SEVENTH FRAMEWORK PROGRAMME

Information & Communication Technologies
Trustworthy ICT

NETWORK OF EXCELLENCE

A European Network of Excellence in Managing Threats and Vulnerabilities in the Future Internet: *Europe for the World* [†]

Deliverable D6.2: Intermediate Report on the Security of the Connected Car

Abstract: The next generation of vehicles will be communicating with each other, with road-side objects and be constantly connected to the Internet. Many applications will be offered to the drivers and smaller hand-held devices like Android and iPhones will be seamlessly integrated into the vehicles' networks. In this report we summarize the challenges and security work taking place in this area. We highlight the complexity of the problems, communication technologies being used and the security challenges we face together with some possible solutions. We show the shortcomings of many of the existing technologies used to secure traditional systems and discuss what is needed in the vehicular environment.

Contractual Date of Delivery	August 2012
Actual Date of Delivery	September 2012
Deliverable Dissemination Level	Public
Editor	Tomas Olovsson
Contributors	All *SysSec* partners
Quality Assurance	Magnus Almgren, Davide Balzarotti, Zlatogor Minchev, Philippas Tsigas

[†] The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 257007.

The *SysSec* consortium consists of:

FORTH-ICS	Coordinator	Greece
Politecnico Di Milano	Principal Contractor	Italy
Vrije Universiteit Amsterdam	Principal Contractor	The Netherlands
Institut Eurécom	Principal Contractor	France
IICT-BAS	Principal Contractor	Bulgaria
Technical University of Vienna	Principal Contractor	Austria
Chalmers University	Principal Contractor	Sweden
TUBITAK-BILGEM	Principal Contractor	Turkey

Contents

List of Figures

Foreword

In the *Smart Environment* work package in the *SysSec* network of excellence, we especially consider the security of networks and devices that comprise smart environments. In the first deliverable, *Report on The State of the Art in Security in Sensor Networks*, we considered low-capability devices such as sensor nodes and their respective networks. Research-wise, we considered the fundamental network-service algorithms for such environments. In this second deliverable, *Intermediate Report on the Security of The Connected Car*, we consider a specific application area to focus the discussion. We choose the example of the connected car as it is an area being developed actively both by industry and in academia. It is also an area, as will be shown in the report, with reported security problems that need to be considered before it is possible to move forward. The importance of such research has been stressed, for example in the *First Report on Threats on the Future Internet and Research Roadmap* (2011), where we invited a number of experts to discuss current and future trends in the area.

The objective of this deliverable is to give a broad survey of areas of importance to the security of the connected car, especially in regards to European projects. As we also want to give the deliverable some depth, we include an appendix describing research into a specific problem in this environment.

1 Introduction

The Internet has now reached our vehicles and many new services will be introduced in the coming years. Some services target drivers and passengers such as navigation and driver assistance systems, and other focus on the vehicle itself such as remote diagnostics and remote software updates. Most car manufacturers have plans to offer a fairly large number of services and we now face the challenge to implement new functionality without sacrificing traffic safety. The vehicle is a complex safety-critical system with components that must function at all times, and security problems should never result in safety problems or in an immediate halt of all systems. Instead the vehicle must operate in a degraded and fail-safe mode when under attack and when security problems have been detected.

Today's vehicles have an internal network consisting of 50 to 100 computers or Electrical Control Units, ECUs. The internal network is of the size of a small company and internal security is currently more or less absent[1]. The software in a modern car contains tens of millions of lines of code with a total size of more than 100 MBytes [46]. This vehicle will now be connected to the infrastructure around it, i.e. to road-side objects, to other vehicles and to the Internet, and security is a key component which must be in place when these new services are introduced.

1.1 The connected car

Communication between vehicles and the outside world will in almost all cases be wireless, exceptions may be found in repair shops and when vehicles are parked. It is possible to access the internal network by connecting a device directly to the internal buses of a vehicle, for example in a repair

[1] We will not write about specific car brands or manufacturers unless it is important for the discussion. Actual implementations may vary between car brands and models.

shop to diagnose problems and to update the software, but also by car owners and other people in order to "enhance" or change the car's functionality. With a sound security design, it should not matter whether the communication is wired or wireless. However, physical modification of ECUs and the possibility to physically attach devices to the internal network must be paid special attention since, as shown later, it can not be ruled out that the car owner modifies or adds equipment to the car that interferes with its normal functionality.

Vehicular communication, VC, is divided into two or sometimes three categories, collectively called V2X:

- V2I: Vehicle-to-Infrastructure communication. Many services will be implemented and most are related to safety, for example to alert drivers about traffic lights, speed limits and to inform about road works ahead.
- V2V: Vehicle-to-Vehicle communication. This is the area most researchers and application developers focus on. Typical services are anti-collision systems such as early break warnings from other vehicles, information about emergency vehicles approaching, synchronized lane change support, traffic jam ahead warnings, and services facilitating the driving experience such as car platooning.
- V2M: Vehicle-to-Mobile device which refers to communication using for example Bluetooth and near field communication (NFC) techniques. In the rest of this document, we will not include V2M communication in the V2X concept unless explicitly stated.

Different communication technologies are used for vehicular communication: WLAN (IEEE 802.11a,b,g or n) can be used to connect vehicles to conventional access points, for example to download multimedia contents when parked at home. Mobile phones offering Bluetooth connectivity for hands-free operation is already implemented, most likely without considering that telephones also have a GPRS/3G data connection to the Internet and therefore may bridge the car with the Internet.[2] A new standard for dedicated short range communications (DSRC) has been developed. It is based on the new IEEE 802.11p standard which is intended to be used for communication with the infrastructure around the vehicle and with other vehicles. There are also many other devices that are communicating with the vehicle, for example wireless keys, RFID cards identifying drivers, radio communication for traffic information (RDS) and navigation (GPS).

Communication patterns and the actual technology used depend highly on the application, some are classical client-server applications where the vehicle connects to a server or a portal. Other services communicate with road-side objects or are based on ad-hoc communication between different

[2]We will discuss a case study of Bluetooth communication in detail in Appendix A.

parties where short-lived VANETs (Vehicular ad-hoc networks) are formed. For some of the services, it is essential that the parties are fully authenticated and their identities are known by all participants, and for other services privacy can be more important than being able to identify each other. A compromise may sometimes be acceptable, for example where vehicles can be anonymous to each other but road-side objects are able to do proper authentication. Protection against non-repudiation and Sybil attacks (multiple identities) is important for some services, i.e. even if the parties do not want to reveal their real identities to each other, there should be some limitations to what damage they may cause, and it should always be possible for an authorized party to reveal their identities.

1.2 Services and benefits

There are many services that can be offered and car manufacturers have long lists of applications they would like to offer, and there will likely be something similar to an "AppStore" in the car where the owner, driver and passengers can chose to install both free applications and subscribe to different services. In addition, mobile phones and other hand-held devices will be seamlessly integrated into the vehicles' driving experience.

Applications will be offered by many parties such as the car manufacturer, government organizations, trusted third parties teaming up with the manufacturers and by independent third-party application developers. Some applications may be mandatory and offer services from legal authorities, others are safety improving services (e.g. driver assistance and accident reporting systems) and yet other are services the owner, driver and passengers would like to install.

It yet remains to define an architecture that allows third party applications to run in the car environment in a safe and secure way. Many solutions such as certification, sandboxing and isolation have been proposed, but it will likely take many years before car manufacturers can allow third parties to freely develop software for the vehicles and still be able to guarantee the safety of the vehicle.

The services can roughly be categorized as:

- *Services improving the driving experience* and safety on the roads by giving advice and notifying drivers about events. Examples include car platooning (cooperative driving) and collision avoidance systems that give early warnings and notifications to drivers. Vehicles talk to each other and are fully aware of other vehicles' plans and act accordingly, for example by notifying drivers to give way to emergency vehicles approaching at high speed.

- *Critical safety services*, for example emergency actions from the vehicle when a crash is impossible to avoid and to call for help when sensors detect that an accident has occurred [59]. Many of these services are active and the car will take action and help the driver in difficult situations.
- *Traffic optimization*: Information is spread among vehicles about road conditions, congestion problems, accidents and overall traffic throughput.
- *Commercial services* such as automatic payments of parking, road tolls and taxes based on when and where a vehicle has been driven.
- *Subscriptions to improved car functionality*. The car may have more functionality than the owner has initially purchased; it may be possible to use the Internet to purchase or for a limited time rent functionality such as *automatic parking support* without having to visit the car dealership. Cars may also be remotely diagnosed and the car manufacturers can offer updated services and patch software in vehicles on the field, not only in the repair shops.
- *Services unique to the driver*, not to the car. Insurances may in the future follow the driver and not necessarily be tied to a particular vehicle. Drivers may subscribe to services that are available to them regardless of what vehicle they are driving.

All this together, the use of many different communication technologies, different types of communication requirements, real-time requirements, authentication mixed with anonymity, etc., makes security work very challenging. In addition we will soon have a large number of applications in our vehicles and several of them will be third party applications. It is obvious that all communicating systems must be protected against external threats (attackers), but there are also many different parties involved with the vehicle that do not necessarily trust each other. The driver may not always have the same interests as the owner of the car, for example regarding sharing information about where and how aggressively the car has been driven. Furthermore, owners and drivers may not always be trusted by the car manufacturers, since there may be optional services offered by the manufacturers that the owner must purchase or subscribe to, to be able to use. If there is an easy way to obtain such services for free, e.g. to patch the software in the vehicle, many owners will probably use this opportunity.

In the remaining of this deliverable we will investigate how the new ways vehicles will communicate affect safety and security, what threats emerge and what can be done to mitigate these security threats. We also look at different projects and standardization efforts with respect to communications and security.

2 Complexity and security challenges

In order to reach the market in time, the current approach to security has been to implement services and solve security problems one application at a time, while waiting for standards and established methods to emerge. This approach will for obvious reasons not be successful in the long run, and more general methods are under development that can be applied to a large number of applications and use cases, such as the framework developed by the European Evita project [1]. Standardization work is also in progress to specify security frameworks for V2X services, such as the ETSI *ITS station* which is an attempt to standardize V2X communication nodes [3] and IEEE who recently have standardized new protocols for wireless communications (1609 and 802.11p). Standardization and certification are discussed in more detail in Chapter 4.

Since car manufacturers act on multinational markets where they have to comply with different legal requirements, there may also be conflicting requirements such as privacy vs. traceability, and general security solutions are needed where it is possible to change functionality of the same car model depending on the country it is shipped to.

What further complicates security work is the vast number of services, communication technologies and protocols that must be supported. There are also real-time requirements and safety aspects that affect security and the functionality of the systems. In the following paragraphs, we will show the complexity of the problem and highlight the challenges we face in each area. We will also show what consequences a lack of security may have by looking at some practically demonstrated security problems.

2.1 Trust and privacy problems

There are many parties involved with the vehicle during its lifetime. It is obvious that people with no legitimate access to the vehicle (i.e traditional

attackers) must be considered in a threat model, but serious security problems may also be caused by authorized people with access to the vehicle.

Trust is problematic in the vehicular environment. For example, the owner of the car is not necessarily trusted by the car manufacturer, the driver may not be fully trusted by the owner, and repair shops may not be fully authorized or trusted by the car manufacturer or the car owner (i.e. should not have full access to all data and programs in the vehicle). There are also third party program developers who want to offer services which none of these parties can fully trust. All involved parties have different views of security work and what is important to address, and they may therefore also use different non-cooperating security mechanisms in their work.

The complexity of security problems can be seen in Figure 2.1 where different parties, communication technologies, and security attributes are listed that must be considered [43]. A security assessment of the vehicle must consider all possible combinations of actors, communication technologies, paths and security attributes, a quite complex task. This is discussed in more detail in Section 6.4.

Many parties such as the owner also have physical access to the vehicle and its internal network. It is not unlikely to assume that if it is easy to enhance the car's functionality, this will become popular. The internal car architecture must therefore also have a certain degree of protection against "attackers" with physical access to the vehicles even if it happens to be the car owner.

Privacy issues are important. Most car owners and drivers do not want to reveal their identities to everyone at all times. However, being completely anonymous opens up for attacks against many services. Sybil attacks, i.e. to be able to use multiple identities could be useful for example by an attacker to spread false information about congestion in order to have the road for him/herself. Therefore, even if it is possible to use pseudonyms in many situations, legal and other requirements may require that a trusted third party is able to reveal the real identity behind a pseudonym, if needed.

Privacy issues related to different applications must also be addressed. The owner may want to track the vehicle and how it is driven, something the driver may not always want. Some external services such as automatic payment of road tolls may also need to access private information. These and other privacy problems are further discussed in Chapter 8.

2.2 V2X communication technologies

Communication with the outside world can be done using many different technologies at the same time. Examples include Dedicated Short Range Communication (DSRC) which uses IEEE 802.11p, normal WLAN communication using 802.11a,b,g,n, and cellular communication (GSM, GPRS, 3G,

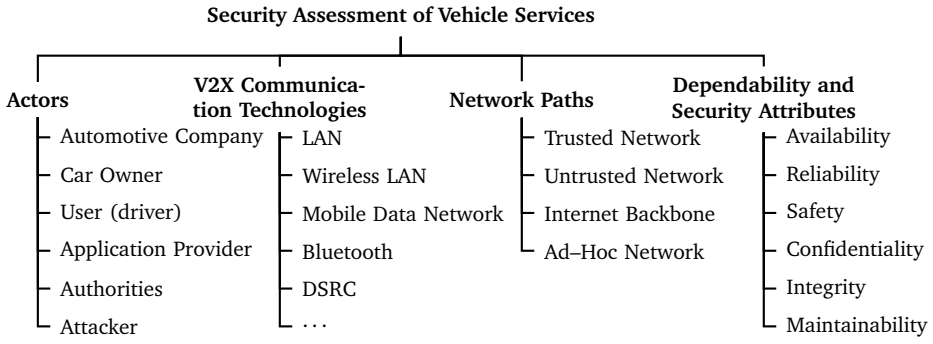


Figure 2.1: Security Assessment Tree

4G) for traditional client-server applications using the Internet, Bluetooth, and NFC (near field communications) devices. The large number of technologies complicates security work since different applications may use different technologies for its services and it may be hard to know which traffic should be allowed on what interface. There are many attack surfaces in the system and not just one communication stack that needs to be protected.

A more detailed discussion of these communication technologies and related protocols can be found in Chapter 3.

2.3 In-vehicle communications

The in-vehicle network spans the whole vehicle and consists of networks of different bus-system technologies: *CAN*, *LIN*, *MOST*, and *FlexRay*. These networks are connected to each other through special *gateway ECUs*, although the internal architecture of the car varies depending on brand and car model. The Evita project [1] has defined a "use-case" architecture model that describes a possible configuration of a vehicular network, shown in Figure 7.1. The use of this model and how to separate traffic between internal networks is discussed in Chapter 7.

Traditional security mechanisms used to secure internal networks cannot be used directly due to limitations and constraints specific to vehicles and the car industry:

- Resource constraints of the ECUs, i.e. the ECU has limited processing power and memory. Complex cryptographic operations or storage of larger amounts of data in ECUs are not possible.
- Due to severe limitations of cost for the connected devices (ECUs), all security solutions must be very cost efficient. The cost of a typical ECU is in the order of $1 and even a marginal increase of cost is problematic to introduce. If the cost of each ECUs within a car that contains 100

ECUs increase by $1, the yearly cost for a larger car manufacturer producing 1,000,000 cars a year would be $100,000,000.

- Lifetime of the solution, the vehicle may be in use for at least 10-15 years, preferably even longer. A complicating factor is that the design phase of vehicles is by tradition very long; vehicles to be released 10 years from now are already on the drawing board and many design decisions are made already today. Security designs must therefore be modular and sound to survive for such a long time.

