued by the PCA -Send authorization response to PCA
Security discussion	Secure storage of IDK private key, Long-term private key, pseudonym private keys
	Certificate request and response is encrypted
Stakeholder/Business discussion	Is it necessary that RCA, LTCA and PCA are permanently accessible via communication link?
	Batch-Download of LTC into production side could be discussed. In this case no online connection is necessary to the LTCA in order to register ITS-S. This would mean that the long-term key pair is generated by the production site and the public key is added to a certificate and issued by the LTCA. When the ITS-S is initialized, the private and public key is loaded to the secure storage of the ITS-S. This implies additional trust in the production site as not only the ITS-S knows the LTC private key.
	[2] D. Nilsson, U. Larson: "Secure Firmware Updates over the Air in Intelligent Vehicles," Communications Workshops, 2008. ICC Workshops '08. IEEE International Conference on , vol., no., pp.380-384, 19-23 May 2008
	[3] S. Idrees et al.: EVITA D3.3: On-Board Protocols Specification. 2010. Deliverable of the EVITA-project (Reference number 224275, ICT-2007.6.2).

Table 8.6: ITS Initialization and Registration

Use case label	5.7
Use case name	Secure software update of OBU (by the Commercial Vehicle Check)
Actors	Buyer, Commercial Vehicle Check, Installation Application Server
Precondition	The software (not only the VSS) of the OBU is protected
	The Commercial Vehicle Check is allowed (i.e. certified) to update the OBU
	The new software is certified (signed) by a Installation Application Server. The certificate of this Installation Application Server is verifiable by VSS
Postcondition	The new software has been installed
Trigger 1 (T1)	Selling car to other country/PKI domain
Trigger 2 (T2)	New functions / protocols are to be installed
Trigger 3 (T3)	OEM/supplier goes out of business
Trigger 4 (T4)	New OEM/supplier enters the market
Trigger 5 (T5)	Backend connectivity
Process.ITS-S.OBUSoftwareUpdate	T1, T2, T3, T4, T5
	-Buyer goes to Commercial Vehicle Check (e.g. dealer or garage) -The Commercial Vehicle Check authenticates himself to the VSS -The Commercial Vehicle Check tries to update the OBU -The Platform Integrity Module of the VSS verifies the integrity of the update -The update is accepted and registered
Security discussion	Entire process must be secure
	Why only in garage? Possibly secure remote access, e.g., to OEM server?
Stakeholder/Business discussion	Does it make sense to say that a Commercial Vehicle Check (e.g. dealer) is allowed to change the format of the certificates?
	All Dealers must be allowed to repair a vehicle

Table 8.7: Secure Software Update of the OBU

Use case label	5.8
Use case name	Secure software update of VSS
Actors	Buyer, Commercial Vehicle Check, RCA, LTCA, PCA
Precondition	The VSS software is protected
	The Commercial Vehicle Check is allowed to update the VSS
	The new VSS software is certified (signed) by a Installation Application Server. The certificate of this server is issued by the RCA and therefore verifiable by VSS.
Postcondition	The new software has been installed
Trigger 1 (T1)	Selling car to other country/PKI domain
Trigger 2 (T2)	(Security) service provider goes out of business
Trigger 3 (T3)	New (security) service provider enters the market
Trigger 4 (T4)	Cryptosystem broken
Trigger 5 (T5)	HSM tampered
Trigger 6 (T6)	Significant vulnerability in specification
Trigger 7 (T7)	Backend connectivity
Process.ITS-S. VSSSoftwareUpdate	T1, T4, T5, T6
	-Buyer goes to Commercial Vehicle Check -The Commercial Vehicle Check authenticates himself to the VSS -The Commercial Vehicle Check tries to update the VSS -The Platform Integrity Module verifies the integrity of the update -The update is accepted and registered
Process.ITS-S. DownloadRCACert	T2, T3, T4, T5, T6
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. DownloadLTCACert	T2, T3, T4, T5, T6
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. DownloadPCACert	T2, T3, T4, T5, T6
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. LTCRequest	T2, T3, T4, T5, T6
	See Use-case “ITS-S initialization and registration” Table: 8.6

Use case label	5.8
Use case name Process.ITS-S. PCRequest	Secure software update of VSS T7
	See Use-case “ITS-S initialization and registration” Table: 8.6
Security discussion	Why only in garage? Possibly secure remote access, e.g., to OEM server?
Stakeholder/Business discussion	Does it make sense to say that a Commercial Vehicle Check is allowed to change the format of the certificates?
	All Commercial Vehicle Check must be allowed to repair a vehicle

Table 8.8: Secure Software Update of the VSS

Use case label	5.9
Use case name	Physical update / replacement of OBU including the HSM
Actors	Buyer, Commercial Vehicle Check, RCA, LTCA, PCA
Precondition	Value chain is structured according to actor and stakeholder list.
	Physical replacement is done locally (interoperability not taken into account)
	This scenario supposes that it is possible to make a link between the long term certificate and the owner of the vehicle. Therefore it is necessary to change the long term certificate
Postcondition	The new OBU has been installed
	ITS-S is operational for V2X communication
	New credentials have been installed
Trigger 1 (T1)	Repair / non-periodic inspection
Trigger 2 (T2)	HSM Faillure / Broken
Trigger 3 (T3)	Cryptosystem broken
Trigger 4 (T4)	HSM tampered
Trigger 5 (T5)	Vulnerability in OBU/VSS/HSM
Trigger 6 (T6)	Backend connectivity
Process.ITS-S. OBUReplacement	T1, T2, T3, T4, T5
	-Buyer goes to Commercial Vehicle Check in order to replace his OBU -Commercial Vehicle Check contacts LTCA to revoke the long term certificate of the V2X system (see use case "Revocation of ITS-S") -Commercial Vehicle Check replaces the old OBU with a new one
Process.ITS-S. DownloadRCACert	T1, T2, T3, T4, T5
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. DownloadLTCACert	T1, T2, T3, T4, T5
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. DownloadPCACert	T1, T2, T3, T4, T5
	See Use-case “ITS-S initialization and registration” Table: 8.6

Use case label	5.9
Use case name **Process.ITS-S. LTCRequest**	Physical update / replacement of OBU including the HSM T1, T2, T3, T4, T5
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. PCRequest	T6
	See Use-case “ITS-S initialization and registration” Table: 8.6
Security discussion	Entire process must be secure
	What if the connection between the LTCA-issued certificate and owner is not possible, e.g., not direct but only through an additional entity?
Stakeholder/Business discussion	Commercial Vehicle Check (Dealer) must be certified by the OEM

Table 8.9: Physical Update/Replacement of the OBU

Use case label	5.10
Use case name	Physical update/replacement of HSM
Actors	HSM, Commercial Vehicle Check, (Destruction site?), RCA, LTCA, PCA
Precondition	Vehicle has valid credentials
	Vehicle has outdated HSM version
Postcondition	Vehicle is equipped with a new HSM
Trigger 1 (T1)	Repair/non-periodic inspection
Trigger 2 (T2)	HSM failure/Broken
Trigger 3 (T3)	Cryptosystem broken
Trigger 4 (T4)	HSM tampered
Trigger 5 (T5)	Significant vulnerability in specification
Trigger 6 (T6)	Vulnerability in OBU/VSS/HSM
Trigger 7 (T7)	End of vehicle lifetime
Trigger 8 (T8)	New functions/protocols are to be installed (if HW implication(s))
Trigger 9 (T9)	Backend connectivity
Process.ITS-S.HsmReplacement	T1, T2, T3, T4, T5, T6, T8
	-Vehicle is brought to a Commercial Vehicle Check which performs the physical update/replacement of the HSM -Commercial Vehicle Check contacts LTCA to revoke the long term certificate of the V2X system (see use case "Revocation of ITS-S") This process could happen during a non-periodic inspection (detection of problem)
Process.ITS-S.HsmReplacement	T7
	-Vehicle is brought to a Commercial Vehicle Check which performs the physical disassembly of the HSM -Commercial Vehicle Check contacts LTCA to revoke the long term certificate of the V2X system (see use case "Revocation of ITS-S")
Process.ITS-S.DownloadRCACert	T1, T2, T3, T4, T5, T6, T8
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S.DownloadLTCACert	T1, T2, T3, T4, T5, T6, T8
	See Use-case “ITS-S initialization and registration” Table: 8.6

Use case label	5.10
Use case name	Physical update/replacement of HSM
Process.ITS-S.DownloadPCACert	T1, T2, T3, T4, T5, T6, T8
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S.LTCRequest	T1, T2, T3, T4, T5, T6, T8
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S.PCRequest	T9
	See Use-case “ITS-S initialization and registration” Table: 8.6
Security discussion	Ensure in Process.ITS-S.HsmReplacement a secure destruction of the HSM to ensure all critical data (private keys) are erased and cannot be misused.
Stakeholder/Business discussion	The Commercial Vehicle Check has to be certified to access the HSM
	Recycling of HSM?
	As the HSM is integrated into the OBU and as the OEM objective is to replace what is the most accessible, then this use case may not happen and be done into the use case "replacement of OBU"