Some specialized ECUs like a few *gateway ECUs* may be equipped with additional functionality such as hardware support for encryption, but most other ECUs should preferably contain the same hardware as today. This places many constraints on the security solutions that can be implemented in a vehicle.

2.4 Real-time requirements

Real-time requirements put boundaries on security functionality, both on internal and external communication and many applications require hard real-time, or near real-time responses to work. Collision avoidance systems need to react in a short time to be useful, yet they may have to communicate with many other vehicles and be able to verify the correctness of messages they receive.

Security functionality must be designed in such a way that real-time requirements can be fulfilled and that it does not disturb other real-time functions in the car. In addition, protection against misbehaving nodes trying to do denial of service (DoS) attacks such as flooding the internal network with traffic, is important since it may cause essential functions to fail.

2.5 Product life cycle and legal requirements

The expected lifetime of a vehicle is very long compared to other businesses. It is not unlikely to assume that many vehicles will be in use after 15-20 years, and the lifetime of security solutions and implemented mechanisms must be adapted accordingly. It is not possible to foresee what threats and risks we face that far in the future, and security mechanisms must therefore be dynamic and possible to change during the full lifetime of the vehicle. The architecture of the vehicle must be designed in such a way that security functionality such as cryptographic algorithms, keys, firewalls, etc., easily can be updated in the future without major software and hardware changes. The SeVeCOM project has therefore suggested an architecture where security functionality is isolated from the rest of the software and implemented

as plug-ins into the network stack to allow easy modification. This approach is described in Section 7.2.

Since it will be necessary to quickly patch vulnerabilities as soon as they are discovered, it is necessary to implement support for secure remote software updates. This is something used for a long time in the computer domain (e.g. Microsoft's patch Tuesday), and vehicles have to be patched not only when they return for scheduled service once every second or third year. A secure software update mechanism is needed which uses the Internet.

The software developed for the automotive industry must follow country laws and requirements. This puts constraints on functionality and the development processes. It is likely that countries will have different requirements on security and privacy issues. Law enforcement agencies in different countries may have different views on what information they need to access. There will also be different organizations handling vehicle identities and issuing certificates, yet all vehicles should be able to participate in future communications and authenticate each other in a safe and secure way, as will be discussed in Chapter 8.

The different vendors will also have different implementations of similar services, yet they need to talk to the same infrastructure around the vehicles and with other vehicles. Coordination and testing of such systems will be a major challenge in the future.

All these new services and security functionality which will be introduced may also fail and affect the reliability of the vehicles. Failure in active security functions, like intrusion detection systems making mistakes, may render vehicles unable to function and cause problems for the manufacturers. We will see attacks in the future with the aim to ground a complete fleet of vehicles of a particular brand.

3 Communication technologies

Wireless communication with the outside world will be based on several communication technologies. Security solutions must be general enough to handle applications that use different techniques, possibly at the same time, to communicate with the outside world.

DSRC, Dedicated Short Range Communications is a general term often seen in V2X documents but are not only used for this purpose and in this environment. The meaning of the term DSRC has changed over time and may mean RFID communication, WLAN communication or any other of several different short range communication systems. In the vehicular domain, DRSC is synonymous with the use of the 5.9 GHz frequency band which is dedicated for V2X communications. Standardization is going on in this domain and the new **IEEE 1609 WAVE** (Wireless Access in Vehicular Environments) protocol which uses the IEEE 802.11p link-layer protocol will likely be used.

IEEE 802.11p is a technology based on 802.11a but has some special functions more suitable for short-lived ad-hoc communication. Higher layers are expected to follow the IEEE 1609 WAVE standard which covers layer 3 to 7 where both short broadcast communication as well as TCP/IP is supported. These protocols and their security features are further discussed in Section 3.1.

WLAN technology, i.e. traditional IEEE 802.11a,b,g,n communication can also be used and offer Internet access to traditional client-server applications, for example by car owners who want to connect the car's multimedia system to their home network. The car manufacturers and third party software vendors may also want to use WLAN technology for remote diagnostics and software updates while the vehicle is not moving.

GSM/GPRS/3G. Cellular networks will be used primarily for client-server based services, similar to WLAN connectivity, but can be used where no WLAN networks are present. It will also be used for safety-related services,

such as calling for help if internal sensors have detected a crash and by many other types of applications when road-side devices cannot be used. Cellular communication can also be the communications method used to check validity of certificates by consulting on-line revocation (CRL) lists.

Bluetooth technology is used today to connect mobile phones to the multimedia system, primarily for hands-free operation. This may seem harmless, but the multimedia system is connected to the internal CAN bus, and if compromised, arbitrary messages may be sent. Most mobile phones are at the same time connected to the Internet and may function as bridges between the Internet and the internal vehicle network. In the future, Smartphones can also use Bluetooth communication to offer many other services to vehicles. We discuss in depth our efforts within the SysSec consortium to add a security layer that sits on top of the Bluetooth standard in Appendix A.

NFC, Near Field Communication, such as RFID cards are used today for driver identification and can in the future offer more functionality. Vehicles communicating with other devices using Bluetooth, NFC, etc., are often called *V2M* (vehicle to mobile communication) and are sometimes, but not always, included in the *V2X* concept.

GPS and RDS radio communication. RDS information is used to receive broadcast transmissions about traffic conditions and road work. GPS and RDS technology is in place today, but security problems may arise if ECUs within the car start relying on the transmissions or if services offered to vehicles depend on the correctness of the data. A driver may, for example, change his GPS position in order to avoid road tolls and other fees.

3.1 Broadcast V2V and V2I communication

Broadcast DSRC communication will be used for most V2V and V2I services. Messages can contain information such as a vehicle's location, speed, directions, maneuvers such as braking, future plans e.g. to change lane or pass an intersection, etc., and are used to spread awareness between vehicles close to each other.

Wireless access in vehicular environments has recently been standardized by IEEE 1609 (the (WAVE protocol) and it is likely that new services will begin using it for all short-range communications. Unfortunately, standardization work is currently ongoing and some parts of the IEEE 1609 documents such as the 1609.2 trial standard describing security, have recently been withdrawn. When a new version of the standard is released, it remains to be seen whether it will be universally accepted by all vendors and all countries, i.e. whether WAVE will be the new universal standard or not.

WAVE supports V2V and V2I communication using the DSRC 5.9 GHz band dedicated for the intelligent transport system (ITS). WAVE standard-

izes both V2V and V2I communications and describes data exchange, security, and service advertisement between communicating parties and is intended to be a framework for application developers when implementing services. Figure 3.1 shows an overview of the protocol hierarchy and the different protocols used at the different communication layers.

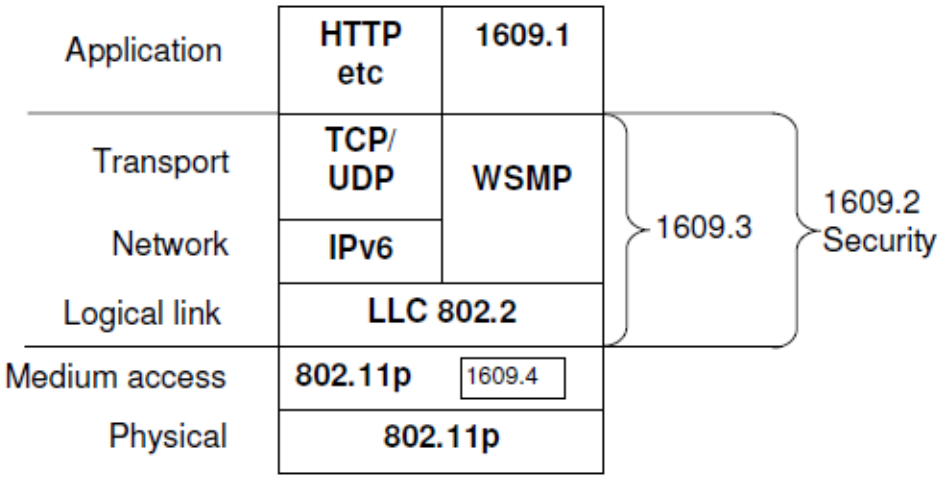


Figure 3.1: WAVE = IEEE 1609 + IEEE 802.11p

The WAVE standard covers layer 3 to 7 and governs modes of operation in the DSRC band including architecture and resource management (1609.1), management, security services and message formats (1609.2), and network services (1609.3). A new protocol, the *WAVE short message protocol* (WSMP) has also been specially designed for broadcast V2X communications, although two special service channels intended for safety-critical applications do support IPv6.

WAVE relies on the **IEEE 802.11p** protocol for the physical (PHY) and link (MAC) layers, which is a modified version of the 802.11a WiFi protocol. It supports up to 27 Mbps using 7 dedicated channels in half duplex mode and communication range is in the order of 300 meters. 802.11p does not explicitly address security and encryption (like WPA and WPA2 which is present in 802.11a) and it must therefore be addressed by upper layer protocols. Security was omitted because of the short-lived communication patterns and problems with authentication of users on lower protocol layers. Instead of spending valuable time computing crypto-keys and exchanging messages with a base station, vehicles should be able to quickly create ad-hoc networks and exchange signed messages directly with as little overhead as possible.

3.2 Vehicular ad-hoc Networks, VANETs

VANETs are often considered to be a subset of MANETs, Mobile ad-hoc networks often discussed in computer science. The most important differences are the short lived communication patterns in VANETs and that it is unlikely that the same nodes forming a VANET ever meet again, thus vehicles will frequently participate in new dynamic ad-hoc networks with new participants they have never seen before. It is therefore not that meaningful to store any credentials or information to speed up reconnection to the VANET in the future. Many VANET applications such as collision avoidance systems also have hard real-time requirements for the communication.

The lifetime of a VANET can be very short. Consider a vehicle approaching a road-side object in 100 km/h. If we assume that communication can take place in a 100m radius from the object, they can only communicate during 7 seconds which includes time for network discovery and possible authentication procedures take. The lifetime of communications between cars driving in opposite directions may be even shorter, and it may therefore be necessary for other vehicles to forward messages to extend the communication range.

It is likely that some messages need to be repeated or forwarded by the nodes in the network to reach all participants, possibly with a delay to allow vehicles to move to increase communication range. Some messages like emergency messages should be spread rapidly and reach as many recipients as possible in a short time, but this can result in network flooding if lots of nodes repeat the same message. New protocols are needed where congestion avoidance is considered, yet still all vehicles in vicinity of the problem should receive the message.

One interesting technique is to allow vehicles (nodes in the communication network) to store information and at a later time and position, retransmit the data, i.e. data is carried from one location to another by vehicles. This will increase communication range for messages, but it also results in new routing problems. Several methods for how this can be done have been suggested, for example using group communication techniques. All these requirements and restrictions make many algorithms developed for MANETs less useful and new methods for communication are therefore under development.

3.3 High-level protocols - reliability and safety

There will be many application-level protocols, some based on TCP/IP and communicating in traditional client-server manner, and other broadcast-based communicating with objects around the car (V2X). Some applications will take care of security themselves, others will rely on security offered by

the communication technology itself, for example for encryption and authentication. ETSI has been mandated to standardize V2V communications within the European Union. Future standards such as the ITS Station architecture work by ETSI [2–4] will therefore likely dictate security requirements for nodes participating in V2X communications, but it will not help developers with how security functionality should be handled within the vehicle and how it may be implemented in various applications.

3.3.1 Attack surfaces

There are many protocol stacks implementing link-network-transport services in a modern vehicle that can potentially be exploited. In addition, not only attacks against the network stack are possible, but all application level protocols that are introduced into the vehicle are potential targets for attacks. Manipulation of data from application layer protocols may also result in severe security problems, for example simple buffer overflow attacks against a third party application may enable modification of the functionality of an ECU and introduce Trojans and viruses.

Unauthorized manipulation of traffic to vehicles may also result in unwanted behavior, for example to enforce non-existing speed limits, to avoid empty but seemingly congested roads, announce lane-changes without the driver having any such plans, cause problems with car platooning and in general disturb functions in and around the car. A more dangerous scenario is an attack that sends spoofed messages to ECUs, for example that orders full speed ahead and disabling the break ECUs when the sensors in the car detect pedestrians in the way.

3.3.2 Security vs. reliability

All services that affect safety must be reliable. The more complex the vehicle is and the more services being implemented, the more vulnerable the system becomes. If drivers start to rely on collision avoidance systems, a silent failure of that system may result in an incident. At the same time, a system that gives too many warnings or behaves erroneously may give bad reputation to the car brand.

Security functionality must focus on preventing accidents. The functions must be designed to support all safety-critical functionality at the expense of other functions if needed. The system must continue to operate even after a security problem, possibly with some limitations or with degraded service for the driver or passengers.

Mechanisms implemented for safety, e.g. fault detection mechanisms must be implemented in such a way that they cannot be used by an attacker and result in security problems or evade protection mechanisms. Intrusion

detection and prevention systems (IDS and IPS systems) may be used to detect the malicious activities and dangerous scenarios, but IPS systems can also be used by attackers to spawn other types of attacks. It may also be a legal problem with having IPS systems taking over the control of a car if it results in an accident. IDS systems are discussed more in detail in Section 7.5.

4 Standardization organizations and research projects

Intelligent Transportation Systems, ITS, that aim to simplify the operation of vehicles by offering various V2X services are no longer limited to laboratories and test facilities of companies. Many functions have reached standardization organizations and other consortia, industrials, and academics, who work with protocols and communication platforms. Their goal is to set the requirements, guidelines, and to standardize communication systems and platforms.

In this chapter, we briefly survey the most important standardization efforts and projects that can affect security work in the vehicular environment. Many of them are described in their original structure by highlighting their scope and goals. The mentioned projects are almost all geographically located in Europe, and their outcome mainly affect ETSI (European Telecommunications Standards Institute) and therefore also end up in ISO for standardization.

4.1 Standardization bodies and consortia

Standardization and provision of standardized communication between vehicles (V2V) and between vehicles and infrastructure (V2I), can be done, and is done, by various standardization organizations and consortia. The following bodies are responsible for planning, development and adoption of the European standards:

4.1.1 International Organization for Standardization (ISO)

ISO/TC 204 (Intelligent transport systems) was created in 1992 and all its Intelligent Transport Systems activities are organized in 14 working groups. The exception is the in-vehicle transport information and control systems

area which is covered by TC 22. They are closely cooperating with ETSI TC ITS, see below.

The *ISO 26262* standard, which is an adaptation of the functional safety IEC 61508 standard, provides methods to mitigate the effect of faults and random failures in hardware. ASIL, defined in ISO 26262, is a way to certify components in the automotive industry with respect to acceptable failure rates and can be used to control and predict the failure behavior of components. The intention is to assess and be aware of the *impact* and possible *damage* that may emerge from failures of components in the vehicle. The ASIL standard addresses safety only (not security) and introduces four different safety levels (1-4), the highest (4) required by safety critical components.

ASIL can be useful when defining requirements and evaluating particular components (ECUs), but when components communicate and use data from other components, it becomes hard to foresee all possible failure modes of the whole system. And even if security is not addressed today, it will affect how critical components in the system can be designed.

4.1.2 European Telecommunications Standards Institute (ETSI)

ETSI has received the mandate to standardize V2V communications within the European Union. Although ETSI has as its primary responsibility standardization work in the telecommunications sector, it also has a committee working with ITS deployment dealing with applications, security and networking.

They have standardized an *ITS station* which is intended to be used in all external vehicular communications, both in vehicles and in road side units [2]. The standard describes the functionality that should be contained in all nodes participating in V2X communications. ETSI has also published a Threat, Vulnerability and Risk Analysis (TVRA) methodology which is applied to the proposed ITS station with its communication and a design of the needed security services [3, 5]. This methodology is supposed to be used to assess and evaluate security related functionality in the vehicular domain.

The ETSI ITS Technical Committee (TC) has different working groups: WG1 develops the basic set of application requirements and services, WG2 provides the architecture specification and addresses the cross layer issues, WG3 provides the 5.9 GHz network and transport protocols, WG4 provides the European profile investigation of 802.11p, and WG5 works with the security architecture

More about the ETSI project and the ITS station is discussed in Section 6.3.

4.1.3 Car2Car Communication Consortium (C2C-CC)

C2C-CC is an industry consortium initiated by European car manufacturers in summer 2002. It is an open non-profit organization with several partners and mainly consists of research institutes, car manufacturers and suppliers. C2C-CC cooperates closely with ETSI TC ITS and the ISO/TC 204 on the specification of the ITS European and ISO standards.

The main goal of the consortium is standardization of protocols and interfaces used in wireless communication between vehicles and infrastructure, i.e. most V2X communications. The aim is to make the different vehicle brands and road-side objects interoperable.

They have also developed a reference architecture which can be used to assess security in communicating vehicles, see Section 6.4

4.1.4 European Committee for Standardization (CEN)

CEN is an international non-profit association created in 1975. It is a major provider of European Standards and technical specifications. Most of the activities in ITS are developed within the CEN/TC 278 “Road transport and traffic telematics”. The technical committee has several working groups working on Dedicated Short Range Communication (DSRC), eSafety and Co-operative systems.

4.2 Larger ITS Projects

The European Commission research and innovation programs fund several framework programs with a variety of topics. The 7th Frame Program, FP7, is the currently ongoing program where the majority of the R&D activities within ITS are handled. Some of the more influential ITS projects within FP6 and FP7 are introduced here.

4.2.1 EVITA (E-safety vehicle intrusion protected applications)

EVITA was a European project, funded under the 7th Framework Program (FP7) 2008-2011. Its main objective was to design, verify and implement a hardware security module to be used in an architecture for securing onboard networks. In this architecture, security relevant components are protected against tampering and sensitive data are protected against compromise. By focusing on protecting the intra-vehicle communication, EVITA complements other projects which mainly focus on V2X communications.

In EVITA, an architecture is created which should enable ECUs to implement cryptography operations in a secure manner. The ECU is equipped

with a cryptographic co-processor protected in a Hardware Security Module, HSM. This module is responsible for performing all cryptography applications. More details on the HSM can be found in Section 7.4.

4.2.2 PRECIOSA (Privacy enabled capability in cooperative systems and safety applications)

PRECIOSA, 2008-2010, addressed questions like whether cooperative systems can comply with future privacy regulations. The major objectives of the PRECIOSA project were to:

- Define an approach for privacy evaluation of cooperative systems in terms of communication privacy and data storage privacy.
- Define a privacy-aware architecture for cooperative systems which involves suitable trust models and a V2V and V2I privacy verifiable architecture which includes components for protection, infringement detection, and auditing.
- Define and validate guidelines for privacy aware cooperative systems.

4.2.3 SeVeCOM (Secure Vehicular Communication)

SeVeCOM, 2006-2010, was a European project funded under the 6th Framework Program. The project focused on the full definition, design, and implementation of security and privacy requirements needed for vehicular communications. The major objectives of the SeVeCOM project were:

- Identification of the variety of threats: attacker's model and potential vulnerabilities; in particular, to study attacks against the radio channel and transferred data, but also against the vehicle itself through internal attacks, e.g. against ECUs, the telematics unit, and the internal control bus.
- Specification of an architecture and of security mechanisms which provide the right level of protection. It addresses issues such as the apparent contradiction between liability and privacy, and the extent to which a vehicle can check the consistency of claims made by other vehicles.
- Definition of cryptographic primitives which take into account the specific operational environment. The challenge was to address (1) the variety of threats, (2) the sporadic connectivity created by moving vehicles and the resulting real-time constraints, and (3) the low-cost requirements of embedded systems in vehicles.

The SeVeCOM project and the architecture created is discussed more in Section 7.2.

4.2.4 NoW (Network on Wheels)

NoW, 2004-2008, was a German project that developed communication protocols and security algorithms for inter-vehicle ad-hoc communication systems. The main objectives of the project were to:

- Support active safety applications and infotainment applications with an infrastructure between vehicles.
- Enhance radio systems based on IEEE 802.11 technology.
- Be active in standardization at European level with the Car2Car Communication Consortium.
- Implementation of a reference system.
- Planning of introduction strategies and business models.

It also provided solutions for (1) position based routing and forwarding protocols, (2) adaptation of wireless LAN under realistic radio conditions, (3) fundamental questions on vehicular antennas, (4) data security in vehicular ad hoc networks, and (5) secure and fast communication between vehicles.

4.2.5 PRESERVE (Preparing Secure V2X Communication Systems)

PRESERVE, 2011 - 2014, is a European project funded under the 7th Framework Program. Its mission is, to design, implement, and test a secure and scalable V2X security subsystem for realistic deployment scenarios. The results from earlier research projects will be combined in PRESERVE and the purpose is to develop and integrate them to a pre-deployment stage. The project aims at providing comprehensive protection ranging from the vehicle sensors, through the on-board network and V2V/V2I communication, to the receiving application.

The main objectives of the project are:

- Create an integrated V2X Security Architecture (VSA) and design, implement, and test a close-to-market implementation termed V2X Security Subsystem (VSS).
- Prove that the performance and cost requirements for the VSS arising in current and future product deployments can be met by the VSS, especially by building a security ASIC for V2X.
- Provide a ready-to-use VSS implementation and to support field operational tests and interested parties so that a close-to-market security solution can be deployed as part of such activities.

- Solve open deployment and technical issues hindering standardization and product pre-development.

4.2.6 OVERSEE (Open VEhiculaR SEcurE platform)

OVERSEE, 2010-2012, is a European project under the 7th Framework Program. The overall goal of OVERSEE is to contribute to the efficiency and safety of road transport by developing the OVERSEE platform, which provides a secure, standardized and generic communication and application platform for vehicles. The main objectives of the project are to create an open platform for the execution of OEM and non OEM applications, a secure single point of access to internal and external communication channels.

4.3 Certifications

Even if standards for security design are developed and legal requirements are defined, we still face the problem of guaranteeing that a particular design fulfills all requirements. Certification of individual components and functions using procedures such as Common Criteria and FIPS may be a way to, at least to some degree, guarantee a sound design. Such certifications may be useful for individual components in a vehicle, but will likely not be applicable to a whole network in a car since any change of the software or hardware requires a full re-certification of the vehicle, a very time consuming and work-intensive process. Therefore, since not all functionality can be certified within a vehicle, a large burden will still be placed on the implementation of services and on the architecture and platforms that run the software.

It is likely that all V2X traffic is sent by stations similar to ETSI's ITS station that fulfill a wide range of security requirements, some even certified. The use of standardized and certified communication nodes for V2X short range communications have several advantages, not just for interoperability, but cooperative development of common modules can enhance security significantly.

Our conclusion is therefore that certifications are to some degree useful but come with several shortcomings: First of all, only smaller components can be certified to higher levels due to increased complexity. Often individual components, such as firewall functionality in a subsystem can be certified, but not a full vehicle with hundreds of communicating ECUs. Second, requiring only certified products in vehicles may prevent deployment of new functionality and create unacceptable delays of patches to known problems. And finally, certifying a product is no guarantee for a secure platform; Microsoft Windows XP was certified at Evaluation Assurance Level 4 (EAL 4) which is quite high, but it is still not considered to be a very secure platform.

Even if it would be possible to certify the functionality of a full vehicle, a problem still exists when new functionality is added or patches are applied. A re-certification is needed as soon as *any* functionality or code is changed. Delta certifications (e.g. as used in Common Criteria certifications where only changes are re-evaluated) may be done, although a complete re-evaluation is needed on regular basis. This is another reason why it is more or less impossible (or at least extremely costly and time consuming) to certify a full vehicle with all its components at any higher level.

5 Security threats and the lack of security

The Internet is a constant source for malicious traffic [40], and since the Internet-connected car will have a public IP address, it will like any other Internet-connected computer system be a target for this traffic. Security mechanisms similar to what we use to protect traditional computer systems and networks are needed, although they must be adapted to the specific requirements and limitations vehicles have, and still be able to meet the high security demands we have for safety-critical systems.

Most research in the area have focused on algorithms and what messages need to be exchanged for various applications, such as for control systems for platooning, how to implement cooperative lane-change support, display messages about approaching dangers, etc., but security is often treated as something to be done in the future [44]. Standards for external V2X communication are emerging (e.g. the trial standard IEEE 1609.2), but the security parts were recently withdrawn and it is not clear when a new standard proposal will be presented. To conclude, we can easily identify a plethora of threats, but available and usable security mechanisms are to a large extent missing and, as we will see, much more work is needed in this area.

5.1 Threats to the in-vehicle network

Most work with security has focused on identifying and showing the lack of security in vehicular systems and the need for protection mechanisms, rather than specifying how the problems should be solved given the special requirements and restrictions that vehicular communications have.

5.1.1 Lack of security mechanisms

Many researchers have shown that there is a significant lack of necessary security mechanisms in in-vehicle networks. Koscher et al. [46] conducted

experiments on two vehicles and by using techniques such as packet sniffing, packet fuzzing, and reverse-engineering, they found a number of attacks that could be performed against the in-vehicle network. Wolf et al. [70] did some early investigations of possible attacks against different buses in the in-vehicle network and demonstrated the lack of security. The results from these studies may not be very surprising since security is not designed into the protocols being used (CAN, LIN, MOST) and any device, ECU or external, that can send messages can forge arbitrary legal messages.

Hoppe and Dittmann [32] used simulations for evaluating security. They investigated the possibility of performing sniffing and replay attacks on the CAN-bus by simulation of an electronic window lift system. To classify their attacks, an adapted version of the CERT Taxonomy proposed by [37] was used, which classifies the vulnerabilities into three classes: *design, implementation* and *configuration*. In a later study, they also performed attacks against the electronic window system using real hardware as well as attacks against the warning lights of the anti-theft system and the air-bag control system [33].

Nilsson and Larson [51] introduced the concept of a vehicle virus. The virus was listening for the message on the CAN-bus that locks the doors remotely, and when that message was captured, the virus would soon afterwards unlock the doors and start the engine.

5.1.2 Security problems

There are several types of security problems that have been, and still can be, exploited that must be addressed:

- *lack of sufficient bus protection.* The CAN-bus lacks necessary protection to ensure confidentiality, integrity, availability, message authenticity, and non-repudiation [33]. Messages on the CAN-bus can be read by other nodes, they have no sender or receiver address (only message types that ECUs can subscribe to), and are not protected by any MAC or digital signature and lack necessary protection of data authentication, data confidentiality and data freshness.

- *weak authentication.* It is possible to illicitly reprogram ECUs with new firmware [46]. The reason is weak authentication and sometimes no authentication at all, see Section 5.2.1.

- *misuse of protocols.* Attacks against the in-vehicle network can be performed by misusing well-chosen mechanisms in the protocols [70]. On the LIN-bus, sending malicious sleep frames could disable the whole network, and on the CAN bus, a DoS attack may be carried out by misusing the bus arbitration mechanism by continuously sending messages with the highest priority, resulting in that no one else can access

the bus. Furthermore, well-formed malicious error messages can be used to attack the fault detection mechanism implemented in CAN and FlexRay and cause ECUs to disconnect from the network.

- *poor protocol implementation.* In some cases the protocol implementation is such that it does not properly reflect the protocol standard. For example, for safety reasons the standard specifies that it should not be possible to put the vehicle or its ECUs into programming mode while the vehicle is moving. However, in some implementations it is indeed possible to launch a command that would disable the CAN communication and put ECUs into programming mode despite the fact that the vehicle is moving [46].

- *information leakage.* Information leakage from the vehicle can be triggered by manipulating the diagnostic protocol, creating a potential privacy violation. Hoppe et al. [35] have demonstrated this by sniffing an ordinary diagnostic session, and then replayed it with slightly modified commands. Since the gateway ECUs are unable to differ between ordinary traffic and diagnostics traffic, both types of traffic will be forwarded and processed.

5.2 External threats

5.2.1 Access to the OBD-II port

Since the internal networks lack protection, people and devices with access to the vehicle can perform all kinds of actions against the ECUs. The vehicle network is easily accessed through the standardized On-Board Diagnostics (OBD-II) port present in all vehicles. This port is used, for example by mechanics in repair shops to check vehicle configurations, change settings, update software and to read diagnostic error messages. The port interface is standardized and easy to connect to, although the internal messages are brand specific and some knowledge is needed to decode them. Some of the attacks, described above, used reverse engineering approaches to figure out the meaning of these messages.

Windows-based PCs can be connected to the OBD-II port which opens up for attacks “enhancing” the functionality in the vehicle. The port can be used by car owners who what to change some functionality, or by attackers using an Internet-connected PC as a gateway to the vehicular network. All traditional attack methods may be used by the attacker, and proper protection is essential, for example in workshops and other environments where such devices are used. It is not unlikely to expect that many other third parties, for example official agencies performing yearly inspections of vehicles,

will use this functionality to check configuration and diagnostic messages, and they all contribute to the complex problem of securing the vehicle.

There are devices available to be purchased such as the "*ELM327 Bluetooth OBD-II*", see Figure 5.1, that interfaces to the OBD-II port and offers Bluetooth connectivity to the CAN bus. The owner who installs the adapter can read, display and clear engine fault codes, view live engine sensors, etc. When installed, the safety of the vehicle depends on the driver's smartphone or PC and whether or not it is compromised. It may make the vehicle's network publicly available on the Internet.

ECUs can also be reprogrammed through the OBD-II interface, although there is a 16-bit security access code needed. Brute-force methods to crack the access code for an ECU takes around a week provided the attacker has access to the vehicle [46]. It is also possible to work in parallel with several ECUs in the vehicle. The protection is not very strong, although it prevents ECUs from being reprogrammed immediately by a malicious message.

Figure 5.1: OBD-II to Bluetooth adapter unit

5.2.2 Attacks against core protocols

All protocols in the network stack can be attacked. The Internet-connected car uses the IP protocol and therefore all Core protocols (DHCP, ARP, ICMP, DNS, TCP, UDP) as well as link-level protocols (802.11p, WLAN, cellular 3G/4G, etc.) face similar threats as other systems do. Application level protocols are unique for this domain and will be obvious targets for attacks and must be designed to be both secure and robust.

The V2X communication protocols (WAVE, IEEE 802.11p, WSMP) can also be targets for attacks, as well as Bluetooth, DSRC and RFID communi-

cations. In Section 5.3 we show that vulnerabilities in the Bluetooth stack in the telematics module have been exploited in real attacks.

Attacks against core protocols have existed for decades and many tools exist that can be used by attackers but also by developers to perform their own penetration tests to make sure that at least all old and well-known vulnerabilities are handled properly. There are many well-known core protocol attacks, for example the Land attack (victim's address present in both source and destination fields), Ping-of-death (oversized IP datagrams), and header exploits (e.g. the length specified in protocol headers differ from the actual length) which may cause various problems from crashes to execution of arbitrary code.

This list of known problems *it is short enough* to allow implementers to test their implementations against almost all of them. However, history has shown that many new implementations do not take these known attacks into consideration, thus they re-appear in new implementations and cause systems to fail. Examples include Microsoft's new network stack in the beta version of Windows Vista where they failed to perform such tests [31], and in Smartphone operating systems where a single link-level datagram can make the device freeze completely and only a removal of the battery makes it functional again [28]. These attacks mainly target the communications unit in the vehicle, although a failure in it may result in a denial of service, or worse, that malware is installed that can send messages fabricated remotely by attackers, on the internal buses.