Table 8.10: Physical Replacement of HSM

Use case label	5.11
Use case name	Refill of pseudonym certificates
Actors	LTCA, PCA, VSS, HSM, Commercial Vehicle Check
Precondition	ITS station is registered at PKI with Device Identity Key (IDK) of HSM
	ITS station has valid long-term certificate and key pair
	-ITS station can communicate with PKI. -Communication channel does not have to be secure. Transport protection of certificate request and response is ensured by packet signature and encryption. -Communication channel does not have to be stable and can be ad-hoc based. Establishing of a session between ITS station and PKI is not necessary.
	ITS station has valid address of PKI server. (e.g. IP-address and port number)
	ITS station has not previously requested maximum number of pseudonym certificates for a specific time period
Postcondition	ITS station has sufficient valid pseudonym certificates for upcoming time period(s)
Trigger 1 (T1)	Back-end connectivity to PKI
Trigger 2 (T2)	OBU of ITS station is low on pseudonym certificates
Trigger 3 (T3)	Vehicle at annual inspection
Trigger 4 (T4)	Roaming between PCA or change of PKI domain
Trigger 5 (T5)	Buying a new car with V2X system
Trigger 6 (T6)	Buying an after-market V2X system
Trigger 7 (T7)	ITS station at non-periodic inspection
Trigger 8 (T8)	(Security) service provider (i.e. CA operator) goes out of business
Trigger 9 (T9)	PKI hacked. Assuming the root CA is not compromised
Process.ITS-S. DownloadRCACert	T1, T2, T3, T4, T5, T6, T7, T8, T9
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. PCRequest	T1, T2, T3, T4, T5, T6, T7, T8, T9
	Use Commercial Vehicle Check as gateway to get communication link to the PCA See Use-case “ITS-S initialization and registration” Table: 8.6
Process.PCA. PCRequest	T1, T2, T3, T4, T5, T6, T7, T8, T9
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.LTCA. PCRequest	T1, T2, T3, T4, T5, T6, T7, T8, T9
	See Use-case “ITS-S initialization and registration” Table: 8.6

Use case label	5.11
Use case name	Refill of pseudonym certificates
Process.ITS-S. NoPCs	T4
	-ITS-S uses only certificates that are valid in the new PKI domain. Invalid pseudonyms should not be deleted before they expire, as they might be valid in other domains. -If no valid pseudonym certificate available, stop signing and wait until back-end communication is available.
Security discussion	Memory requirement on ITS station
	Memory requirement at PKI
	Privacy protection at PKI
	Revocation of PKI entities
	Lifetime of pseudonym certificate
Stakeholder/Business discussion	No manually interaction of PKI operator necessary

Table 8.11: Refill of Pseudonym Certificates

Use case label	5.12
Use case name	Update of Long Term Certificate
Actors	PKI, VSS, HSM, Commercial Vehicle Check
Precondition	VSS has no long-term certificate
	Long-term certificate of ITS station is expired or expires soon
	ITS station has valid root certificate
	-ITS station can communicate with PKI. -Communication channel does not have to be secure. Transport protection of certificate request and response is ensured by packet signature and encryption. -Communication channel does not have to be stable and can be ad-hoc based. Establishing of a session between ITS station and PKI is not necessary
	ITS station has valid address of PKI servers. (e.g. IP-address and port number)
Postcondition	Long-term certificate of ITS station is valid and has future expiry timestamp
	Long-term certificate is permitted to refill pseudonym certificates
Trigger 1 (T1)	Installation of OBU in car
Trigger 2 (T2)	Selling car to other country/PKI domain
Trigger 3 (T3)	Vehicle at annual inspection or ITS station at non-periodic inspection
Trigger 4 (T4)	New functions / protocols are to be installed
Trigger 5 (T5)	LTCA service provider goes out of business or PKI hacked (assuming the root CA is not compromised)
Trigger 6 (T6)	Backend connectivity
Process.ITS-S. LTCRequest	T1, T3, T4, T5
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.LTCA. DeactivateRegistration	T2, T5
	See Use-case “Revocation of ITS-S” Table: 8.14
Process.LTCA. DeactivateLTC	T2, T5
	See Use-case “Revocation of ITS-S” Table: 8.14

Use case label	5.12
Use case name Process.ITS-S. DeleteLTCACert	Update of Long Term Certificate T5
	See use-case “Revocation of Long-term CA” Table: 8.16
Process.ITS-S. DownloadLTCACert	T5
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. PCRequest	T6
	See Use-case “ITS-S initialization and registration” Table: 8.6
Security discussion	
Stakeholder/Business discussion	

Table 8.12: Update of Long Term Certificate

Use case label	5.13
Use case name	Misbehavior Detection / Reporting of ITS station
Actors	PKI, OBU / VSS and HSM of ITS station
Precondition	-ITS station is able to receive V2X messages from neighboring ITS stations -Authenticity and integrity of message sender and message content can be verified at ITS station -ITS station can communicate with PKI. -Communication channel must not be secure. Transport protection of certificate request and response is ensured by packet signature and encryption. -Communication channel can be unstable and ad-hoc based. Establishing of a session between ITS station and PKI is not necessary.
Postcondition	Faulty or malicious behavior of sending ITS station detected
Trigger 1 (T1)	HSM Failure / Broken
Trigger 2 (T2)	Cryptosystem broken
Trigger 3 (T3)	HSM tampered
Trigger 4 (T4)	Significant vulnerability in specification
Trigger 5 (T5)	Vulnerability in OBU/VSS/HSM
Trigger 6 (T6)	ITS station receives bogus messages
Trigger 7 (T7)	Backend connectivity
Trigger 8 (T8)	PKI receives misbehavior report
Process.ITS-S. DetectOBUManipulation	T1, T2, T3, T4 or T5
	-ITS station is manipulated or defective and sends V2X messages with faulty content -Triggers use case physical/sfotware update

Use case label	5.13
Use case name	Misbehavior Detection / Reporting of ITS station
Process.ITS-S. DetectMisbehavior	T6
	-Manipulated or defective ITS station sends messages that are detected as misbehavior at message receiving ITS stations in communication range -VSS of receiving ITS station collects information from suspicious sender (i.e. suspicious message content and sender's pseudonym certificates) -VSS of receiving ITS station evaluates relevance of misbehavior according to its context -VSS of receiving ITS station creates misbehavior reports for suspicious ITS stations and sign them with the currently used pseudonym certificate -Misbehavior reports are stored in a non volatile memory in LIFO manner inside the VSS
Process.ITS-S. SendMisbehaviorReport	T7
	-ITS station send all misbehavior reports to central misbehavior report evaluation entity at PKI -Misbehavior report storage inside VSS is cleared
Process.PKI. ProcessMisbehaviorReport	T8
	-Misbehavior reports are collected and evaluated based on their content. -Long-term ID of faulty and malicious ITS stations is used to revoke the ITS station. See use-case Revocation of ITS-S -As soon as correct functionality of OBU, VSS and HSM is ensured, the long-term certificate of the ITS station is activated.

Use case label	5.13
Use case name	Misbehavior Detection / Reporting of ITS station
Security discussion	-Maximum size of misbehavior report storage inside VSS must be defined -Resolution of pseudonym to long-term ID is not defined in the PKI. This resolution concerns privacy protection measures
StakeholderBusiness discussion	No manually interaction of PKI operator necessary

Table 8.13: Misbehavior Detection, Reporting, Evaluation, or Reaction

Use case label	5.14
Use case name	Revocation of ITS station
Actors	LTCA, VSS, HSM, Commercial Vehicle Check
Precondition	ITS station has valid long-term certificate
Postcondition	ITS station is not able to request new pseudonym certificates from PKI
Trigger 1 (T1)	End of vehicle lifetime
Trigger 2 (T2)	Change PKI domain (ITS station is registered at other LTCA), incl. car sale, owner relocation, etc.
Trigger 3 (T3)	Vulnerability of VSS, OBU or HSM
Trigger 4 (T4)	Misbehavior detection detects possible failure or tampered VSS, OBU or HSM
Process.LTCA.DeactivateRegistration	T1, T2
	-Commercial Vehicle Check or misbehavior detection entity sends deactivation request to LTCA -LTCA marks the affected ITS-S registration as invalid. -Subsequent requests for new LTC or LTC updates from this ITS station are rejected
Process.LTCA.DeactivateLTC	T1, T2, T3, T4
	-LTCA marks the affected long-term certificate as invalid -Subsequent pseudonym certificate requests from this ITS station are rejected -If the inactive long-term certificate is expired then the related entry is deleted from the LTCA database in order to save resources at the CA -If the ITS-S is re-registered and the LTC is still valid, then the deactivated LTC can be reactivated at the LTCA. Otherwise a new long-term certificate must be issued for the ITS-S.
Process.LTCA.DeletePCAuthorizations	T1, T2, T3, T4
	See Use-case “Revocation of Pseudonym CA” Table: 8.17

Use case label	5.14
Use case name	Revocation of ITS station
Security discussion	Communication between Commercial Vehicle Check and PKI must be secured (authenticity, integrity, confidentiality, privacy)
	The VSS software of the ITS station should make sure that only one long-term certificate is stored and used by the HSM.
	Measures for remote integrity check and functional security checking has to be defined
Stakeholder/Business discussion	