Many of the threats to vehicles can be analyzed using the same methods used to secure other Internet-connected systems, and similar tools can and should be used to test the robustness of the implementations. Lang et al. [47] provide an interesting discussion of the security implications when the vehicle is connected using an IP-based network. Nine "hypothetical attack scenarios" are suggested based on attacks known from ordinary IT systems, i.e. attacks on the communication protocols, malicious code, and social engineering. Each scenario was analyzed with respect to confidentiality, integrity, availability, authenticity, and non-repudiation. Also, an attempt to quantitatively estimate the impact on safety was conducted and for each of the scenarios, a SIL value (see Section 4.1.1) was proposed.

5.2.3 Vehicle-specific problems

In the vehicular domain, Jenkins and Mahmud [39] have discussed security problems and attacks against the vehicle and they look at both inter-vehicle and in-vehicle communications, and also at software and hardware attacks. Traditional methods derived from the CERT Taxonomy can be used (proposed by Howard and Longstaff [37]), where it is assumed that the attacker can read, spoof, drop, modify, flood, steal, and replay traffic to the vehicle.

Traditional attacks like *DoS attacks* and to send malicious traffic for example to perform *buffer overflow attacks* in order to plant malware, are real threats. Mechanisms similar to what is used in traditional computer-security work (firewalls, MACs, encryption, etc.) can, at least to some extent, also be used to protect vehicles against external threats.

External network traffic can also be attacked or misused in ways that are more specific for vehicle communications. It may be possible to *replay* correctly signed transmissions for example at other locations, or to *modify* one's own messages and alter information such as position, speed and direction to get better service or cause confusion. Such attacks may be hard to detect since the messages can be properly signed by the vehicle sending out the information.

If stolen (i.e. copied) certificates are used, their use may be undetected due to problems with immediately being able to verify CRLs (assuming the theft is discovered in the first place). Another vehicle's identity may be used, for example to obtain new firmware versions with enhanced functionality. Other attacks include *impersonation* such as identity theft and Sybil attacks where multiple identities are used for example to spread false congestion information.

Confidentiality is not a major issue for VANETs and broadcasts of signed clear-text messages will to a large extent be used. The exception is closed group communication where messages from some applications may be designed to use encryption. Clear-text communication has many advantages, reduces cost of devices and may help to meet real-time requirements.

Vehicle identities could be confidential, although they may be necessary to use in some types of communications. If so, they may give information to third parties, for example reveal drivers' locations. A scenario we do not want to see is that people start to listen to vehicle broadcast messages in cities and cooperate over the Internet to follow vehicle motions and publish their locations and movements on web sites. (This is done today for ships through the AIS system at *marinetraffic.com*.)

5.3 Demonstrated security threats

The lack of security in today's vehicles has recently been demonstrated by several research groups. By connecting a device to the in-vehicle network, for example through the On-Board Diagnostics port OBD-II, it is possible to issue arbitrary commands to the vehicle. Many of the practically demonstrated attacks are performed through this diagnostic interface, which until today has required physical access to the vehicle. However, in a near future, the same attacks will be possible to perform through a wireless connection, with similar results.

5.3.1 Compromised ECUs can send arbitrary messages

A team of researchers from University of Washington and University of California have practically shown that infiltrated ECUs in a vehicle can be used to send arbitrary messages to the CAN bus [46]. They have demonstrated this functionality both in the lab and in road tests.

They started with listening (sniffing) the CAN bus to determine how units communicated and what messages were sent. Replay attacks were then trivial to perform. To enhance the functionality in an ECU, they performed reverse engineering by dumping the ECU code over the bus using a third-party debugger. This enables an attacker to silently "enhance" the functionality of ECUs and still keep its original functionality. They have demonstrated several different attacks: taking total control of the radio (user could not turn it off), produce various sounds in the vehicle (the audio component is used for warning sounds), display arbitrary messages on the instrument panel, open and close doors, honk the horn, disturb engine functions, lock individual brakes, control the A/C, etc.

They also demonstrated that the breaks could be released while driving, making the driver unable to break. These attacks were done locally, i.e. with physical access to the vehicle, but with the correct software, remote attacks with similar results are possible to perform (see below).

The results are interesting but maybe not surprising since the internal networks and protocols lack all kinds of security, but they clearly show what the outcome could be if a node in a vehicle becomes compromised.

5.3.2 Attacks via the media player

The same team of researchers from University of Washington and University of California recently also showed that serious attacks can be performed without physical access to the vehicle [14].

A weakness in the media player was used to gain access to the local CAN bus. The media player they tested came from a large third party supplier and it plays CDs and also accepts MP3 and WMA files. Media players in vehicles normally have access to the CAN bus, for example to be able to change the sound level when the vehicle's speed changes. Using reverse engineering, they discovered that it was possible to do a buffer overflow attack when playing WMA music. The weakness allowed the attackers to create a CD which contained specially crafted music that the player plays perfectly, but also silently performs a buffer overflow attack which can send arbitrary commands on the CAN bus. It is not hard to imagine that music containing such viruses would be popular to distribute on the Internet.

The team also found similar vulnerabilities in the Bluetooth stack, which allowed any paired device to execute arbitrary code in the Telematics unit.[1] The telematics unit also contained vulnerabilities in its cellular communication protocols opening it up for remote Internet attacks, similar to the Bluetooth attack.

The attacks show the necessity to have good security practices in place when designing software for ECUs and equipment connected to the internal networks. In this case, the media player and the telematics unit in the vehicle had traditional buffer overrun bugs, enabling arbitrary messages to be sent on the internal CAN bus. In the future, it is likely to expect that many ECUs and subsystems like these contain software implemented by third parties, over which the car manufacturer have limited or no possibilities to know all details and be able to evaluate the implementation.

5.3.3 Attacks via wireless tire pressure system

All vehicles manufactured in the U.S. after 2007 are required to have a Tire Pressure Monitoring System (TPMS) installed. Rouf et al. [60] have demonstrated an attack where the wireless communication (RF transmitter) between the vehicle and the sensors in the tires are compromised.

Each tire sensor has a unique 32-bit identifier to prevent vehicles from displaying messages from other vehicles. The radio communication between the tires and the vehicle turned out to be unprotected and reached around 40 meters with low-cost antennas and amplifiers. They also showed that remote spoofing of messages was possible, and all messages with correct IDs were unconditionally accepted, triggering warning messages for the driver. The equipment used was available on the market for around $1,500.

During their tests with spoofed messages, they also managed to completely crash the ECU receiving the tire-pressure messages, and the only recovery possible was to return the vehicle to the repair shop and have the ECU replaced.

They also concluded that the 32-bit identifiers used to uniquely identify each tire pressure sensor, can be used to track vehicles and therefore creates privacy problems for the owners.

5.3.4 100 cars disabled remotely

In March 2010 media, including *wired.com*, reported that more than 100 cars were disabled remotely. Until the cars were fully paid for, the cars were controlled by the car dealer and had functionality to be remotely disabled if the customer slipped with the monthly payments. 1,100 cars were reported to have this functionality.

[1] We discuss in depth some of the research done within the SysSec consortium related to the Bluetooth standard in Appendix A.

This day, a former disgruntled 20 year old employee used a fellow employee's account to log in to the dealership's computer system and disabled more than 100 vehicles for their customers. The ignition system was disabled and he managed to honk the horns in the middle of the night. The only way the customer could turn off the horn was to remove the battery.

This security problem again shows the impact of installing third party applications in vehicles. There may have been limitations to what the system could have done, but we have also seen that the vehicles lack protection if an ECU or a connected system wants to send arbitrary messages.

6 Securing the external communication

With external communication, we mean all V2X connectivity as well as Internet connectivity offered to vehicles. As described in Chapter 5, there are many protocols and potential weaknesses that need to be addressed. External traffic can be subject to traditional network attacks described by CERT and many other sources. All protocols, at all levels, must ensure they have proper protection against:

- *Eavesdropping:* attacker reads (copies) data from the network. For wireless communications, sensitive and confidential data needs to be encrypted. An example of confidential data can be the software (firmware) for the ECUs.
- *Deletion and modification:* data is dropped or modified during transmission. Application-level protocols must either have their own detection mechanisms or ensure that the underlying network protects the data. Modification of traffic can be done by a man-in-the-middle, for example by abusing link-level protocols and changing the routing of traffic.
- *Injection and data origin spoofing*: new traffic is injected into an ongoing session either by a man-in-the-middle or by a remote attacker. In a repair shop, an attacker may insert additional diagnostics messages into an existing session in order to steal data, change a vehicles settings or configuration, or possibly even update the firmware with new "enhanced" functionality.
- *Impersonation*, also called identity spoofing. The identity of other trusted users or devices may be spoofed with similar results as for traffic injection.
- *Recording, replay and delay:* data from older sessions are reused. It could contain old authentication information or the attacker could re-

send information that was intended to be used during other circumstances, at other locations or reuse data that should have been valid only during a short time period. Even authentic data which is immediately relayed over the Internet to another location and replayed, could cause confusion and problems.

- *Denial of Service attacks:* often done by flooding networks or by using known vulnerabilities that cause nodes to crash or exhaust their resources.

- *Malicious traffic:* Malformed packets that should not normally be seen on networks but can cause systems to malfunction, crash or execute arbitrary code.

6.1 Certificates and authentication

Authentication is an important security function needed in most types of communication and the use of public key signatures has been proposed to simplify deployment of a large authentication system. Several organizations can help in distributing certificates in their local region. However, the deployment is still far from trivial and vehicles may have to be constantly connected to a central system to be updated about revoked certificates (i.e. to check or download CRLs).

Although certificates have been proposed and are believed to be the way to implement authentication [20], there are still problems to be solved with this technology. There will be lots of issuers of certificates and vehicles are not limited to country borders. Traditionally each country has had their own national organization issuing license plates, but it is far from clear that these organizations are willing to take over the role of issuing digital certificates and work with key generation and distribute certificate revocation lists (CRLs). In the US, it is not even a national issue but each state is responsible for its vehicles and vehicles are frequently crossing the borders. The frequency of changing keys and what (limited) lifetime they should have is still open for discussion. Similar problems exist with CRL distribution lists, their scope and how global or local they should be.

Proof of data correctness is often more important than identifying the sender. Data can be forwarded by many devices and it is the correctness of the data and not who forwarded it that is important. Some data can be trusted based on its contents given the conditions for when and how it is sent, even if it is sent by a node whose identity cannot be properly verified.

A longer discussion about authentication, the need for anonymity, distribution of temporary identities and use of pseudonyms can be found in Chapter 8.

6.2 Group communication

Since IEEE 802.11p lacks security and does not support a BSS (Basic Service Set) mode like traditional WLANs using an access point, this has to be taken care of at higher levels. The WAVE architecture addresses this issue and allows groups to be formed in a "software-implemented BSS mode" on higher levels. This allows closed groups to be formed where only participating members can exchange messages.

Traffic within a group are signed with keys handed out by the group leader. When vehicles leave the group, keys can either be changed or the group can continue to live for some time.

Vehicles are likely to be participants of several VANETs at the same time where groups are formed based on different properties. Group communication can, at least to some degree, solve the problem of network flooding when forwarding messages. By introducing group leaders, communication within the group can be more efficient, and group to group communication is done via group leaders only. However, the creation of groups is not trivial and algorithms for selection of group leaders are still being discussed.

6.3 The ITS station standardized by ETSI

ETSI has standardized an "ITS station" which is a standardized communications node intended to be used by all V2X communication systems [2]. The ETSI ITS station reference architecture describes the functionality and tries to standardize how V2X communications should be performed. The standard specifies the architecture for hosts, gateways, routers and border routers. The (ITS-S) gateway is a bridge (or protocol converter) connecting two protocol stacks at layers 5 to 7 and in order to achieve this, it requires two full protocol stack implementations. It covers functionality like classification, prioritization and channel assignment and maintenance which can be requested by applications.

The use of such a standardized platform may make it possible to certify highly specialized communication nodes and make them highly resistant against failures, including security-inflicted problems.

Limited DoS-protection can also be part of such a node if the hardware has proper protection against what can be transmitted. For example, there could be hardware-enforced limitations to what can be transmitted and how many messages can be transmitted during a certain time period. For such functionality to be fully trusted, it is essential that it cannot be modified through software updates or by any type of failure of the ECU, i.e. that (at least parts of) the communications module is fully isolated from the rest of the ECU.

6.4 A framework for assessing security

Although there is a lot of research going on in vehicular systems, we have found very little research referring to models of the connected car and how to assess the security of emerging vehicle services, e.g. remote diagnostics, remote software download, and other Internet services brought into future vehicles.

The Car-2-Car Communication Consortium (C2C-CC) have created a reference architecture which is divided into three domains; the *in–vehicle*, the *ad hoc* and the *infrastructure* domains [12]. The in-vehicle domain is represented by the vehicle, its applications and mobile devices directly associated to the vehicle. The ad-hoc domain is represented by the vehicles and the road-side units, where the road-side units further can be connected to the infrastructure domain. In their architecture, the access network, the Internet, and possible nodes connected to the Internet are shown as part of the infrastructure domain.

A more detailed framework for security assessment has been developed in [43] which consists of *a model* for the infrastructure of the connected car and *a security assessment tree*. Such a framework can help to understand and evaluate how secure protocols and applications should be evaluated and designed in different vehicle settings. Since the connected car will contain a large number of services, communication technologies and network types, the assessment of security is far from trivial [41, 63, 69]. The proposed model together with the security assessment tree make it easier to identify the weaknesses of the system and the existence of threats both when designing new services and when assessing security as a whole.

This model is shown in Figure 6.1. *The infrastructure* is divided into two domains, the managed infrastructure and the vehicle communication domain. The managed infrastructure is further divided into five regions: automotive company applications' center, third party applications' center, trusted network, untrusted network, and the Internet backbone. *The vehicle communication* describes the possible means of communication with the vehicle. These concepts are further explained in the following paragraphs.

6.4.1 Managed infrastructure

The five regions of the managed infrastructure show different levels of trust which may require different protection mechanisms for transmitted data.

- *Automotive Company Applications' Center*. In the literature, the automotive company applications' center has different names, e.g. a *portal* or a *remote service center*. To summarize, it consists of a set of servers providing services to their vehicles. It holds necessary information about the vehicle, such as information from previous services (e.g.,

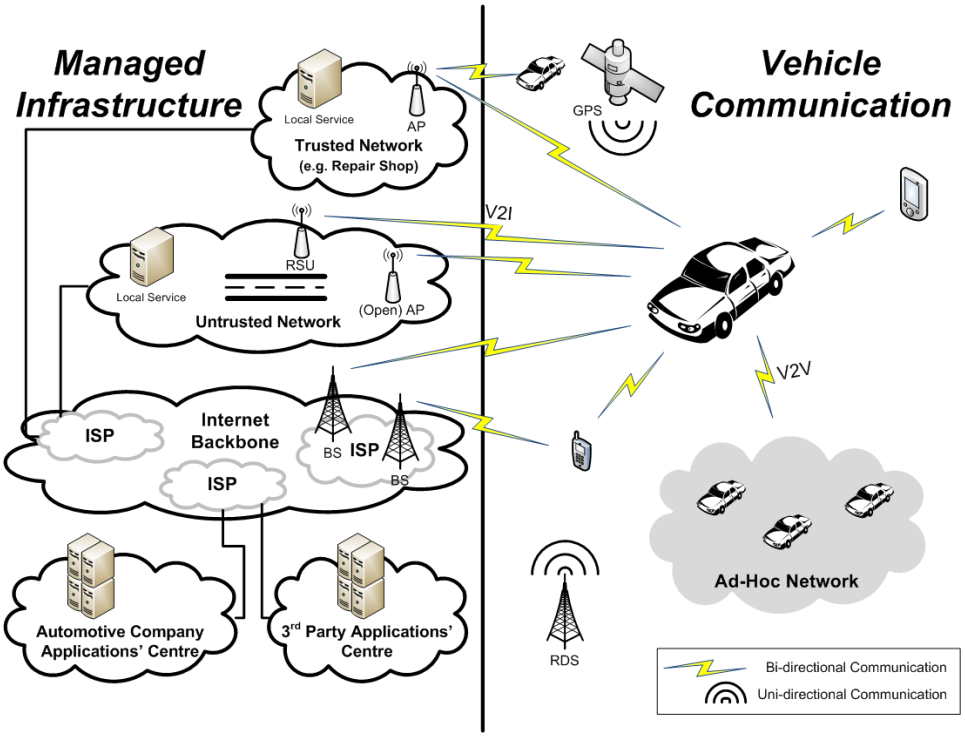


Figure 6.1: Communication scenarios and trust relationships

diagnostics data), configuration data, cryptographic keys, as well as new software available for the ECUs.

- *Third Party Applications' Center*. Apart from services provided by the automotive company, third party services can be provided to the vehicle. We could imagine that large "application stores" for vehicles will be available in the future. These applications can provide any kind of service to the vehicle.

- *Trusted Network*. Some networks can be considered to be trusted by the applications' centers and the vehicle. For example, a repair shop may be considered to be a trusted network by the automotive company and the vehicle. In delivering a service to this network, it may well be that some requirements in an implementation can be relaxed.

  An example where the security requirements in the implemented service can be relaxed is when performing remote diagnostics over a wireless network in a repair shop. If appropriate link layer encryption is applied, the security of the wireless communication could be considered to be equal to that of a cable. This will not be the case when the communication with the vehicle is performed through the Untrusted

Network; here end-to-end application encryption might be the only choice.

- *Untrusted Network*. All networks, except for the trusted networks, are considered to be untrusted. In these networks, the services provided to the vehicle have to be adapted to the hostile environment of the Internet. In the same way as for the trusted networks, other local services may also be provided in these networks.

- *Internet Backbone*. The Internet backbone, with its ISPs, is the core network for connecting the other four regions together. A backbone network is usually well protected and operated by network specialists in a networks operations center, NOC. Therefore, when network traffic has reached the Internet backbone, it is assumed in the model that it is unlikely that the data will be intentionally modified.

6.4.2 Vehicle communication

The vehicle communication domain describes the possible communication means between the vehicle and the managed infrastructure and with other objects. The following communications scenarios exist.

- *vehicle to wireless AP*. The vehicle can establish a connection to a wireless AP. All open APs (hotspots) are considered to be part of the untrusted network. Furthermore, a protected AP, where the vehicle needs authentication keys, can be available in both trusted networks and in untrusted networks.

- *vehicle to road-side units*. Road-side units, RSUs, can also be used to establish a connection from the vehicle to other networks in the managed infrastructure.

- *vehicle to cellular base stations*. A mobile data network, can be used to establish a connection from the vehicle to the Internet.

- *vehicle to mobile devices*. Mobile devices can be connected to the vehicle. For example, a connection can be established to a mobile phone, a laptop, or a PDA. Furthermore, the vehicle can also act as a gateway for the mobile device, so that the mobile device can reach the same network as the vehicle.

- *vehicle to cellular base station via a mobile device*. If the vehicle lacks the possibility to connect directly to a cellular base station, another mobile device with a connection to the cellular base station can be used as a gateway. One example is to use the driver's mobile phone.

- *vehicle to other vehicles*. Finally, the vehicle can connect to other vehicles and create a VANET. This V2V communication will be critical in future traffic- and safety-related services.

It should be noted that the description of the vehicle communication above is based on just one vehicle; any connected car will have the same communication surroundings. This means that the vehicle may possibly reach the managed infrastructure, via other vehicles or other mobile devices acting as gateways.

6.5 Using the framework to assess the security of vehicle services

From the model of the infrastructure of the connected car, there are different aspects that can be discussed regarding the V2V and the V2I communication. One of them is the security of the services delivered to the vehicle. Figure 2.1 presents a brief taxonomy of the security of these services. Four categories are described; the *actors*, the *V2X communication technologies*, *network paths*, and the *dependability and security attributes*. A description of them follows below.

- *actors*. Six different actors that can be involved in a service have been identified. Common for them all are that they have interests in how the service is being designed and delivered; the automotive company and the application provider can state requirements, the car owner and the driver can have concerns on how the data from a service is processed, the authorities can issue legal requirements, and an attacker can try to manipulate the service in an unwanted way.

- *V2X communication technologies*. A number of communication technologies are available for connecting the vehicle to other devices. Examples of these are listed. An extended list, including classifications of the communication technologies, can be found in [16].

- *network paths*. The service may be delivered to the vehicle using one of several network paths. The model describes four possible network paths that the service can be delivered through (see Figure 6.1); the trusted network, the untrusted network, the Internet backbone, and an ad-hoc network.

- *dependability and security attributes*. To deliver the service in a secure and safe manner, the six attributes for dependability and security need to be considered [8].

From these four categories, an analysis can be made to further clarify how a service will work in the infrastructure, and also highlight the dependability and security attributes that need to be addressed in providing such a service.

With the security assessment tree, the problem with the vast number of issues that need to be considered in securing a service, is identified. It helps us to state requirements regarding security and provides us with a framework and a template for security assessment by identifying threats and communication patterns and to define countermeasures.

7 In-vehicle security

In this chapter we focus on the in-vehicle network, and architectures proposed to implement security into these networks are discussed. The CAN, LIN, MOST and FlexRay-protocols lack security functionality and were not designed with security in mind. All traffic is sent in clear-text and no protection against altered, spoofed or injected messages exists. When external communication is forwarded to these networks, appropriate security mechanisms must be in place. Border protection of vehicular networks can, and should, be done by the communications unit (CU) that handles external communication, but internal protection mechanisms must be present as well. Ideally, each ECU should be able to protect itself against malicious traffic and be able to identify the origin of all messages. However, there is a long way to go before these goals are fulfilled.

7.1 The need for a planned architecture

By having a proper internal architecture similar to what is proposed by the Evita project [1], some threats can be eliminated or at least the consequences, from a security point, can be limited to the subnet (bus) from where it originated, see Figure 7.1. There are several attempts to deal with this problem, but the main constraint that limits the applicability of solutions is the cost of the solutions, especially if it requires more powerful ECUs. Without these limitations, conventional security mechanisms used to protect corporate networks from malicious Internet traffic could be used directly.

A defense-in-depth approach for securing the vehicle is discussed by Larson and Nilsson [48]. They discuss how to prevent unauthorized access, use IDS and logging mechanisms for detection and IPS systems as a countermeasure, use honeypots for information retrieval and detection of new at-

tack methods, and the necessity of traceability to perform recovery. In [52], Nilsson and Larson extend this discussion.

A well-planned architecture with traffic separation, internal firewall functionality preferably in all ECUs with possibility to detect malformed, spoofed and incorrect messages is the overall goal. This can be accomplished in different ways. Some approaches take just a few steps in this direction, other try to solve the overall problem but with higher complexity and cost as a result. In the following paragraphs we investigate some proposed solutions.

7.1.1 Leave access control to higher layers

Chavez et al. suggest the use of the security services of the OSI Reference Model (ISO 7498-2) for securing the CAN-protocol [13]. The OSI model describes five security services, confidentiality, integrity, authentication, non-repudiation, and access control. According to this, they propose that access control should be taken care of at higher layers in the protocol, that integrity should be enforced by using hash algorithms, and that confidentiality should be enforced by using RC4 encryption of the CAN-frames. The remaining two OSI services, authentication and non-repudiation were not considered to be useful in this context. Apart from resource constraints with encrypting all messages, key distribution between internal nodes is a major problem.

7.1.2 Firewall functionality in Gateway ECUs

The EVITA project has designed a reference architecture that is useful when discussing vehicular networks. The vehicle network is divided into sub-networks controlled by Gateway ECUs. These gateways can have some firewall functionality built-in to protect their network from unwanted communications to and from the other networks. They can make sure that most messages on the local network remain local and that only selected messages are sent to other networks. Similarly, they can restrict what messages may be forwarded to the local network.

External communication is mainly done through the Communication unit (CU) but with some exceptions: USB and Bluetooth communication is performed by the multimedia subsystem.

This model is not entirely new. Car manufacturers (e.g. Volvo, BMW and Volkswagen) already use multiple internal buses for separation of traffic, although they use their own proprietary designs. If such a standard will emerge or whether each manufacturer will use their own, remains to be seen. The model is still very useful and can be used in discussions of separation, firewalls, IDS functionality, etc., and it is used as a basis for discussion in this deliverable.

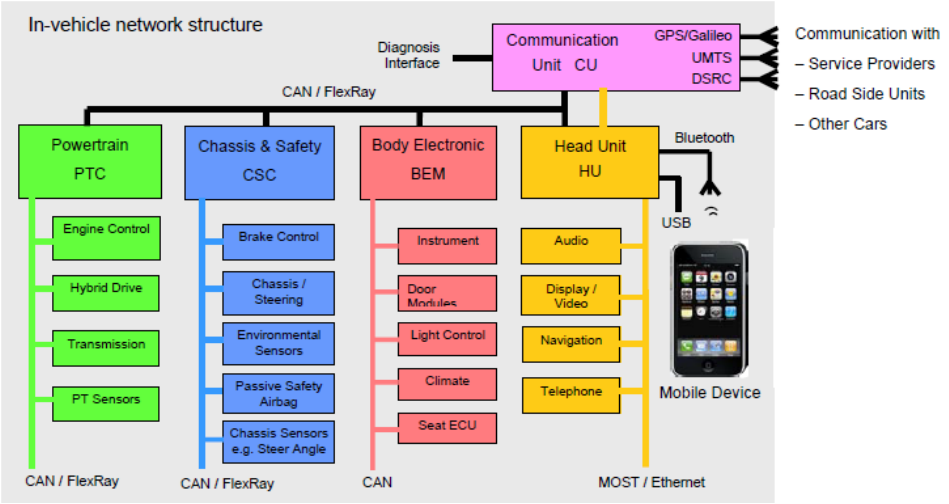


Figure 7.1: Evita project use-case architecture

7.2 The SeVeCOM project

The goals with the SeVeCOM project have been to create a component based security architecture that can have a very long lifetime and easily be extended and changed when new and unforeseen threats emerge [42]. They have concentrated security functionality into a *Security Manager* which is responsible for all security in all modules in the vehicle. It has hooks to the communication stacks (possibly created by different vendors) and can request to inspect and modify traffic at all layers. The security manager can implement services like firewalling, identity management and inspection of signatures in incoming messages. It is also the interface to a hardware security module (HSM or TPM) which is physically protected (tamper resistant) and stores private keys and can perform cryptographic operations using these keys.

The idea with a separate security manager is that it is relatively easy to change functionality like cryptographic algorithms, keys, firewall- and IDS functionality, in all communication stacks. Application programmers may, at least to some degree, therefore be relieved of considering all aspects of security and how to implement security services. It should, by using this kind of architecture, be possible to implement security into an existing network stack with minimal changes.

The security manager contains components for identification, trust management and privacy management. These components may subscribe to certain events from the hooks, for example to special types of messages.

The ideas presented are interesting and, as the authors point out, the use of hooks is similar to how Linux interfaces with its network stacks for similar tasks.

7.3 Security mechanisms

To secure internal communications, several traditional mechanisms have also been proposed. These include message authentication codes (MAC) for traffic integrity, firewalls both for external traffic and for internal traffic implemented in gateway ECUs, use of intrusion detection systems to detect unusual activities on the networks, certificates for identification of various devices (vehicles, road-side objects, drivers and ECUs). These and many other mechanisms are described in the following text. Figure 7.2 provides a summary of papers describing protection mechanisms in different areas.

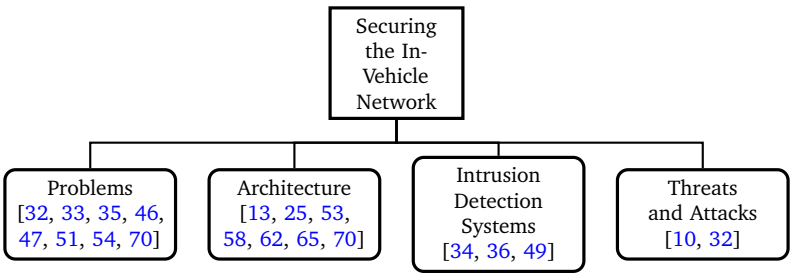


Figure 7.2: Protection mechanisms for in-vehicle networks

Message Authentication Codes, MACs, can be used to provide integrity of the messages. However, since internal security is not standardized, this means that in order to implement MACs, it is necessary to modify the existing protocols (see Section 7.3.2). Other approaches that have been proposed is to create new security architectures, for example with gateways limiting cross-traffic between different parts of the vehicle. However, these approaches still have to be evaluated considering the limited resources of the in-vehicle network. In addition, they do not explain to application writers how security should be implemented, what traffic needs to be filtered in ECUs and gateway nodes, and what subnets should be implemented in their particular vehicles and models.

Firewalls can be used to protect both in-vehicular traffic and traffic from external sources. For example, Wolf et al. [70] briefly discuss the concept of an internal firewall in each ECU, but we know of no attempts to really introduce a firewall where traffic is filtered by ECUs. We also note that out of the four protocols used for the in-vehicle network (CAN, LIN, MOST, and

FlexRay), almost all research have addressed CAN and very little work with the other protocols have been done.

Intrusion Detection Systems have been studied to some degree, and both anomaly-based and specification-based IDS have been suggested for the CAN protocol. However, no approaches have been found for other protocols, and since FlexRay also lacks appropriate security mechanisms and eventually will replace the CAN-protocol, an IDS for FlexRay should also be investigated. A longer description of IDS systems can be found in Section 7.5.

7.3.1 Trusted communication groups

Groll and Ruland [25] propose an architecture where they divide the communication into trusted groups. All ECUs within a trusted group share the same symmetric key to encrypt and decrypt the communication. A Key Distribution Center (KDC) within the vehicle is used to create and distribute the symmetric keys for these trusted groups. The trusted groups are defined by access control lists, ACLs, and are signed by the automotive company. One ACL is stored in each ECU and defines the trusted groups that the ECU belongs to.

To distribute the symmetric keys for communication within the trusted groups to an ECU, the ECU sends its ACL to a Key Distribution Center, KDC. After the KDC has verified the signature on the ACL, the KDC sends back the symmetric keys for those trusted groups defined by the ACL. To protect the distribution of the trusted group keys, asymmetric encryption is used between the ECU and the KDC. The asymmetric keys needed must also be signed by the automotive company.

A more generic approach has been suggested by [62]. This is accomplished by the introduction of a data management system, DMS. Instead of letting all ECUs exchange data with each other, data is stored in specific nodes within the vehicle. By using a DMS for storing data, security mechanisms such as access control could be enforced on access of data and to ensure data integrity. The method also opens for the possibility to store a global state to a protective storage in the case of an accident. Three different approaches to deploying the DMS were investigated: a centralized approach, a distributed approach, and a hybrid approach, in which a DMS is deployed for each sub-network. The hybrid DMS approach was found to be the most attractive.

7.3.2 Authentication of ECUs

Wolf et al. [70] suggest ways to improve the security of the communication by requiring authentication of the ECUs and by encrypting the communication. First, each ECU has to be authenticated by the gateway by means of

a certificate. After authentication, the ECU will receive a symmetric encryption key that is shared with other authenticated ECUs on that local network to make secret data exchange possible.

7.3.3 Message authentication

Nilsson et al. [53] propose the use of a MAC for providing data integrity and data authentication in the CAN communication. To achieve this, a 128-bit key is shared between the two communicating ECUs. The MAC is calculated over four consecutive CAN-messages and the resulting MAC is divided into four 16-bit blocks and transmitted in the CRC-field of the next four CAN-messages. The protocol introduces a delay before the data integrity and data authentication can be verified. In total, eight messages are needed for the verification to be completed. Two of the remaining challenges with the protocol were that if the MAC calculation fails, the actual individual message that was wrong can not be identified, and that there is no protection against replay attacks.