Table 8.14: Revocation of ITS Station

Use case label	5.15
Use case name	Revocation of Root CA
Actors	RCA, LTCA, PCA, VSS, HSM
Precondition	Revoked root CA certificate is used in VSS of ITS station as trust anchor
Postcondition	New trusted PKI exists
	VSS software of ITS station is updated
Trigger 1 (T1)	RCA private signing key compromised, detected RCA misbehavior or other RCA breach
Trigger 2 (T2)	Vehicle is in garage for service
Trigger 3 (T3)	Vehicle is connected with OEM for software update (remote firmware update)
Process.RCA. RCABootstrapping	T1
	- Bootstrapping of RCA - Manual creation of CA key pair - Certificate creation and self signing of RCA-C - Storing private key of RCA in secure storage - Make access to private key only possible if other CAs should be issued. RCA should not be accessible via network or internet
Process.RCA. RevokeRCACert	T1
	- Set old RCA-Cert on CRL that is signed with the new RCA-Cert private key - Delete CA-Certs that are issued by the revoked RCA - Trigger update of LTCA certificates and long-term certificates that are related to the revoked root CA. Revocation of affected LTCA certificates is not necessary when LTCA not hacked but new certificates should be created that are issued by the new root that is trusted. Afterwards, all long-term certificates of affected ITS stations have to be updated. - Trigger update of PCA certificates that are issued by the revoked root CA. Revocation of affected PCA certificates is not necessary when PCA not hacked but new certificates should be created that are issued by the new root that is trusted.
Process.RCA. RevokeCrossRCACert	T1
	- Revoke cross-certificates. Set old RCA-Cross-Certs on CRL that is signed with the new RCA-Cert private key

Use case label	5.15
Use case name	Revocation of Root CA
Process.RCA. ProvideRCACertificate	T1, T2, T3
	- Provide connectivity information of new RCA via data service (e.g. IP address) - Proide new Root-CA-Cert to requesting ITS-S via data service - Provide CRL containing revoked CA-Certs to requesting ITS-S via data service
Process.LTCA. DownloadRCACert	T1
	See Use-case “Installation Long-term CA” Table: 8.2
Process.LTCA. LTCABootstrapping	T1
	See Use-case “Installation Long-term CA” Table:8.2
Process.LTCA. DeletePCACert	T1
	- Delete PCA-Certs at LTCA that are issued by the revoked RCA-Cert
Process.RCA. LTCABootstrapping	T1
	See Use-case “Installation Long-term CA” Table: 8.2
Process.PCA. DownloadRCACert	T1
	See Use-case “Installation Pseudonym CA” Table: 8.3
Process.PCA. PCABootstrapping	T1
	See Use-case “Installation Pseudonym CA” Table: 8.3
Process.PCA. DeleteLTCACert	T1
	- Delete LTCA-Certs at PCA that are issued by the revoked RCA-Cert
Process.RCA. PCABootstrapping	T1
	See Use-case “Installation Pseudonym CA” Table: 8.3
Process.ITS-S. DownloadRCACert	T2, T3
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. DeleteLTCACert	T2, T3
	- Delete LTCA-Certs from VSS database and HSM that are issued by the revoked RCA-Cert
Process.ITS-S. DeletePCACert	T2, T3
	- Delete PCA-Certs from VSS database and HSM that are issued by the revoked RCA-Cert
Process.ITS-S. DeleteLTC	T2, T3
	- Delete LTC from VSS database and HSM that is issued by the LTCA that is issued by the revoked RCA-Cert
Process.ITS-S. LTCRequest	T2, T3
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.LTCA. LTCRequest	T2, T3
	See Use-case “ITS-S initialization and registration” Table: 8.6

Use case label	5.15
Use case name	Revocation of Root CA
Process.ITS-S. DeletePC	T2, T3
	- Delete PCs at ITS-S that are issued by the PCA that is issued by the revoked RCA-Cert
Process.ITS-S. PCRequest	T2, T3
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.PCA. PCRequest	T2, T3
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.LTCA. PCRequest	T2, T3
	See Use-case “ITS-S initialization and registration” Table: 8.6
Security discussion	Using the shell model for revocation [128], all issued certificates have to be renewed after revocation of the root CA. But the issued CA certificates must not be added on the CRL as the issuer (RCA) is already added to the CRL.
	Certificate request and response is encrypted
Stakeholder/Business discussion	PKI operator is involved to manually renew root CA credentials. Interaction with other PKI providers is probably necessary if cross-certificates exist.
	OEM has to update all ITS stations that have installed the revoked root certificate as trust anchor
	If the Root CA was hacked and the Root CA certificate is compromised, additional action would be necessary since it might take years to update the certificate in all cars.

Table 8.15: Revocation of the Root CA

Use case label	5.16
Use case name	Revocation of Long-term CA
Actors	RCA, LTCA, PCA, VSS, HSM
Precondition	ITS station has valid root certificate
	-ITS station can communicate with PKI. -Communication channel does not have to be secure. Transport protection of certificate request and response is ensured by packet signature and encryption. -Communication channel does not have to be stable and can be ad-hoc based. Establishing of a session between ITS station and PKI is not necessary
	-ITS station has valid address of PKI servers. (e.g. IP-address and port number)
Postcondition	VSS of ITS station has latest CRL that contains the ID of the revoked LTCA.
Trigger 1 (T1)	PKI hacked, assuming the root CA is not compromised
Trigger 2 (T2)	Refill of pseudonyms fail due to invalid signature of pseudonym request
Trigger 3 (T3)	Back-end connectivity to PKI and time of next CA CRL download is reached
Process.RCA. RevokeLTCACert	T1
	Add LTCA certificate ID to CA CRL and sign this list with RCA's private key
Process.RCA. DeleteIssuedLTCACert	T1
	Delete revoked LTCA-Cert from RCA database
Process.RCA. ProvideRCACertificate	T1
	See Use-case "Installation Root CA" Table: 8.1
Process.LTCA. LTCABootstrapping	T1
	See Use-case "Installation Long-term CA" Table: 8.2
Process.RCA. LTCABootstrapping	T1
	See Use-case "Installation Long-term CA" Table: 8.2
Process.PCA. DownloadRCACert	T1
	Download regularly CRL from Root CA and check whether revoked LTCAs are involved in pseudonym requests. See Use-case "Installation Pseudonym CA" Table: 8.3
Process.PCA. DeleteLTCACert	T1
	Delete revoked LTCA-Cert from database of PCA

Use case label	5.16
Use case name	Revocation of Long-term CA
Process.ITS-S. DownloadRCACert	T1, T2, T3
	ITS station downloads the CA CRL from trusted RCA. If LTCA is hacked then the ITS station should not download the CRL from this entity. See Use-case “ITS-S initialization and registration” Table:8.6
Process.ITS-S. DeleteLTCACert	T1, T2, T3
	Delete revoked LTCA-Cert from database of VSS and HSM
Process.ITS-S. DownloadLTCACert	T1, T2, T3
	Download LTCA-Cert from new LTCA that is not revoked. New LTCA must not listed in the CRL See Use-case “ITS-S initialization and registration” Table:8.6
Process.ITS-S. DeleteLTC	T1, T2, T3
	-ITS-S checks whether its long-term certificate is issued by a revoked LTCA -Delete LTC from VSS database and HSM that is issued by the revoked LTCA. The HSM should delete also related private and public keys from the secure storage.
Process.ITS-S. LTCRequest	T1, T2, T3
	See Use-case “ITS-S initialization and registration”
Security discussion	Using the shell model for revocation [128], all issued long-term certificates are useless after revocation of the LTCA
Stakeholder/Business discussion	The PKI operator is involved to manually revoke LTCA certificate and create new LTCA credentials. A new LTCA certificate will be issued by the root CA only if all requirements of the respective policy are fulfilled.

Table 8.16: Revocation of the Long Term CA

Use case label	5.17
Use case name	Revocation of Pseudonym CA
Actors	RCA, LTCA, PCA, VSS, HSM
Precondition	ITS station has valid root certificate
	ITS station can communicate with PKI. - Communication channel does not have to be secure. Transport protection of certificate request and response is ensured by packet signature and encryption. - Communication channel does not have be stable and can be ad-hoc based. Establishing of a session between ITS station and PKI is not necessary.
	ITS station has valid address of PKI servers. (e.g. IP-address and port number)
Postcondition	VSS of ITS station has latest CRL that contains the ID of the revoked PCA.
	Messages from other ITS stations that are signed with pseudonyms issued by the revoked PCA are not accepted.
	Stored pseudonym credentials that are issued by the revoked PCA are deleted
Trigger 1 (T1)	PKI hacked
Trigger 2 (T2)	Certificate from neighbor ITS station with unknown CRL series is received
Trigger 3 (T3)	Back-end connectivity to PKI and time of next CA CRL download is reached.
Process.RCA. RevokePCACert	T1
	Add PCA certificate ID to CA CRL and sign this list with RCA's private key
Process.RCA. DeleteIssuedPCACert	T1
	Delete revoked PCA-Cert from RCA database
Process.RCA. ProvideRCACertificate	T1
	See Use-case "Installation Root CA" Table: 8.1
Process.LTCA. DownloadRCACert	T1
	Download regularly CRL from Root CA and check whether revoked PCAs are involved in pseudonym authorization requests. See Use-case "Installation Long-term CA" Table: 8.2
Process.LTCA. DeletePCACert	T1
	Delete revoked PCA-Cert from database of LTCA
Process.LTCA. DeletePCAuthorizations	T1
	Delete information about PC authorizations from database of LTCA.