7.3.4 Authentication of multiple destinations

An approach to provide authentication of messages for time-triggered applications is proposed by Szilagyi and Koopman [65]. A protocol was designed to be able to authenticate multiple destinations at the same time, which requires that each pair of communicating nodes share a symmetric encryption key. These keys are used for calculating the MAC over the messages for each destination. Each MAC is further stripped down to a few bits and concatenated to the end of the message. Since it is easier to forge a message with only a few bits of the MAC available, the authors propose that authentication is provided by successfully verifying the MAC over a set of messages. For the two types of messages investigated, state-changing messages and reactive control messages, an upper boundary of the probability of performing a successful attack is discussed.

The proposed protocol also has protection against replay attacks. The protocol is further discussed in [66], where an analysis with the help of simulated attacks is provided.

7.4 Hardware security modules, HSMs

A hardware security module (or a Trusted platform module, TPM) contains security-critical functionality needed by other components of the car, such as to protect private keys (used in asymmetric encryption), to distribute session keys, and to sign messages. It contains memory, a processor and software capable of performing basic cryptographic operations and preferably also a

good random number generator. It should ideally be a tamper-proof device and be protected against physical access, i.e. that all attempts to access its contents should render it useless.

A valid and reliable clock is needed by many services to avoid replays of messages and to detect reuse of older messages in other environments. The HSM module is a good place for such functionality and a timestamp can be applied to the message at the same time as it is digitally signed. This prevents individual ECUs which are compromised from sending and reusing old messages.

HSM functionality is likely implemented in special hardware similar to the well-known IBM 4578 cryptographic processor with tamper-resistant protection. It should have a well-defined API that can be used by ECUs to perform crypto-related operations,such as to sign and verify signatures. Simpler devices like smart-cards may be useful in some situations, but they lack functionality such as offering a trusted time source. However, smart cards and RFID cards offering crypto-operations can be used in other situations such as when identifying owners, drivers and service technicians.

HSMs have also been suggested to be used by internal ECUs in order to guarantee execution of authentic code. The functionality can be similar to newer Windows laptops, where a TPM chip together with the first boot-loader (BIOS) verifies the integrity of the software before storing it and only authentic (genuine) software will be executed. This can be done in steps and even include signed applications from third party developers. All non-trusted and modified software, including malware, will automatically be discarded by this system. There are many advantages with this solution, although the drawback is the increased complexity of the ECUs.

7.4.1 Extending the TPM functionality

An extended version of TPM functionality has been suggested by Oguma et al. [58]. They propose an architecture where only ECUs with successfully validated software will be able to exchange symmetric keys for further encrypted communication. It uses three components:

- a center outside the vehicle,
- a master ECU within the vehicle, and
- all other ECUs within the vehicle.

The center stores the information regarding all vehicles, and the master ECU is used to do local attestation since the center might not always be reachable. The master ECU holds a list of hash values that are valid for the software running on each ECU in the vehicle. Instead of using asymmetric encryption within the vehicle, a Key Predistribution System, KPS, is used

where each pair of ECUs share a separate key. After the attestation process has been performed, encryption keys are generated for each pair of validated ECUs using the KPS. Both encrypted messages and signed messages are supported. These messages also hold information to prove that the ECU has been validated and a counter to protect against replay attacks. Lee et al. [50] further discuss the attestation-based security architecture. By using ProVerif, they propose a way to formally verify the protocol.

7.4.2 EVITA HSMs

To offer different levels of security functionality and performance, three different types of hardware security modules have been defined within the EVITA project: full, medium, and light.

- The full module is deployed in one or two high-performance communication ECUs in the vehicle, and has hardware for asymmetric cryptographic operations needed by complex and demanding external communications such as V2X communication. It is likely used only in central communication gateways.
- The medium module is used in two to four central multi-purpose ECUs, likely Gateway ECUs isolating traffic between internal networks (see Section 7.1.2). It supports asymmetric cryptographic operations, but lack hardware support and is less powerful than the full module.
- The light module is needed for less powerful but still security-critical ECUs. It has only a hardware accelerated symmetric cryptographic engine, a hardware random number generator and a UTC clock. Typical use is in sensors and actuators.

7.4.3 Event data recorders

Event data recorders (EDR) are devices that record important events in tamper-proof storage similar to the black boxes used in aircrafts. It is likely that government agencies and vehicle manufacturers will require devices like this to be present when more advanced applications are introduced into vehicles. The EDR makes it possible to investigate reasons behind crashes and other safety-critical events.

7.5 Intrusion detection and prevention systems

Research on vehicular IDS systems has been targeting the CAN protocol. Both specification-based and anomaly-based detection methods have been investigated.

7.5.1 Specification-based detection

Larson et al. [49] propose and evaluate a specification-based IDS for the CAN 2.0 and CANopen 3.01 protocols. They conclude that, since these protocols lack information about the producer and consumer of messages, there is not enough information available for using network-based intrusion detection. Instead, they propose host-based detection, i.e. one detector is placed in each ECU. Incoming and outgoing traffic can then be investigated based on information from the protocol stack and the object directory of the CAN-protocol at the specific ECU. For the detector in the ECU, security specifications for the communication protocol and the ECU behavior can be developed.

From their evaluation of the specification-based approach, they conclude that the gateway ECU is the most important ECU to protect. If the gateway ECU is compromised, all attacks they investigated could be performed. Unfortunately, performing detection in the gateway ECU is harder than in ordinary ECUs since the detectors for the different interfaces at the gateway have to cooperate to detect certain attacks, e.g. to detect lost or modified messages.

7.5.2 Anomaly-based detection

Hoppe et al. [36] demonstrate an anomaly-based IDS for the CAN protocol. In contrast to the specification-based approach by Larson et al. [49], where the IDS is placed in the ECU, they listen to the network traffic on the CAN-bus. By looking at the rate of how often specific messages are transmitted on the bus, and comparing that to what is considered to be normal, deviations of the number of transmitted messages can be detected. This was exemplified by investigating the system that detects physical vehicle break-ins. When the anti-theft alarm is activated, the system sends messages to the lights of the vehicle to turn them on and off, so that they are flashing. An attacker does not want these lights to be activated, but since the CAN-bus is a broadcast network, messages sent by the alarm system can not be deleted (except possibly in gateways). Instead the attacker has to create new messages to turn the light off as soon as it is lit. These new messages will be a deviation from the normal number of messages sent, and detected by the anomaly-based IDS.

The SeVeCOM project [42] recommends vehicles to use an anomaly based IDS system internally. However, they do not address in detail how and what the IDS system should do except that “appropriate reactions should be taken to get the system back to a secure and safe state”.

7.5.3 Handling intrusion alerts and IPS systems

One crucial issue with intrusion detection is to decide what to do with an alarm. It may be possible to send the alarm to a central portal where a security officer takes care of it. However, it may not be realistic to assume that the portal should have such resources for the large number of cars connected. Further, the car may not be continuously connected to the portal for obvious reasons. Thus, it seems more realistic to inform the driver of the alarms. Such an approach is proposed by Hoppe et al. [34] where various security-related events can be presented to the driver. Depending on the severity of the event, different methods are used: visual for non-critical events, acoustic for critical events, and haptic for severe events.

They also propose an "adaptive dynamic decision model". By using the sensors in the vehicle, the environment of the vehicle can be evaluated at the time of the alert. If the currently used ways of alerting the driver is not considered to be enough, the alert-level can be increased.

Since the communication within the vehicle is safety-critical, discarding the wrong message may have catastrophic effects. An attacker may also use the fact that an IPS system is present and force it to make incorrect decisions. Hoppe et al. discuss the problem of intrusion response and point out that an active response system might not be allowed to actively make decisions in the vehicle due to legal requirements for safety-critical systems [36].

7.5.4 Honeypots

A honeypot is another security mechanism that may be applied to collect information and to analyze attacks against vehicles. To our knowledge, only one such approach has been described so far, by Verendel et al. [67]. It is suggested that the honeypot is attached to the gateway node in the vehicle and simulates the in-vehicle network. The data collected from the honeypot can then be sent to a common portal and analyzed in detail. The purpose of this is to learn about new attacks and distribute solutions as early as possible.

A very important property of the honeypot is how realistic the simulation of the target is. If the simulation is not realistic enough, the attacker will realize that he is not attacking a real vehicle. However, making a realistic honeypot may be very hard and they address this by proposing three different models. Another complication is that, for security and safety reasons, dedicated hardware should be used for the honeypot and not a vehicle in use on the streets.

8 Authentication and privacy

8.1 Authentication, certificates and access control

In the earlier discussions, authentication and proof of origin of messages have been discussed several times. There are many situations where proper authentication of the communicating parties is needed and where we want to guarantee the integrity, authenticity and authorization of senders and messages. We have identified many situations where proper authentication of communicating parties is needed:

- *Authentication and integrity protection of internal messages* on the CAN, LIN, MOST, and FlexRay buses. The receiving ECU wants to know that the message is authentic, correct (not modified, replayed) and sent from an ECU which is authorized to send such messages.
- *Verification of authentic software* before executing code in ECUs, for example in combination with hardware security (HSM) modules.
- *Identification of V2X traffic and/or objects*: to be able to identify roadside objects and other entities. In many situations it may be enough to be able to verify the correctness of a message without identifying the party transmitting the message. Examples include transmission of warning and informational messages to other vehicles and when receiving information about speed limits.
- *Authentication and integrity protection of data* sent to and received from external systems and units such as diagnostics equipment in repair shops, when performing remote software updates, when changing configurations of vehicles, etc.
- *Confidentiality protection* of both in-vehicle and external communication. Closed communication V2X-groups may for example require encryption to preserve message confidentiality and integrity.

- *Identification of remote traffic* to and from servers and services that the vehicle, the driver or one of its passengers is using. Traffic from third parties, for example requests from agencies tracking vehicles for automatic road toll payments, also need to be properly authenticated.
- *Authorization of persons* such as owners, drivers, passengers and service technicians interacting with the vehicle. Application examples include permission to update a vehicle's software, to drive the vehicle and to order new functionality to it. Other services may be based on the driver's identity such as insurance coverage and payment of parking fees.

8.2 Trust - different privilege levels

Trust is a complicated matter in the vehicular domain. As previously described, different entities involved may have different views of trust and privacy. It also depends on what application is being used, as some applications can be more trusted than other.

We have identified many different persons or identities that need to interact with the vehicle during its lifetime:

- *The manufacturer* of the vehicle. They will continue to offer services to the vehicle during its full lifetime, not only in repair shops. Critical software updates have to be applied more or less immediately, not two or three years after a vulnerability has been discovered and the car eventually visits an authorized repair shop.
- *The owner* of the vehicle. He/she should have access to most, but not all, functions in the car. It should also be possible to delegate privileges to other users to configure some settings in the vehicle. The owner may change over time and transfers of ownership must be handled.
- *The driver(s)* of the vehicle. Many services will in the future be based on who is driving the car.
- *Passengers* and authentication of passengers to access various applications and (remote) services, for example from e-commerce servers. Some services may be available in the car based on a passenger's identity, for example multimedia contents or navigational software.
- *Authorized technicians/repair shops* should have access to most of the vehicle's data, but not necessarily to private information related to the owner, driver or passengers. The technicians should not be able to use their identities in, for example commercial activities, nor to access personal information such as driver's behavior, GPS logs containing vehicle location history, or to install non-authorized software.

- *Third party application suppliers* may be granted access to certain parts of the vehicle network, but not to all. This may be a rather complex issue to solve, similar to privilege levels offered for example for Android devices.

- *Trusted authorities* who may need access to certain data, for example after an accident or a crash or in real-time to track vehicles for various reasons; road tolls, parking fees, etc.

8.3 Privacy and identity theft

There are many examples of information that may be sensitive and should not be universally available: Insurance companies may want to track driver behavior and may adjust insurance fees based on driving patterns. Police cars and road-side objects may automatically collect information about vehicles and could automatically issue fines when speed limits are exceeded. Vehicles and individuals may be tracked using GPS information and from communication with road-side objects and other vehicles.

A vehicle's or a person's identity may be stolen, e.g. a Trojan planted in a vehicle may be used by a remote attacker to read, modify and even send data to a third entity pretending to be (the compromised) owner. It may be possible to use the victim's identity in real-time transactions to sign messages that actually belong to someone else. One example could be when a message about a road toll is sent to the attacker's vehicle, it is immediately forwarded over the Internet to a compromised vehicle to be signed, and then sent back to the attacker to be forwarded to the road-toll system. The consequences may be that the victim is fined for traffic violations, parking tickets and road tolls. An attacker with access to many compromised vehicles may easily perform all kinds of Sybil attacks.

Information from earlier drivers may also be present in the car and may potentially be reused or stolen. Different approaches have been suggested for how to maintain privacy of the vehicle and still be able to solve the problem of non-repudiation requirements. In short, all solutions conclude that keys used to sign messages should be short lived and changed regularly. This prevents stolen identities from being long-lived, although they do not solve the problem with real-time access to vehicles.

8.4 PKI and certificates

Certificates are likely to be used in all V2X applications that require authentication. Certificates can be distributed in some, but not all, broadcast messages or when new vehicles are identified by the sending node (see below). Despite the complexity with generating and distributing certificates,

it is currently the only investigated alternative for authentication of vehicles and road-side objects.

8.4.1 Group signatures

Some applications, as mentioned in Section 6.2, may want to communicate in closed groups. Traffic should be encrypted and only certain vehicles are allowed to participate, for example those subscribing to a particular service or those of a specific car brand.

Traffic within a group should be signed with keys handed out by a group leader. When vehicles leave the group, keys can either be changed or the group can continue to live for some time. Group signatures may be promising but more work is needed before they can be used in practice [42].

Group communication can also be accomplished by getting a temporal symmetric key by an RSU when entering a region. Only authenticated vehicles get the key and are able to communicate. Techniques like this may lessen the burden on vehicles to authenticate all other vehicles when reaching populated areas.

8.4.2 Certificate revocation

Even if the identities of vehicles and road-side objects are verified using certificates, they do not guarantee the correctness of the device. A road-side object may be manipulated to send arbitrary information although its identity can be verified by vehicles. To limit the problem, certification revocation lists (CRLs) must in some way be distributed and be available to all vehicles. Researchers are investigating ways to implement more efficient distribution of large lists and methods to reduce their size. Whether they will be efficient enough to work in reality even in densely populated environments remains to be seen.

Certification revocation lists are problematic to deal with since many organizations in different countries need to be involved. It also requires vehicles to either check all certificates on-line or to download huge lists of revoked certificates and keys. To make the list of revoked certificates shorter, the lifetime of certificates could be made shorter. The drawback is that vehicles then would have to connect to the CA more frequently to get new certificates. In some countries it may not be a problem, in other it may be. Vehicles lacking proper keys are therefore not likely being able to participate in communications with other vehicles, but they can still receive information from its surroundings. CAs may also have the possibility to tell vehicles to erase keys from its tamper-proof memory if they believe keys are compromised or misused. CRL lists may also be distributed locally when a malicious vehicle has been detected.

8.4.3 Implementation issues

ECUs have limited cryptographic capabilities and the cost for performing such operations can be substantial. Most researchers agree that conventional X.509v3 certificates are too large for efficient and fast communication and the certificates to be used should be a subset of this standard. Instead of using RSA/DSA signatures, ECC (Elliptic Curve Cryptography) will likely be used which is substantially faster, although it is not unreasonable to believe that vehicles still need dedicated hardware to be able to verify signatures in real-time. One potential problem with using ECC is that patents guard the technology.