Use case label	5.17
Use case name	Revocation of Pseudonym CA
Process.PCA. PCABootstrapping	T1
	See Use-case “Installation Pseudonym CA” Table: 8.3
Process.ITS-S. DownloadRCACert	T1, T2, T3
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. DeletePCACert	T1, T2, T3
	See Use-case “Revocation of Root CA” Table: 8.15
Process.ITS-S. DownloadPCACert	T1, T2, T3
	Download PCA-Cert from new PCA that is not revoked. New PCA must not listed in the CRL See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. DeletePC	T1, T2, T3
	ITS-S checks whether its pseudonym certificates are issued by a revoked PCA - Delete PCs that are issued by the revoked PCA from VSS database and HSM. The HSM should delete related private and public keys from secure storage.
Process.ITS-S. PCRequest	T1, T2, T3
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.PCA. PCRequest	T1, T2, T3
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.LTCA. PCRequest	T1, T2, T3
	See Use-case “ITS-S initialization and registration” Table: 8.6
Security discussion	Using the shell model for revocation [128], all issued pseudonym certificates are useless after revocation of the PCA
Stakeholder/Business discussion	The PKI operator is involved to manually revoke PCA certificate and create new PCA credentials. A new PCA certificate will be issued by the root CA only if all requirements of the respective policy are fulfilled.

Table 8.17: Revocation of the Pseudonymous CA

Use case label	5.18
Use case name	Changing security format/protocol
Actors	Buyer, Commercial Vehicle Check, RCA, LTCA, PCA
Precondition	The VSS software is protected
	The Commercial Vehicle Check is allowed to update the VSS
Postcondition	The new security format/protocol has been installed
Trigger 1 (T1)	New functions / protocols are to be installed
Trigger 2 (T2)	Cryptosystem broken
Trigger 3 (T3)	Significant vulnerability in specification
Trigger 4 (T4)	Vulnerability in OBU/VSS/HSM
Trigger 5 (T5)	Backend connectivity
Process.RCA. SoftwareUpdate	T2, T3, T4
	- Update software of Root CA - Trigger revocation of RCA-Cert if keys or certificates are affected. See Use-case "Revocation of Root CA" Table: 8.15
Process.LTCA. SoftwareUpdate	T2, T3, T4
	Update software of LTCA Trigger revocation of LTCA-Cert if keys or certificates are affected. See Use-case "Revocation of LTCA" Table: 8.16
Process.PCA. SoftwareUpdate	T2, T3, T4
	Update software of PCA - Trigger revocation of PCA-Cert if keys or certificates are affected. See Use-case "Revocation of PCA" Table: 8.17
Process.ITS-S. VssSoftwareUpdate	T1, T2, T3, T4
	See Use-case "Secure software update of VSS" Table: 8.8
Process.ITS-S. DownloadRCACert	T1, T2, T3, T4
	See Use-case "ITS-S initialization and registration" Table: 8.6
Process.ITS-S. DownloadLTCACert	T1, T2, T3, T4
	See Use-case "ITS-S initialization and registration" Table: 8.6
Process.ITS-S. DownloadPCACert	T1, T2, T3, T4
	See Use-case "ITS-S initialization and registration" Table: 8.6
Process.ITS-S. LTCRequest	T1, T2, T3, T4
	See Use-case "ITS-S initialization and registration" Table: 8.6
Process.ITS-S. PCRequest	T5
	See Use-case "ITS-S initialization and registration" Table: 8.6
Security discussion	

Use case label	5.18
Use case name	Changing security format/protocol
Stakeholder/Business discussion	Does it make sense to say that a Dealer is allowed to change the format of the certificates?
	All Dealers must be allowed to repair a vehicle
	What is if the cryptographic mechanisms are modified that are used for the V2V communication? For instance, ECDSA is replaced by a new signature scheme that is resistant to quantum computing attacks?

Table 8.18: Changing security format protocol

Use case label	5.19
Use case name	Changing certificate format
Actors	OBU, Buyer, Dealer, Certificate authority
Precondition	The VSS software is protected
	The Dealer is allowed (i.e. certified) to change the format of the certificates
Postcondition	New certificate format installed
Trigger 1 (T1)	Selling car to other country/PKI domain
Trigger 2 (T2)	New functions / protocols are to be installed
Trigger 3 (T3)	Cryptosystem broken
Trigger 4 (T4)	Significant vulnerability in specification
Trigger 5 (T5)	Vulnerability in OBU/VSS/HSM
Trigger 6 (T6)	Backend connectivity
Process.RCA. SoftwareUpdate	T2, T3, T4
	See Use-case “Changing security format/protocol” Table: 8.18
Process.LTCA. SoftwareUpdate	T2, T3, T4
	See Use-case “Changing security format/protocol” Table: 8.18
Process.PCA. SoftwareUpdate	T2, T3, T4
	See Use-case “Changing security format/protocol” Table: 8.18
Process.ITS-S. VssSoftwareUpdate	T1, T2, T3, T4, T5
	See Use-case “Secure software update of VSS” Table: 8.8
Process.ITS-S. DownloadRCACert	T1, T2, T3, T4, T5
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. DownloadLTCACert	T1, T2, T3, T4, T5
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. DownloadPCACert	T1, T2, T3, T4, T5
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. LTCRequest	T1, T2, T3, T4, T5
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. PCRequest	T6
	See Use-case “ITS-S initialization and registration” Table: 8.6

Use case label	5.19
Use case name	Changing certificate format
Security discussion	Interoperability between different versions of certificates
Stakeholder/Business discussion	What is if the cryptographic mechanisms are modified that are used for the V2V communication? For instance, ECDSA is replaced by a new signature scheme that is resistant to quantum computing attacks?

Table 8.19: Changing the certificate format

Use case label	5.20
Use case name	Changing crypto
Actors	OBU, Buyer, Commercial Vehicle Check, RCA, LTCA, PCA
Precondition	The VSS software is protected
	The Commercial Vehicle is allowed (i.e. certified) to change the crypto
Postcondition	New certificate format installed
Trigger 1 (T1)	Cryptosystem broken
Trigger 2 (T2)	Significant vulnerability in specification
Trigger 3 (T3)	Vulnerability in OBU/VSS/HSM
Trigger 4 (T4)	Back-end connectivity
Process.RCA. SoftwareUpdate	T1, T2
	See Use-case “Changing security format/protocol” Table: 8.18
Process.LTCA. SoftwareUpdate	T1, T2
	See Use-case “Changing security format/protocol” Table: 8.18
Process.PCA. SoftwareUpdate	T1, T2
	See Use-case “Changing security format/protocol” Table: 8.18
Process.ITS-S. VssSoftwareUpdate	T1, T2, T3
	- The Commercial Vehicle Check tries to modify the configuration of the VSS for it to use another crypto - The platform Integrity Module verifies the integrity of the new crypto system. See Use-case “Secure software update of VSS” Table: 8.8
Process.ITS-S. DownloadRCACert	T1, T2, T3, T4
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. DownloadLTCACert	T1, T2, T3, T4
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. DownloadPCACert	T1, T2, T3, T4
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. LTCRequest	T1, T2, T3, T4
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S. PCRequest	T5
	See Use-case “ITS-S initialization and registration” Table: 8.6

Use case label	5.20
Use case name	Changing crypto
Security discussion	
Stakeholder/Business discussion	For the second process, the LTC is supposed to identify the crypto system. Therefore if this last one is changed.
	The LTC must be changed too
	What is if the cryptographic mechanisms are modified that are used for the V2V communication? For instance, ECDSA is replaced by a new signature scheme that is resistant to quantum computing attacks?

Table 8.20: Changing crypto

Use case label	5.21
Use case name	End of lifetime of ITS-S
Actors	LTCA, VSS, HSM, Commercial Vehicle Check
Precondition	Vehicle has valid long-term certificate and keypair
Postcondition	Vehicle has no valid credentials
Trigger 1 (T1)	End of vehicle lifetime
Trigger 2 (T2)	HSM failure/broken
Trigger 3 (T3)	HSM tampered
Trigger 4 (T4)	Selling car to other country/PKI domain
Process.LTCA. DeactivateRegistration	T1, T2, T3, T4
	See Use-case “Revocation of ITS station” Table: 8.14
Process.LTCA. DeactivateLTC	T1, T2, T3, T4
	See Use-case “Revocation of ITS station” Table: 8.14
Process.LTCA. DeletePCAuthorizations	T1, T2, T3, T4
	See Use-case “Revocation of Pseudonym CA” Table: 8.17
Security discussion	How to do it in the field? Self-destruction? Internal action? If not, then secure connection to backend is needed.
	If pseudonyms are expired and registration of HSM deactivated then the HSM cannot be used to particiate in V2X communication. It is not necessary to delete the keys from the HSM as the ITS-S is deactivated at the LTCA.
Stakeholder/Business discussion	If the the registration of the HSM is deativated, then in order to free memory in the VSS and HSM, the keys should be deleted. But this is not necessary form security point of view.

Table 8.21: End of ITS-S Lifetime

Use case label	5.22
Use case name	End of lifetime of Root CA
Actors	RCA, LTCA, PCA, VSS, HSM
Precondition	Root CA certificate expires and is used in VSS as trust anchor
Postcondition	Another RCA exisits that can be used by LTCA, PCA and ITS-S
	VSS software of ITS station is updated
Trigger 1 (T1)	(Security) service provider goes out of business
Trigger 2 (T2)	Backend connectivity
Process.RCA.StopProvidingRCACert	T1
	-Shut down data service. Stop providing RCA-Cert and CRL -Destroy private key of RCA-Cert -Revocation of RCA-Cert should not be necessary as RCA-Cert expires -Cross-Certifications with other RCAs expire automatically with the expiration of the RCA-Cert
Process.LTCA.DownloadRCACert	T1
	See Use-case “Installation Long-term CA” Table: 8.2
Process.LTCA.LTCABootstrapping	T1
	See Use-case “Installation Long-term CA” Table: 8.2
Process.LTCA.DeletePCACert	T1
	Delete PCA-Certs at LTCA that are issued by the outdated RCA-Cert
Process.RCA.LTCABootstrapping	T1
	See Use-case “Installation Long-term CA” Table: 8.2
Process.PCA.DownloadRCACert	T1
	See Use-case “Installation Pseudonym CA” Table: 8.3
Process.PCA.PCABootstrapping	T1
	See Use-case “Installation Pseudonym CA” Table: 8.3
Process.PCA.DeleteLTCACert	T1
	Delete LTCA-Certs at PCA that are issued by the outdated RCA-Cert
Process.RCA.PCABootstrapping	T1
	See Use-case “Installation Pseudonym CA” Table: 8.3

Use case label	5.22
Use case name	End of lifetime of Root CA
Process.ITS-S.DownloadRCACert	T1, T2
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S.DeleteLTCACert	T1, T2
	Delete LTCA-Certs from VSS database and HSM that are issued by the outdated RCA-Cert
Process.ITS-S.DeletePCACert	T1, T2
	Delete PCA-Certs from VSS database and HSM that are issued by the outdated RCA-Cert
Process.ITS-S.DeleteLTC	T1, T2
	Delete LTC from VSS database and HSM that is issued by the LTCA that is issued by the outdated RCA-Cert
Process.ITS-S.LTCRequest	T1, T2
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.LTCA.LTCRequest	T1, T2
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S.DeletePC	T1, T2
	Delete PCs at ITS-S that are issued by the PCA that is issued by the outdated RCA-Cert
Process.ITS-S.PCRequest	T1, T2
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.PCA.PCRequest	T1, T2
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.LTCA.PCRequest	T1, T2
	See Use-case “ITS-S initialization and registration” Table: 8.6

Use case label	5.22
Use case name	End of lifetime of Root CA
Security discussion	Using the shell model for revocation [128], all issued certificates have to be renewed after expiration of the RCA-Cert. But the issued CA certificates must not be added on the CRL as the issuer (RCA) is expired.
	Certificate request and response is encrypted
Stakeholder/Business discussion	No manual interaction necessary as involved. Certificates expire automatically.
	OEM has to update all ITS stations that have installed the expired root certificate as trust anchor

Table 8.22: End of RCA Lifetime

Use case label	5.23
Use case name	End of lifetime of Long-term CA
Actors	RCA, LTCA, PCA, VSS, HSM
Precondition	ITS station has valid root certificate
	ITS station can communicate with PKI. -Communication channel must not be secure. Transport protection of certificate request and response is ensured by packet signature and encryption. -Communication channel must not be stable and can be ad-hoc based. Establishing of a session between ITS station and PKI is not necessary.
	ITS station has valid address of PKI servers. (e.g. IP-address and port number)
Postcondition	VSS of ITS station is equipped with long-term certificate issued by a new LTCA
Trigger 1 (T1)	(Security) service provider goes out of business
Trigger 2 (T2)	Backend connectivity
Process.LTCA.StopProvidingLTCACert	T1
	- Shut down data service. Stop providing LTCA-Cert -Destroy private key of LTCA-Cert -Revocation of LTCA-Cert should not be necessary as LTCA-Cert expires
Process.PCA.DeleteLTCACert	T1
	-Delete expired LTCA-Cert from database of PCA
Process.ITS-S.DeleteLTCACert	T1, T2
	- Delete expired LTCA-Cert from database of VSS and HSM
Process.ITS-S.DownloadLTCACert	T1, T2
	Download LTCA-Cert from new LTCA that is not expired. See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S.DeleteLTC	T1, T2
	See Use-case “Revocation of Long-term CA” Table: 8.16
Process.ITS-S.LTCRequest	T1, T2
	See Use-case “ITS-S initialization and registration” Table: 8.6

Use case label	5.23
Use case name	End of lifetime of Long-term CA
Security discussion	Using the shell model for revocation [128], all issued long-term certificates are useless after expiration of the LTCA
	The LTCA is not able to issue certificates with an expiration date larger than the own expiration date. As result, all ITS-S detect that their long-term certificate expires soon and trigger an update with an appropriate LTCA.
Stakeholder/Business discussion	No manual interaction necessary. Certificates expire automatically.

Table 8.23: End of LTCA lifetime

Use case label	5.24
Use case name	End of lifetime of Pseudonym CA
Actors	RCA, LTCA, PCA, VSS, HSM
Precondition	ITS station has valid root certificate
	ITS station can communicate with PKI. -Communication channel must not be secure. Transport protection of certificate request and response is ensured by packet signature and encryption. -Communication channel must not be stable and can be ad-hoc based. Establishing of a session between ITS station and PKI is not necessary.
	ITS station has valid address of PKI servers. (e.g. IP-address and port number)
Postcondition	VSS of ITS station is equipped with pseudonym certificates issued by a new PCA
	Messages from other ITS stations that are signed with pseudonyms, issued by the expired PCA, are not accepted.
Trigger 1 (T1)	(Security) service provider goes out of business
Trigger 2 (T2)	Backend connectivity
Process.PCA.StopProvidingPCACert	T1
	- Shut down data service. Stop providing PCA-Cert -Destroy private key of PCA-Cert -Revocation of PCA-Cert should not be necessary as PCA-Cert expires
Process.ITS-S.DeletePCACert	T1, T2
	See Use-case “Revocation of Root CA” Table: 8.15
Process.ITS-S.DownloadPCACert	T1, T2
	Download PCA-Cert from new PCA that is not expired. See Use-case “ITS-S initialization and registration” Table: 8.6
Process.ITS-S.PCRequest	T1, T2
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.PCA.PCRequest	T1, T2
	See Use-case “ITS-S initialization and registration” Table: 8.6
Process.LTCA.PCRequest	T1, T2
	See Use-case “ITS-S initialization and registration” Table: 8.6

Use case label	5.24
Use case name	End of lifetime of Pseudonym CA
Process.LTCA.PCRequest	T1, T2
	See Use-case "ITS-S initialization and registration" Table: 8.6
Security discussion	Using the shell model for revocation [128], all issued pseudonym certificates are useless after expiration of the PCA
	The PCA is not able to issue certificates with an expiration date larger than the own expiration date. As result, all ITS-S detect that their pseudonym certificates expires soon and trigger an refill of pseudonyms with an appropriate PCA.
Stakeholder/Business discussion	No manual interaction necessary. Certificates expire automatically.

Table 8.24: End of pseudonym CA lifetime

Use case label	5.25
Use case name	Revocation/Deletion of credentials
Actors	HSM, Garage
Precondition	Vehicle has valid long-term certificate and keypair
Postcondition	Vehicle has no valid credentials
Trigger 1 (T1)	End of vehicle lifetime
Trigger 2 (T2)	HSM failure/broken
Trigger 3 (T3)	HSM tampered
Trigger 4 (T4)	Selling car to other country/PKI domain
Process.ITS-S.DeleteHSMCredentials	T1, T2, T3, T4
	-Vehicle is brought to the garage -Maintenance server removes credentials from HSM
Security discussion	How to do it in the field? Self-destruction? internal action? If not needs secure connection to backend
Stakeholder/Business discussion	This use case assumes that a vehicle will be ”discarded” in an organized manner at the end of its life-cycle. But vehicles may end up in other countries or at the junk yard. It might be an unreasonable assumption that these vehicles will connect to the backend to request the end-of-lifecycle procedure because a junk yard may have no motivation to spend time for that. Therefore, it might be a better assumption that no such interaction will take place. Maybe it is worth assuming that the OBU is part of the theft protection mechanism of a car (component identification) such that the OBU is deactivated once it is removed from the vehicle. Other on-board components are handled today in the same manner, e.g. radios, are deactivated in order to harden the theft of such components.