Vehicle beaconing may be used for some services, e.g. vehicles will broadcast some information to all its neighbors such as its location and speed, for example every 100 ms. On a crowded road, a vehicle may receive hundreds of messages per second to verify. And for each new vehicle, the certificate is also needed in order to verify its origin. It has been suggested [11] that certificates are attached not to every beacon transmission, but to a fraction of them in order to minimize overhead. Nodes, e.g. road-side objects may also delay certificate delivery in their messages until a new node is detected, a scheme that would work quite well in smaller networks, which is a situation that is likely to exist for some time until all vehicles are equipped with this technology.

There are also other situations and applications where checking the signature can be omitted. It may be informational messages that are not essential for the vehicle, or it can be situations where on-board sensors and other systems can verify whether the message is likely to be authentic.

Private keys must be properly protected, and a hardware security module (HSM) is a solution likely to be used which offers tamper-resistance and makes it hard to steal other vehicles' identities.

8.5 Pseudonyms and privacy issues

It is important to have a system that can prevent vehicles from *being traced* (i.e. follow its actions for example on the Internet) and that can *preserve the privacy of communicating parties* and make it impossible for an unauthorized party to link the vehicle to its driver or owner. To implement this, pseudonyms must be supported. With pseudonyms, the real identity of the object is hidden for, for example, other vehicles and road-side objects. Pseudonyms require that the system supports short-lived public key certificates. These pseudonym certificates must be issued by CAs and are therefore not trivial to implement [24]. One possibility is to equip vehicles with a number of pseudonyms when it is in contact with the CA and give it certificates that can be used for a short while.

The pseudonyms must be signed by a CA to allow an authorized entity (an authority) to track the real identity of a vehicle, if needed. The system must support non-repudiation to make it impossible for an entity to deny having sent a message, for example if a vehicle fabricates warning messages to other vehicles. The use of many pseudonyms makes handling of ACLs more complicated since caching of certificates becomes even less effective.

A similar privacy issue is that the communicating vehicle's MAC address (and IP address, if used) can be used for identification since it is unique for the vehicle. When a vehicle decides to use another pseudonym certificate for identification, the addresses need to change as well. The solution can be to use random addresses from a pool earlier given to the vehicle, similar to how pseudonym certificates are generated and used.

There are likely many other protocols that leak information which could be used to fingerprint vehicles. As shown in Section 5.3 the Tire Pressure Monitoring System (TMPS) sends out 32-bit identifiers that are unique to the vehicle, transmissions that can be picked up by anyone. To conclude, applications and systems that leak information which may be used to fingerprint and identify particular vehicles, may be seen as serious privacy threats.

9 Conclusions

Modern vehicles contain 50 to 100 networked computers, ECUs. Many new communication technologies and protocols exist or are soon ready to be introduced into the vehicular domain, new applications are designed, and all new vehicles will soon be connected to the Internet. Securing the connected car is a relatively new field and in this deliverable we have highlighted many of the security challenges we face.

New applications are soon to be introduced that cooperate with other vehicles and with road-side objects (V2X) which offer improved safety on the roads. Examples of such applications are informational messages from traffic lights, road signs and cooperation between vehicles in urban traffic. Other applications will be offered by the car manufacturers, such as remote software updates, remote diagnostics and services subscribed to by car owners and drivers. Yet other applications will be offered by third parties such as automatic road toll payments, navigational systems and automatically transmitted reports from vehicles about current road conditions. Security work must address how to implement these applications in a secure and safe way, how to isolate critical functions from all "nice to have" applications and implement protection against all attacks, internal as well as external, that may compromise the vehicle. In short, we must find ways for how to guarantee the safety of the vehicle at all times.

Standardization work has begun although the work has mainly focused on low-level communications and protocols. It has only to some degree reached security design and architecture, such as the ITS station work done by ETSI, and it still remains to be seen if the proposed standards will be universally accepted by the industry and by all countries in the world.

Internal security is more or less absent in vehicles. We have seen that any device with access to the internal buses can send arbitrary messages. This includes compromised ECUs that fail due to unexpected, but intentional, protocol problems and an attacker exploiting such a vulnerability can be in

full control of the vehicle. In this deliverable, we discuss several demonstrated security threats and their implications. We show why security is important and we describe several projects and standardization efforts and how they address different aspects of security. We have also commented on the usefulness of proposed solutions and our work covers internal (in-vehicle) as well as external (V2X) security.

To secure the in-vehicle communications, several traditional mechanisms are discussed. These include internal separation of traffic, the use of message authentication codes (MAC) to guarantee traffic integrity, firewalls both for external traffic and for internal traffic implemented in gateway ECUs, use of intrusion detection systems, use of certificates for identification of various devices (vehicles, road-side objects, drivers and ECUs) and the problems with distributing revocation lists (CRLs). We also discuss the use of tamper-proof hardware security modules (HSM modules) to speed-up crypto operations and offer safe storage of keys.

Our goal has been to write an objective and comprehensive summary of this interesting new field. We hope that our work will lead to an increased understanding and be an inspiration to future work to make road traffic even safer and more secure than what it is today.

A

Appendix: A security layer for automotive services

A.1 Introduction to the case study

As we discussed throughout this deliverable, modern vehicles collect information through different sources in order to improve the usage of the vehicle. On the downside, this leads to an increased attack surface that may enable an adversary to gain remote control of vehicles.

Adding security mechanisms to vehicles is a challenging task, because vehicles are commonly designed with safety requirements as opposed to security requirements. Further challenges arise as vehicles have typically real-time constraints, use broadcast networks often based on controller area network (CAN), embedded devices, and have some other characteristics that complicate security as discussed in §A.2.

In the main part of this deliverable, we have described the general state-of-the art and the problems that exist for the security of the connected car. To complement the breadth of the main part of the report, we would also like to describe in more detail some of the current research being done to improve the security in this domain. In this appendix, we wish to report a specific security architecture, developed within SysSec, to protect against potential attacks and introduce a communication framework that addresses the challenges raised above. We demonstrate how a smartphone can interact with a vehicle in a secure and safe manner. More specifically, we discuss how a smartphone can pair with a vehicle over a Bluetooth connection and then establish a *security session layer* that provides additional security guarantees—regardless of the security mechanisms already implemented in the physical layer (if any). We implement an asymmetric key-establishment scheme according to the Elliptic Curve Diffie-Hellman (ECDH) protocol (NIST 800-56A [57]) and a standardized ECC-192 curve (NIST P-192 [22]), For the data encryption, we use a symmetric encryption scheme (AES-128 [23]) based on a long-term shared secret. As a result of

our approach, the entire application layer is transparently secured by our security extension. We have designed our solution with performance constraints and real-time requirements in mind. Furthermore, we also took the capabilities of our target architecture into account (e.g., no input capabilities on the vehicle side, limited output capabilities, and lack of trusted execution environment on the mobile-device side). The design of our solution ensures that all these challenges are overcome.

We have implemented our approach for an electric powered two-wheeler (PTW) manufactured by Piaggio and show how the mobile device (an iPhone 4, in our proof-of-concept implementation) can interact securely with the in-vehicle battery-life controller. We performed a brief user study, which indicates that the solution is practical and easy to use. More importantly, our experimental measurements show that the overhead introduced by our security layer is small and reasonable. Interestingly, our approach is not limited to vehicles, but can be used in many other application domains where a smartphone needs to securely interact with an embedded device (e.g., keyless door opening or mobile payment).

In summary, we made the following three key contributions:

- We introduce a security framework for communication between a mobile device and a vehicle that is both theoretically sound and practical: we have implemented both a software simulator and a real implementation for an electric powered two-wheeler.

- The proposed framework is easy to use and simple, the user is not required to make complex interactions to pair a device with the vehicle in a secure manner. In particular, we conducted hands-on tests with actual users (i.e., non-security people) and collected feedback from a large Italian motorbike manufacturer. Both user studies found that the solution is easy to use and the manufacturer was convinced by the simplicity of our design.

- Performance tests suggest that our implementation has a small (almost absent) computational overhead because it leverages a digital signal processor (DSP) to maximize the cryptographic computations.

A technical report describing these results was already published [45] and the results are under submission to an international conference and an international journal for publication.

Table A.1: The attacker's objectives in an automotive system architecture.

NAME	DESCRIPTION	IMPACT
Obj. 1	Disclosure or interruption of the security mechanisms integrated in the vehicle's radio protocol	Unauthorized communication path towards an ECU
Obj. 2	Compromising the software implementation of the vehicle's radio interface or protocol	Unauthorized communication path towards an ECU
Obj. 3	Manipulate the execution flow of an ECU	Execution of arbitrary code
Obj. 4	Transmission of specific network packets towards the in-vehicle network	Manipulation of vehicle settings
Obj. 5	Recovery of any security information from a service user	Impersonation of an authorized entity

A.2 Security Needs in Modern Automotive Services

A.2.1 Attacker Objectives and Model

In the aforementioned communication and execution environment, we assume that an adversary is able to transmit and receive arbitrary data packets on the radio interface, armed with the sole knowledge of the radio protocol in use [60]. Under this assumption, which is perfectly reasonable and realistic, we develop the attacker model summarized in Table A.1. The attacker's objectives may be very different, and they may shift, because they are driven by the underlying economic motivations. For instance, due to the increasing capabilities of ECUs (e.g., recent infotainment system designs are based on mobile hardware architectures [7]) and their enhanced connectivity with the next generation of cellular networks, an attacker objective moves from stealing a vehicle towards using the capacity for his or her criminal ecosystem like already seen on mobile platforms [15, 21, 72].

Our attacker model encompasses these characteristics and outlines the security threats for a generic automotive-based service. This provides the background for a security assessment of our proposed security framework in §A.3. As an example, an attacker motivated to steal a vehicle, would need to accomplish **Obj. 1–5**. On the other hand, an attacker who wishes to transform the ECU—and hence the vehicle—into a member of a so-called botnet would need to execute persistently malicious code on an ECU—**Obj. 1–3**.

A.2.2 In-vehicle Network and CAN Bus

In this work we consider vehicles equipped with *Controller Area Network* (CAN) buses according to ISO11898 [38]. These CANs are highly resilient to external disturbances, and thus are suitable for high-speed distributed applications, which can exchange data (up to 1 Mbit/s) between ECUs positioned in different locations on the vehicle.

Inside the vehicle's network, each ECU connected to the CAN bus is identified with 11 (standard) or 29 (extended) bits. Each ECU can listen, transmit, and receive messages—called *data frames*—on the bus. A data frame is characterized by 8 bytes for the data plus the identifier and several bits for error detection and fault confinement. Typically, ECUs are organized in sub-networks depending on their functionalities and needed speed (e.g., braking system, engine system, or infotainment). The CAN can be viewed as a three-layers protocol: the *object layer* and *transfer layer* are responsible for the post-processing of the message (e.g., synchronization, arbitration, error detection and message filtering) and the *physical layer* handles the electrical issues on the bus.

Clearly, CANs' security requirements differs from the security requirements of the highly-interconnected scenarios typical of modern in-vehicle services. Unsurprisingly, CANs lack confidentiality, integrity, availability, authenticity, and non-repudiation mechanisms, as discussed in [70] and other works such as, for instance, [14, 33, 46, 60, 71]. As a matter of fact, CANs are a closed and proprietary system that cannot be modified to secure automotive services. Therefore, in this work, we concentrate on the security problems that arise when the CAN is connected to external devices, which are more significant—especially in today's automotive services—than the security problems that exist within the CAN itself (e.g., ECUs that send unauthenticated data).

A.2.3 Wireless Connectivity via Bluetooth

Typically, connectivity to the outside world is implemented with the help of a radio interface module connected to the in-vehicle network; more precisely, a special ECU that we call in the following "Gateway ECU" (see §2.3). This ECU acts as a gateway between the internal network and the external world. In our work, we consider the Bluetooth standard as the wireless communication protocol, but the presented concept can be applied to other communication protocols as well.

The Bluetooth protocol has a two-phase session setup: after the so-called *pairing process*, which allows the peers get to know each other and set up the network properties, the actual *communication* between the peers is enabled. During the pairing process, different security features can be applied for a secure network session depending on the Bluetooth version supported by the peers. For instance, the owner of each device must check that the information displayed on each peer (e.g., a random number) is consistent, or has to choose a (static) personal identification number (PIN), usually propagated out of band. Most of the current Bluetooth authentication schemes are driven by a human-based processing. The first Bluetooth standard also includes the possibility to agree on using no security features before starting a communication session—not a recommended setting as it

opens a broad range of potential attacks. The early Bluetooth standard suffered from further security threats due to weak cryptographic primitives, as discussed in [29, 30]. Fortunately, Bluetooth v2.1 enforces the *secure simple pairing* (SSP) protocol [55], which mitigates these security threats and takes into account the constrained resources as well as I/O capabilities of Bluetooth devices. The SSP provides confidentiality and authenticity—unidirectional or mutual—for all peers in a wireless personal-area network. Nevertheless, the SSP protocol still suffers from similar security threats such as the previous Bluetooth security mechanisms (see, e.g., [26, 27]). Unfortunately, most Bluetooth applications' security (especially in embedded scenarios) rely solely on static PINs with no way to change it.

The low security offered by mainstream Bluetooth deployments, the open accessibility of the radio interface, and the closed-world assumption of in-vehicle CANs raise new and important security concerns. As described in the main part of this deliverable, other researches have recently demonstrated the feasibility of successfully compromising automotive services through the radio interface [14, 33, 46, 60, 71]. Most of these attacks exploit software implementation flaws on the ECUs or just the disclosure of the communication protocols of the ECUs to take advantage of implementation flaws. However, no mitigation and defense approaches against these threats have been proposed so far.

A.3 A Security Layer for Automotive Services

Given the attacker model and the application scenario that we described above, it is necessary to devise an *application-level security mechanism* that is independent from the underlying wireless layer, allows secure communication between portable devices and vehicles, and mitigates the security drawbacks detailed in §A.2.

Our approach secures the communication with respect to the (generic) attacker model described in §A.2.1. In addition, our approach brings the benefit of a relaxed dependency on proprietary (untrusted) parts, because it provides a "unified" layer on top of which car-to-X applications can be developed. Our experimental evaluation described in §A.4 shows that our solution has minimal deployment impact; more importantly, it complies with the real-time requirements and constrained resources of the Gateway ECU, the implementation and distribution of the application's counterpart on mobile devices, and the I/O capabilities of the deployed vehicle.

Before explaining the security approach and the details of our implementation, we provide a brief introduction of the background on our security approach and its analysis.

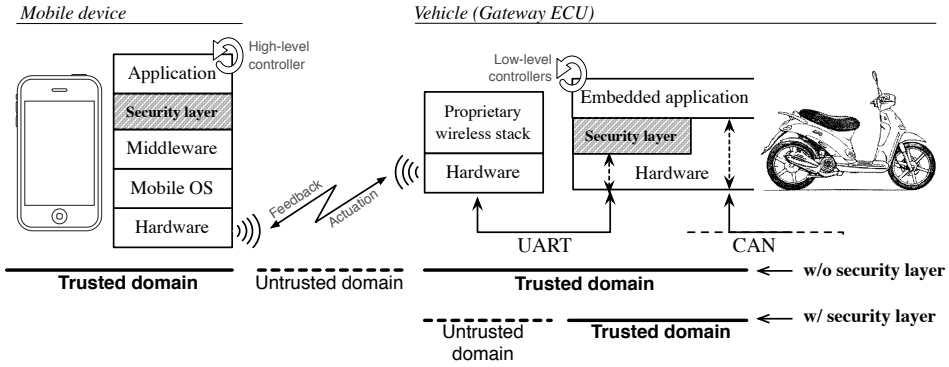


Figure A.1: Overview of the architecture proposed in §A.4 based on our approach of *trusted domains*.

A.3.1 Security Analysis

We derive the requirements of our security layer through the evaluation of the application scenario by means of trust domains and relationships between communicating parties (or entities). A party is a *trusted domain* if we trust in the correct processing and execution of the software implementation and thus in the integrity of the entity. Otherwise we consider the party as an *untrusted domain*. Depending on the characteristics and the security properties of the communication between entities, we can define *trusted relationships* (or *accepted dependence*) between entities.