Table 8.25: Revocation deletion of credentials

Use case label	5.26
Use case name	HSM failure/ breakdown or electronic component wear (memory)
Actors	PKI authority (LTCA), Service organisation, OBU in ITS station, HSM, diagnostic tool
Precondition	-Vehicle has no valid pseudonym certificate and keypair -Vehicle has no valid long-term certificate and keypair -Vehicle has no acces to a valid Device Identity Key
Postcondition	-Vehicle HSM is replaced by a certified HSM with its own IDK (then perform use cases 5.7 and 5.6) -PKI authority (LTCA) has revoked the older vehicle long-term certificate and keypair
Trigger 1 (T1)	Security backend connectivity
Trigger 2 (T2)	Vehicle OBU at Repair & Maintenance services
Trigger 3 (T3)	Diagnostic tool connected to the OBU via an internal network access confirms the failure/braekdown of the HSM
Process.ITS-S.DetectHSMFailure	T1, T2, T3
	1. The OBU communication stack or VSS detects it has no more valid long-term certificate and pseudonym certificates 2. The vehicle OBU logs security event detected in the VSS Security-event data-base (unable to access OBU certificates) 3. The service operator performs OBU tests via a diagnostic tool and confirms HSM breakdown 4. The service operator reports the failed HSM unit (HSM-ID must be at least accessible on a read-only memory) to the PKI authority 5. The PKI authority (LTCA) update the revocation list adding the failed HSM unit (identified by its HSM-ID) 6. The service operator may return the failed HSM to the suppier (Tier 1 or Tier2) for quality analysis

Use case label	5.26
Use case name	HSM failure/ breakdown or electronic component wear (memory)
Security discussion	-Service operators shall have no access to credentials (IDK, LT certificate/keypair) via a diagnostic tool -A secure connection with the PKI LTCA is needed in the service organisation (online or offline)
Stakeholder discussion	In Process.ITS-S.DetectHSMFailure, the Repair & Maintenance service needs to replace the faulty HSM by an (OEM) certified spare part. This is only possible if the HSM is removal, i.e. externally connected to the vehicle OBU via a USB or Ethernet link.

Table 8.26: HSM failure

Bibliography

[1] Research and I. T. Administration, "The national its architecture 7.0," U.S. Department of Transportation, Research and Innovative Technology Administration (RITA), Tech. Rep. 7.0, January 2012. [Online]. Available: http://iteris.com/itsarch/html/entity/paents.htm

[2] European Commission, "Directive 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data," *Official Journal L 281, 23/11/1995 P. 0031 - 0050*, vol. L 281, October 1995.

[3] Statistisches Bundesamt, "Laufende wirtschaftsrechnungen - ausstattung privater haushalte mit ausgewählten gebrauchsgütern 2010," Statistisches Bundesamt, Wiesbaden, Fachserie 15 Reihe 2, August 2011.

[4] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, "Achieving guaranteed anonymity in GPS traces via uncertainty-aware path cloaking," *IEEE Trans. Mob. Comput*, vol. 9, no. 8, pp. 1089–1107, 2010.

[5] P. Hustinx, "Opinion of the european data protection supervisor on the communication from the commission on an action plan for the deployment of intelligent transport systems in europe and the accompanying proposal for a directive of the european parliament and of the council laying down the framework for the deployment of intelligent transport systems in the field of road transport and for interfaces with other transport modes," *Official Journal of the European Union*, vol. 47, no. 2, pp. 6 – 15, 2010.

[6] U.S. Supreme Court, "United states v. knotts, 460 u.s. 276 (1983) united states v. knotts, 460 u.s. 276 (1983)," 1983.

[7] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J. P. Hubaux, "Secure vehicular communication systems: Design and architecture," *IEEE Communications Magazine*, vol. 0163-6804/08, pp. 100 – 109, 2008.

[8] F. Kargl, P. Papadimitratos, L. Buttyan, M. Müter, B. Wiedersheim, E. Schoch, T.-V. Tongh, G. Calandriello, A. Held, A. Kung, and J.-P. Hubaux, "Secure Vehicular Communications: Implementation, Performance, and Research Challenges," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 2–8, November 2008.

[9] B. Wiedersheim, Z. Ma, F. Kargl, and P. Papadimitratos, “Privacy in inter-vehicular networks: Why simple pseudonym change is not enough,” in *Wireless On-demand Network Systems and Services (WONS), 2010 Seventh International Conference on*, feb. 2010, pp. 176 –183.

[10] M. Gerlach and F. Guttler, “Privacy inVANETs using changing pseudonyms - ideal and real,” *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, pp. 2521 –2525, april 2007.

[11] L. Buttyán, T. Holczer, and I. Vajda, “On the effectiveness of changing pseudonyms to provide location privacy in VANETs,” in *Security and Privacy in Ad-hoc and Sensor Networks, 4th European Workshop, ESAS 2007, Cambridge, UK, July 2-3, 2007, Proceedings*, ser. Lecture Notes in Computer Science, F. Stajano, C. Meadows, S. Capkun, and T. Moore, Eds., vol. 4572. Springer, 2007, pp. 129–141.

[12] J. Freudiger, M. Raya, M. Félegyházi, P. Papadimitratos, and J.-P. Hubaux, “Mix-Zones for Location Privacy in Vehicular Networks,” in *ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*, Vancouver, 2007.

[13] E. Schoch, N. Bißmeyer, H. Stübing, B. Lonc, and S. Götz, “Public key infrastructure - memo,” Car 2 Car Communication Consortium, Tech. Rep., 2011.

[14] L. Fischer and C. Eckert, “506 ways to track your lover,” in *VTC Fall*. IEEE, 2008, pp. 1–5.

[15] S. Roudier, “The Sevilla process and the review of the bat reference documents (BREFs),” http://www.esafetysupport.org/download/eSafety_Activities/eSafety_Working_Groups/eSecurityWG/Article29mtg021209/esecurity_20091202_b_bats.pdf, October 2007.

[16] F. Schaub, F. Kargl, Z. Ma, and M. Weber, “V-tokens for conditional pseudonymity in vanets,” in *IEEE Wireless Communications and Networking Conference (WCNS)*, 2010.

[17] A. Pfitzmann and M. Hansen, “A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management,” TU Dresden, Tech. Rep. v.034, August 2010.

[18] M. Gruteser and X. Liu, “Protecting privacy, in continuous location-tracking applications,” *Security Privacy, IEEE*, vol. 2, no. 2, pp. 28 – 34, March 2004.

[19] D. L. Chaum, “Untraceable electronic mail, return addresses, and digital pseudonyms,” *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, Feb. 1981.

[20] P. Papadimitratos, L. Buttyan, J.-P. Hubaux, F. Kargl, A. Kung, and M. Raya, “Architecture for secure and private vehicular communications,” in *7th International Conference on ITS Telecommunications*. IEEE press, June 2007.

[21] P. Papadimitratos, ““On the road” - Reflections on the Security of Vehicular Communication Systems,” in *Proceedings of the IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, Columbus, OH, USA, September 22-24 2008, pp. 359–363.

[22] P. Papadimitratos, V. Gligor, and J.-P. Hubaux, "Securing Vehicular Communications - Assumptions, Requirements, and Principles," in *Proceedings of the Fourth Workshop on Embedded Security in Cars (ESCAR)*, Berlin, Germany, November 2006, pp. 5–14.

[23] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure Vehicular Communication Systems: Design and Architecture," *IEEE Comm. Mag.*, vol. 46, no. 11, Nov. 2008.

[24] F. Kargl, P. Papadimitratos, L. Buttyan, M. Müter, B. Wiedersheim, E. Schoch, T.-V. Thong, G. Calandriello, A. Held, A. Kung, and J.-P. Hubaux, "Secure Vehicular Communication Systems: Implementation, Performance, and Research Challenges," *IEEE Comm. Mag.*, vol. 46, no. 11, Nov. 2008.

[25] IEEE1609.2, "IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages," Jul. 2006.

[26] A. Aijaz, B. Bochow, F. Dötzer, A. Festag, M. Gerlach, R. Kroh, and T. Leinmüller, "Attacks on inter-vehicle communication systems - an analysis," in *3rd International Workshop on Intelligent Transportation (WIT 2006)*, March 2006.

[27] E. Schoch, F. Kargl, and M. Weber, "Communication patterns in VANETs," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 119–125, Nov. 2008.

[28] F. Schaub, Z. Ma, and F. Kargl, "Privacy Requirements in Vehicular Communication Systems," in *Symposium on Secure Computing, IEEE International Conference on Privacy, Security, Risk, and Trust (PASSAT 2009)*, no. March. Vancouver: IEEE, Aug. 2009, pp. 139–145.

[29] J. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security & Privacy Magazine*, vol. 2, no. 3, pp. 49–55, May 2004.

[30] J. Douceur, "The sybil attack," in *IPTPS '01: First International Workshop on Peer-to-Peer Systems*. Springer, 2002, pp. 251–260.

[31] B. Parno and A. Perrig, "Challenges in securing vehicular networks," in *Proceedings of Workshop on Hot Topics in Networks (HotNets-IV)*, Nov. 2005.

[32] F. Kargl, E. Schoch, B. Wiedersheim, and T. Leinmüller, "Secure and efficient beaconing for vehicular networks," in *VANET '08: Proceedings of the fifth ACM international workshop on VehiculAr Inter-NETworking*. New York, NY, USA: ACM, 2008, pp. 82–83.

[33] B. Weyl, M. Wolf, F. Zweers, T. Gendrullis, M. S. Idrees, Y. Roudier, H. Schweppe, H. Platzdasch, R. E. Khayari, O. Henniger, D. Scheuermann, A. Fuchs, L. Apvrille, G. Pedroza, H. Seudié, J. Shokrollahi, and A. Keil, "EVITA Deliverable D3.2: Secure On-board Architecture Specification," EVITA Consortium, Tech. Rep., August 2011.

[34] S. Idrees et al, "EVITA Deliverable D3.3: On-board Protocols Specification," EVITA Consortium, Tech. Rep., 2010, deliverable of the EVITA-project (Reference number 224275, ICT-2007.6.2.

[35] E. Fonseca, A. Festag, R. Baldessari, and R. Aguiar, "Support of anonymity in VANETs - putting pseudonymity into practice," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Hong Kong, March 2007.

[36] B. Wiedersheim, Z. Ma, F. Kargl, and P. Papadimitratos, "Privacy in inter-vehicular networks: Why simple pseudonym change is not enough," in *Wireless On-demand Network Systems and Services (WONS), 2010 Seventh International Conference on*, feb. 2010, pp. 176 –183.

[37] M. Gerlach and F. Guttler, "Privacy in vanets using changing pseudonyms - ideal and real," in *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, april 2007, pp. 2521 –2525.

[38] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux, "Eviction of misbehaving and faulty nodes in vehicular networks," *IEEE Journal on Selected Areas in Communications, Special Issue on Vehicular Networks*, vol. 25, no. 8, pp. 1557–1568, October 2007.

[39] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure vehicular communication systems: Design and architecture," *IEEE Communications Magazine*, November 2008.

[40] P. Papadimitratos, A. Kung, J.-P. Hubaux, and F. Kargl, "Privacy and identity management for vehicular communication systems: A position paper," in *Workshop on Standards for Privacy in User-Centric Identity Management*, Zurich, Switzerland, July 2006.

[41] IEEE, "Trial-use standard for wireless access in vehicular environments - security services for applications and management messages," Institute of Electrical and Electronics Engineers, Tech. Rep. 1609.2 - 2006, July 2006.

[42] C. . C. C. Consortium, "Public key infrastructure memo," 2010.

[43] K. Zeng, "Pseudonymous PKI for ubiquitous computing," in *Public Key Infrastructure*, ser. Lecture Notes in Computer Science, A. Atzeni and A. Lioy, Eds. Springer Berlin / Heidelberg, 2006, vol. 4043, pp. 207–222.

[44] F. Armknecht, A. Festag, D. Westhoff, and K. Zeng, "Cross-layer privacy enhancement and non-repudiation in vehicular communication," in *WMAN '07: 4th Workshop on Mobile Ad-Hoc Networks, Bern, Switzerland*, March 2007.

[45] F. Schaub, F. Kargl, Z. Ma, and M. Weber, "V-tokens for Conditional Pseudonymity in VANETs," in *IEEE Wireless Communications & Networking Conference (WCNC 2010)*. Sydney: IEEE, 2010.

[46] D. Jena, S. K. Jena, and B. Majhi, "A novel untraceable blind signature based on elliptic curve discrete logarithm problem," *IJCSNS - International Journal of Computer Science and Network Security*, vol. volume 7, no. 6, pp. pages 269–275, 2007.

[47] K. G. Paterson and G. Price, "A comparison between traditional public key infrastructures and identity-based cryptography," *Information Security Technical Report*, vol. 8, no. 3, pp. 57 – 72, 2003.

[48] L. Fischer, A. Aiijaz, C. Eckert, and D. Vogt, "Secure revocable anonymous authenticated inter-vehicle communication (SRAAC)," in *Proceedings ESCAR '06*, 2006.

[49] B. H. Kim, K. Y. Choi, J. H. Lee, and D. H. Lee, "Anonymous and traceable communication using tamper-proof device for vehicular ad hoc networks," *International Conference on Convergence Information Technology 2007*, pp. 681–686, Nov. 2007.

[50] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, april 2008, pp. 1229 –1237.

[51] S. S. M. Chow, L. C. K. Hui, and S.-M. Yiu, "Identity based threshold ring signature," in *ICISC*, 2004, pp. 218–232.

[52] D. Huang, S. Misra, M. Verma, and G. Xue, "Pacp: An efficient pseudonymous authentication-based conditional privacy protocol for vanets," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 3, pp. 736 –746, sept. 2011.

[53] J. Y. Choi, M. Jakobsson, and S. Wetzel, "Balancing auditability and privacy in vehicular networks," in *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*. New York, NY, USA: ACM, 2005, pp. 79–87.

[54] Y. Xi, K. Sha, W. Shi, L. Schwiebert, and T. Zhang, "Enforcing privacy using symmetric random key-set in vehicular networks," in *Eighth International Symposium on Autonomous Decentralized Systems ISADS'07*, March 2007, pp. 344–351.

[55] C. Zhang, X. Lin, R. Lu, P.-H. Ho, and X. Shen, "An efficient message authentication scheme for vehicular communications," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 6, pp. 3357 –3368, nov. 2008.

[56] "NoW: Network on Wheels," uRL: http://www.network-on-wheels.de/.

[57] "The Car-to-Car Communication Consortium," http://www.car-to-car.org.

[58] "The eSafety eSecurity Working Group," http://www.esafetysupport.org/en/esafety_activities/\esafetyworking_groups/esecurity.htm.

[59] "DSRC: Dedicated Short Range Communications," http://grouper.ieee.org/groups/scc32/dsrc/index.html.

[60] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and Robust Pseudonymous Authentication in VANET," in *VANET '07*, Montreal, Canada, Sept. 2007.

[61] P. Papadimitratos, G. Calandriello, A. Lioy, and J.-P. Hubaux, "Impact of Vehicular Communication Security on Transportation Safety," in *MOVE '08*, Phoenix, AZ, USA, Apr. 2008.

[62] "IEEE P802.11p/D3.0, Draft Amendment for Wireless Access in Vehicular Environments (WAVE)," July 2007.

[63] "Dedicated Short Range Communication at 5.9 GHz Standards Group," http://www.iteris.com/itsarch/html/standard/dsrc5ghz-b.htm.

[64] "ISO TC204 Working Group 16," http://www.calm.hu/.

[65] P. Papadimitratos, G. Mezzour, and J.-P. Hubaux, "Certificate Revocation List Distribution in Vehicular Communication Systems," in *VANET '08*, San Francisco, CA, USA, Sept. 2008.

[66] P. Papadimitratos, L. Buttyan, J.-P. Hubaux, F. Kargl, A. Kung, and M. Raya, "Architecture for Secure and Private Vehicular Communications," in *ITST'07*, Sophia Antipolis, France, Jun. 2007.

[67] D. Chaum and E. van Heyst, "Group Signatures." in *EUROCRYPT '91*, Brighton, UK, Apr. 1991.

[68] G. Ateniese and G. Tsudik, "Group Signatures à la carte," in *SODA '99*, Baltimore, MD, USA, Jan. 1999.

[69] P. Syverson and S. Stubblebine, "Group Principals and the Formalization of Anonymity." in *FM '99*, Toulouse, France, Sept. 1999.

[70] E. Brickell, J. Camenisch, and L. Chen, "Direct Anonymous Attestation," in *CCS '04*, Washington DC, USA, October 2004.

[71] D. Boneh, X. Boyen, and H. Shacham, "Short Group Signatures," in *Crypto '04*, Santa Barbara, CA, USA, Aug. 2004.

[72] M. Bellare, D. Micciancio, and B. Warinschi, "Foundations of Group Signatures: Formal Definition, Simplified Requirements and a Construction based on Trapdoor Permutations," in *Adv. in Cryptology*, May 2003.

[73] M. Bellare, H. Shi, and C. Zhang, "Foundations of Group Signatures: The Case of Dynamic Groups," in *CT-RSA '05*, San Francisco, CA, USA, Feb. 2005.

[74] D. Boneh and H. Shacham, "Group Signatures with Verifier-Local Revocation," in *CCS '04*, Washington DC, USA, Oct. 2004.

[75] IEEE 1363a-2004, "IEEE Standard Specifications for Public-Key Cryptography - Amendment 1: Additional Techniques," 2004.

[76] "The CVIS project, http://www.cvisproject.org/."

[77] "OpenSSL," http://www.openssl.org.

[78] M. Brown, D. Hankerson, J. Lopez, and A. Menezes, "Software Implementation of the NIST Elliptic Curves Over Prime Fields," in *CT-RSA 2001*, San Francisco, CA, USA, April 2001.

[79] N. Koblitz and A. Menezes, "Pairing-Based Cryptography at High Security Levels," Cryptology ePrint Archive, Report 2005/076, 2005.

[80] D. Page, D. J. Bernstein, and T. Lange, "Report on ebats performance benchmarks," European Network of Excellence in Cryptology, Tech. Rep. IST-2002-507932-D.VAM.9, March 2007.

[81] IEEE1609.2, "IEEE trial-use standard for wireless access in vehicular environments - security services for applications and management messages," July 2006.

[82] M. Gerlach, A. Festag, T. Leinmuller, G. Goldacker, and C. Harsch, "Security architecture for vehicular communication," in *WIT 2007*, Hamburg, Germany, March 2007.

[83] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in vanet," in *ACM VANET*, Montreal, Quebec, Canada, 2007.

[84] P. Papadimitratos, L. Buttyan, J.-P. Hubaux, F. Kargl, A. Kung, and M. Raya, "Architecture for secure and private vehicular communications," in *IEEE ITST*, Sophia Antipolis, France, Jun. 2007.

[85] N. Ristanovic, P. Papadimitratos, G. Theodorakopoulos, J.-P. Hubaux, and J.-Y. L. Boudec, "Adaptive Message Authentication for Multi-Hop Networks," in *International Conference on Wireless On-Demand Network Systems and Services (IEEE/IFIP WONS)*, January 2011, pp. 96–103.

[86] M. Fiore, C. Casetti, C.-F. Chiasserini, and P. Papadimitratos, "Secure neighbor position discovery in vehicular networks," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, june 2011, pp. 71 –78.

[87] S. Brands and D. Chaum, "Distance-bounding protocols," in *Advances in Cryptology — EUROCRYPT '93*, ser. Lecture Notes in Computer Science, T. Helleseth, Ed. Springer Berlin / Heidelberg, 1994, vol. 765, pp. 344–359.

[88] P. Papadimitratos, M. Poturalski, P. Schaller, P. L. and D. Basin, S. Čapkun, and J.-P. Hubaux, "Secure Neighborhood Discovery: A Fundamental Element for Mobile Ad Hoc Networking," *IEEE Communications Magazine*, vol. 46, no. 2, pp. 132–139, February 2008.

[89] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux, "Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possibility," in *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, Tokyo, Japan, March 2008, pp. 189–200.

[90] ——, "Towards Provable Secure Neighbor Discovery in Wireless Networks," in *ACM Workshop on Formal Methods in Security Engineering*, Alexandria, VA, USA, October 2008, pp. 31–42.

[91] S. Capkun, K. Rasmussen, M. Cagalj, and M. Srivastava, "Secure location verification with hidden and mobile base stations," *Mobile Computing, IEEE Transactions on*, vol. 7, no. 4, pp. 470 –483, april 2008.

[92] S. Capkun and J.-P. Hubaux, "Secure positioning in wireless networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 2, pp. 221 – 232, feb. 2006.

[93] Z. Zhu and G. Cao, "Applaus: A privacy-preserving location proof updating system for location-based services," in *INFOCOM, 2011 Proceedings IEEE*, april 2011, pp. 1889 –1897.

[94] A. Studer, F. Bai, B. Bellur, and A. Perrig, "Flexible, extensible, and efficient vanet authentication," *Journal of communication and networks*, vol. 11, no. 6, pp. 574–588, 2009.

[95] "Please Rob Me," http://pleaserobme.com, accessed on 20-5-2011.

[96] "TomTom," http://www.tomtom.com, accessed on 20-5-2011.

[97] "Foursquare," http://foursquare.com, accessed on 20-5-2011.

[98] V. Manolopoulos, P. Papadimitratos, S. Tao, and A. Rusu, "Securing smartphone based its," in *ITS Telecommunications (ITST), 2011 11th International Conference on*, aug. 2011, pp. 201 –206.

[99] G. T. 33.220, "Generic Authentication Architecture (GAA); Generic bootstrapping architecture (Release 9)," 2009.

[100] G. Calandriello, P. Papadimitratos, A. Lioy, and J.-P. Hubaux, "On the Performance of Secure Vehicular Communication Systems," *IEEE Transactions on Dependable and Secure Computing*, to appear.

[101] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich, "How to Win the Clone Wars: Efficient Periodic n-Times Anonymous Authentication," in *CCS*, Alexandria, VA, USA, Oct. 2006.

[102] "Google Latitude," https://www.google.com/latitude, accessed on 31-1-2012.

[103] P. Eckersley, "How unique is your web browser?" in *Privacy Enhancing Technologies*, ser. Lecture Notes in Computer Science, M. Atallah and N. Hopper, Eds. Springer Berlin / Heidelberg, 2010, vol. 6205, pp. 1–18.

[104] C. Y. Ma, D. K. Yau, N. K. Yip, and N. S. Rao, "Privacy vulnerability of published anonymous mobility traces," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, ser. MobiCom '10. New York, NY, USA: ACM, 2010, pp. 185–196.

[105] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*, ser. MobiSys '03. New York, NY, USA: ACM, 2003, pp. 31–42.

[106] R. Shokri, J. Freudiger, M. Jadliwala, and J.-P. Hubaux, "A distortion-based metric for location privacy," in *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, ser. WPES '09. New York, NY, USA: ACM, 2009, pp. 21–30.

[107] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *Distributed Computing Systems, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on*, june 2005, pp. 620 –629.

[108] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: query processing for location services without compromising privacy," in *Proceedings of the 32nd international conference on Very large data bases*, ser. VLDB '06. VLDB Endowment, 2006, pp. 763–774.

[109] J. Meyerowitz and R. Roy Choudhury, "Hiding stars with fireworks: location privacy through camouflage," in *Proceedings of the 15th annual international conference on Mobile computing and networking*, ser. MobiCom '09. New York, NY, USA: ACM, 2009, pp. 345–356.

[110] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: anonymizers are not necessary," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '08. New York, NY, USA: ACM, 2008, pp. 121–132.

[111] R. Anderson and T. Moore, "Information security economics – and beyond," in *Advances in Cryptology - CRYPTO 2007*, ser. Lecture Notes in Computer Science, A. Menezes, Ed. Springer Berlin / Heidelberg, 2007, vol. 4622, pp. 68–91.

[112] R. Shokri, P. Papadimitratos, G. Theodorakopoulos, and J.-P. Hubaux, "Collaborative location privacy," in *Mobile Adhoc and Sensor Systems (MASS), 2011 IEEE 8th International Conference on*, oct. 2011, pp. 500 –509.

[113] T. Jiang, H. J. Wang, and Y.-C. Hu, "Preserving location privacy in wireless lans," in *Proceedings of the 5th international conference on Mobile systems, applications and services*, ser. MobiSys '07. New York, NY, USA: ACM, 2007, pp. 246–257.

[114] 3rd Generation Partnership Project, "3GPP GSM R99," in *Technical Specification Group Services and System Aspects.*

[115] R. Shokri, P. Papadimitratos, and J.-P. Hubaux, "Mobicrowd. a collaborative location privacy preserving lbs mobile proxy," 2010, in *MobiSys - Demo Session*.

[116] "FlashlinQ: A Clean Slate Design for Ad Hoc Networks."

[117] "NIC: Nokia Instant Community."

[118] "Wi-fi direct," http://www.wi-fi.org/wi-fidirect.php.

[119] Z. Cao, J. Kong, U. Lee, M. Gerla, and Z. Chen, "Proof-of-relevance: Filtering false data via authentic consensus in vehicle ad-hoc networks," in *IEEE INFOCOM Workshops 2008*, Phoenix, AZ, USA, April 2008, pp. 1–6.

[120] J. Petit, "Analysis of ecdsa authentication processing in vanets," *3rd international conference on New technologies, mobility and security*, pp. 388–392, 2009.

[121] P. Papadimitratos, F. Kargl, M. Weber, and T. Leinmuller, "Secure vehicular communications: Design and architecture," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 100–109, November 2008.

[122] IEEE, "Draft standard for wireless access in vehicular environments - security services for applications and management messages," Institute of Electrical and Electronics Engineers, Tech. Rep. 1609.2 - 20011 (D9), May 2011.

[123] M. Raya, P. Papadimitratos, V. Gligor, and J.-P. Hubaux, "On Data-Centric Trust Establishment in Ephemeral Ad hoc Networks," in *Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM)*, Phoenix, AZ, USA, April 2008, pp. 1238 – 1246.

[124] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, pp. 382–401, July 1982.

[125] M. Fischer, "The consensus problem in unreliable distributed systems (a brief survey)," in *Proceedings of the 4th International Conference on Fundamentals of Computation Theory (FCT'83)*, August 1983, pp. 127–140.

[126] B. H. Kantowitz, R. J. Hanowski, and S. C. Kantowitz, "Driver acceptance of unreliable traffic information in familiar and unfamiliar settings," *Human Factors*, vol. 39, no. 2, pp. 164–176, 1997.

[127] N. Bißmeyer, J. P. Stotz, H. Stübing, E. Schoch, S. Götz, and B. Lonc, "A generic public key infrastructure for securing car-to-x communication," in *18th World Congress on Intelligent Transportation Systems*. ITS America, October 2011.

[128] H. Baier and V. Karatsiolis, "Validity models of electronic signatures and their enforcement in practice," in *Public Key Infrastructures, Services and Applications*, ser. Lecture Notes in Computer Science, F. Martinelli and B. Preneel, Eds. Springer Berlin / Heidelberg, 2010, vol. 6391, pp. 255–270.