For the sake of backward compatibility, we assume that the security mechanisms available are those provided by the Bluetooth standard—with the known security threats that we discussed in §A.2.3. The mobile device and the Gateway ECU are each defined as trusted domains. To mitigate the security threats by means of a potential adversary, the focus of our proposed security approach is on the vehicle side. On the mobile device side, an application serves trusted relationships to lower software layers with respect to the security mechanisms offered by the mobile operating system. We assume that the integrity of a mobile application relies on the appropriate security mechanisms (e.g., sandboxing, code signing, rights management, or secure storage) of current mobile devices and operating systems, respectively. Therefore, we focus on the integration of a given mobile system architecture and its security mechanisms without interfering the safety of actual automotive architectures. The trusted domains and relationships are summarized in Fig. A.1, which also depicts our application scenario.

A.3.2 Security Requirements

The results of our analysis are the following security requirements, which describe the background of our security framework:

Req. 1: The execution of any data is based on its context (e.g., the Bluetooth module is only dedicated to transmitting and receiving data without interrupting the execution of the ECU's application layer).

Req. 2: No dependencies on proprietary subparts of an ECU and its interfaces towards other entities.

Req. 3: Cryptographic mechanism must be under the developer's authority.

Req. 4: End-to-end confidentiality and authenticity between the application layer of a service user and an ECU.

For applying our security approach on a real application, we must take into account any security flaws of the radio interface and remove any dependency of the application layer towards proprietary parts on an ECU. In contrast to the security requirements, the current architecture introduces a dependence on the provided information flow by the Bluetooth module for the microcontroller of the ECU. Besides creating a dependence on the security mechanisms of the Bluetooth standard and software implementation of the Bluetooth stack, the processing and execution of the embedded application is explicitly influenced by this data source. According to our security analysis, both entities share a trusted relationship between each other and execute the data in a bidirectional way without any security properties.

A.3.3 Implementation Approach

We follow a two-stage approach and rely on standardized and state-of-the-art cryptography. The first stage sets up an end-to-end trusted relationship between both application layers (i.e., on the mobile device or service user, and on the ECU). Due to the constraints of the scenario (e.g., distribution of the mobile application through app stores, connectivity capabilities of the ECU), we cannot pre-compute and store any static credentials or cryptographic keys on the mobile device, nor use a public key infrastructure on the ECU. Therefore, we assume that only the vehicle's owner is able to initiate the first stage by enabling the authorization procedure on the vehicle's side, only allowing the authentication of a mobile device user for a distinct time. Within this time span, the ECU accepts the delivery of a service user's identity and the user receives the identity information of the ECU, respectively. In our implementation, the identity information includes the public keys of an asymmetric cryptographic scheme. The second stage ensures that the real-time communication requirements are met. To this end, it implements

a symmetric cryptographic scheme that establish a secure communication session, where the session key is derived from the long-term shared secret of the first stage.

For integrating our two-stage approach on the Gateway ECU's microcontroller, we implemented an ECDH key-establishment scheme [57], for the authorization of a mobile device, on a standardized curve (NIST P-192) [22]. For each authentication process, the mobile device computes a new random key set and transmit the corresponding public key to the ECU. In contrast to the key set of the mobile device, the ECU possesses a static long-term key set for the key establishment scheme (see *C(1,1)* in [57]). For the session encryption, we implemented the Advanced Encryption Standard (AES) in a chaining block cipher (CBC) mode [23, 56] with a 128-bits key. The key-derivation function is implemented according to the standard and provides a fresh 128-bits symmetric key for each communication session. We rely on the DSP-capability of the underlying hardware layer [6, 68] to compute the long-term secret and the operations on the finite field of an elliptic curve. This allows us to enhance the speed of multi-precision arithmetic operations. To further optimize the implementation, we developed most of the cryptographic primitives in assembly code. Besides these two cryptographic schemes, we implemented the SHA-1 hash function and defined a protocol structure for the integration in a communication protocol stack.

Instead of implementing cryptographic primitives on a mobile device, we opted to use a standardized cryptographic library to guarantee the proper execution and runtime behavior of the cryptographic primitives. For the integration of our security layer on a mobile device, we choose the *OpenSSL* library. Due to the constrained resources of the Gateway ECU, we enable the mobile device's random number capabilities as a source for any random number needed in the cryptographic protocols.

A.3.4 Security Analysis of the Framework

We hereby evaluate our approach with respect to the attacker model discussed in §A.2.1 and show that we can mitigate most of the identified security threats.

A.3.4.1 Unauthorized Communication:

As depicted in Fig. A.1, our security framework implements requirements **Req. 1** and **Req. 2** with the help of cryptography and takes into account a compromised communication interface. This prevents the attacker from fulfilling the objectives **Obj. 1** and **Obj. 2** (i.e., compromising the execution flow of the Bluetooth module). The focus of the attacker is to use the radio interface as an intermediate entity to transmit data to an ECU. However,

even if an attacker obtains access to the ECU via the radio interface, it is not possible to transmit any commands towards the ECU without the knowledge of a session key or the long-term secret.

A.3.4.2 Implementation Flaws or Malicious Code Injection:

The objectives **Obj. 3** and **Obj. 4** represent the deployment of malicious code on the ECU—with the final goal of issuing specific commands to the in-vehicle network. Typically, the attacker accomplishes the deployment by first interrupting the execution flow of an ECU, by exploiting for instance implementation flaws in the executed software code running in the ECU. **Req. 2** removes the dependency from proprietary implementations, and thus reduces dramatically the risk of exploitation. However, in the unlikely event that the attacker compromises the security mechanisms of the communication interface, she will also need to compromise the encryption provided by our security layer, which we can reasonably assume to be impossible. In addition, **Req. 3** ensures that any implementation flaw or security vulnerability in the deployed cryptographic primitives could also be fixed conveniently via a software update: Our security layer do not rely on any hardware device, instead the security is under the developer's authority.

A.3.4.3 Disclosure of Cryptographic Primitives:

The disclosure of cryptographic primitives is one of the most crucial attacks against the security framework. There is always the possibility of dedicated, physical attacks (e.g., side-channel attacks against the ECU's cryptographic implementations) without using tamper-proof devices. Clearly, the attacker would need to obtain physical access to the ECU (see §5.3). In addition, our security framework makes it non trivial for the attacker to obtain the cryptographic, long-term secret. In particular, a man-in-the-middle (MITM) attack is more difficult to conduct than in regular Bluetooth pairing: The attacker would need to be in the range of communication (i) during the legacy Bluetooth pairing process and (ii) during the first stage of our security framework (i.e., the exchange of the public keys). However, it is reasonable to assume that only the owner of the vehicle is able to enable the authorization process for a mobile device (e.g., in his own garage) and, more importantly, within a predefined and very brief time span.

Instead of compromising the ECU's security layer, an attacker may achieve **Obj. 5** with the help of a dedicated attack against the service user (e.g., mobile malware). First, our security framework addresses this type of security threat by fulfilling **Req. 3** and is flexible with respect to future updates to the mobile device or operating system. In fact, we are able to change any cryptographic primitive or protocol in order to protect from actual or future vulnerabilities and thus fulfill **Req. 4**. Second, we assume that the security

of the mobile application—and thus of our security layer–is based on the integrity of the operating system and its services. Note that this assumption applies on any non-trusted computing platform, and thus it is perfectly reasonable.

A.4 Experimental Evaluation

We hereby describe the experimental evaluation that we conducted using our implementation of the security layer detailed in §A.3. The goal of our evaluation is to ensure that our security protocol can be used in practice. To this end, we show that the security protocol does not impose any significant communication overhead. Furthermore, we demonstrate that it can be applied and deployed in real-world applications with minimal development efforts and with almost no impact on the existing hardware architecture that needs to be secured.

A.4.1 Case Study Overview: Electric Powered Two-Wheeler

We deployed and extensively tested our security protocol on an existing prototypical energy-management system for light electric vehicles. This system works as an intelligent range extender. More specifically, the goal of this system is to control and optimize the energy consumed by the vehicle by actively modifying the dynamical behavior of the vehicle in real-time [17, 61]. This task is accomplished with the following cascade structure (refer again to Figure A.1):

High-level controller This is the state-of-charge (SoC) controller, designed so that the battery SoC tracks a reference profile. The desired discharge policy is generated according to the a priori knowledge of the track, thus taking into account the total track distance and its elevation profile [18]. As a further degree of freedom for the user, the reference discharge profile depends on the driver's demand for energy saving. A mobile device implements the SoC controller logic by means of a mobile app, which also includes navigation features that leverage the Internet capabilities of the device.

Low-level control loops These loops prevent speed and acceleration to exceed certain limits [19]. The high-level controller updates these bounds according to control algorithms based on optimization procedures. The Gateway ECU implements and runs the low-level control loops on a 16-bits dsPIC microcontroller with a CPU speed of 20 Mips [6]. The PTW exchanges data with this device via CAN messages.

The Gateway ECU and the mobile device communicate via the Bluetooth standard and exchange the data as summarized in Table A.2. The mobile

device acts as a driver-to-vehicle interface for a service (i.e., energy management) exposed by the vehicle. This paradigm is very appealing and is gaining increasing interest among vehicle manufacturers. First because modern drivers are likely to be already familiar with mobile apps, and secondly because this deployment method facilitates the spread of software updates and the integration with other web-based services (see for instance [64]).

However, as discussed in §A.2, the use of Bluetooth for the communication of real-time data requires safety-critical issues to be addressed. In this kind of applications, sensitive data need to be exchanged between the mobile device and the Gateway ECU on the vehicle (see Table A.2 for a summary). More precisely, this sensitive data include inputs and outputs used to actively control the vehicle through the energy-management system, as opposed to mere logging or display functionality (e.g., a virtual dashboard embedded in the smartphone application). Therefore, if the data is compromised, then the functionality of the control system may be severely altered. As a consequence, the vehicle "driveability" may decrease and, depending on the attacker's skills, the driver could loose control of the vehicle.

Based on the case study, we successfully secured the existing architecture using our security framework (as described in §A.3) and conducted the experiments described in the remainder of this section with an actual PTW developed by Piaggio, a very large Italian motorbike manufacturing company, and currently in production. This PTW is used within the Green Move[1] research project, a two-year project funded by the Lombardy Region, involving eight research centers at Politecnico di Milano (Italy). This case study allows us to assess the feasibility of the security approach, characterize its computational performance, and highlight the impact on the overall control strategy.

In addition to these measurements, we also collected very positive feedback from Piaggio and from a selected pool of end users. Clearly, this is by no means intended to be a thorough usability study (which is part of our planned future work), but it provides a qualitative idea of the perceived ease of use and simplicity of our approach. We argue that this is an important aspect to ensure a good level of acceptance of automotive security solutions.

[1] http://www.greenmove.polimi.it

Table A.2: Information exchanged between the Gateway ECU and the mobile device.

FROM	TO	KIND	SIZE [bytes]	FREQUENCY
Mobile device	Gateway ECU	Initialization	48	One-shot
Gateway ECU	Mobile device	Real-time (control data)	60	5 [Hz]
Mobile device	Gateway ECU	Real-time (control data)	6	every $\bar{s}$ [m]

A.4.2 Working and Measurements Conditions

The overall system has the two fundamental working modes *pairing* and *payload exchange*, during which different types of data are exchanged. These working modes are the most critical ones in terms of computational burden and provide measurable feedback on how the security layer impacts the dynamic behavior of the control system. Thus, we concentrate our performance measurements on these two modes as we discuss in the following:

Pairing (one shot) This mode is active when the mobile device is paired with the vehicle, *after* the typical Bluetooth pairing mechanism has taken place. For this mode, we measure the performance of the asymmetric cryptography both on the mobile device and on the Gateway ECU. Moreover, as the key generation routine runs on the mobile device, we also quantify its performance.

Payload Exchange (runtime) This is when the AES key exchange, encryption and decryption take place. For this mode, we analyze the decryption on the mobile device and the encryption on the Gateway ECU. The payload consists of 64 bytes of data, which includes a padding scheme for supporting arbitrary payload size.

Note that the pairing is a one-shot task, whereas the system normally works in payload-exchange mode. At runtime, the payload exchange must satisfy real-time constraints. Here, the bottleneck lies in the Bluetooth stack as the AES encryption-decryption of the 64-bytes payload is executed each time one of the peers transmits or receives a message via Bluetooth (i.e., every 200 ms). Thus, given its importance, we collect runtime data both with a simulator and on a real implementation deployed on the PTW. This ensures an accurate performance characterization. More precisely, during the simulation, the gateway is not connected to the vehicle, but the Bluetooth connection is still active.

A.4.2.1 Impact of Interrupts:

An important aspect to consider is the non-deterministic behavior of the vehicle that may affect the performance of the system on the Gateway ECU: the interrupts on the microcontroller may interfere with the execution of the security code in a noticeable way and have a significant impact. This issue is manifested at runtime, when interrupts from the CAN bus and the UART decrease the normal sequential behavior of the executed code. Note that we can disable the interrupts on the Gateway ECU at pairing time since we need no measurements from the vehicle— the reason is that the application layer does not execute any safety crucial data during the devices' pairing.

A.4.3 Measured Performance Indicators

Table A.3 summarizes how, when and where we measured the execution time in our experiments. Both at runtime and pairing time, the execution time is a significant performance indicator. To measure the execution time on the Gateway ECU, we acquire the number of instructions N executed when the code runs and divide it by CPU speed c, thus obtaining the time elapsed in seconds, or in number of clocks (short for "clock cycles")—on the Gateway ECU. In the remainder of the paper, we explicitly plot the number of clocks N needed to execute the code so as to make the analysis independent from the actual CPU speed of the Gateway ECU.

We implemented data-logging routines on the mobile device that receive from the Gateway ECU samples of the number of clocks N. These samples are submitted within the exchanged Bluetooth messages. Although this strategy has the minor side effect of an increased payload size, it makes data collection easier compared to, for instance, collecting data directly on the Gateway ECU. In addition, our results show that this has no significant impact on the results.

A.4.4 Performance Measurements

Table A.3 summarizes the average results that we obtained from running our experiments. In the following, we explain these results in more detail and provide more analysis results regarding the performance measurements.

A.4.4.1 Pairing:

The pairing phase is characterized by the computational time needed to generate the key set on the mobile device and the execution of the ECDH protocol needed to perform the authentication scheme between the two devices.

Fig. A.2 and A.3 show the execution time measured on the Gateway ECU and on the mobile device, respectively. The small average values of the measurements both on the ECU and on the smartphone proves the feasibility of our proposed implementation in a real application. Obviously, the bottleneck of the key exchange is the Gateway ECU due to its lower CPU speed: the execution time on the microcontroller is approximately 130 ms, that is 20 time bigger than the average computational time recorded on the mobile device. In addition, notice that the results achieved on the Gateway ECU are very similar both in simulation and while driving with the vehicle; this proves that disabling the interrupts—as explained in §A.4.2—is beneficial for the execution time.

A.4.4.2 Payload exchange:

The communication overhead at runtime affects the day-to-day use of the mechanism. Hence, a significant overhead would lead to functionality issues on the control system. Fig. A.4 shows the measurements from the Gateway ECU. As expected, the simulated results differs from the on-vehicle tests: the quasi-periodical pattern shown by the real-time data while driving the electric PTW is mainly due to the periodical interrupts of the UART and the CAN bus on the microcontroller, as discussed earlier in this section. This is also clearly depicted in the statistical domain, as summarized in the boxplot shown in Fig. A.5, which clearly shows that the average values obtained during four different tests (5000 samples for each test) are remarkably constant. The maximum and minimum values are mainly due to the oscillations induced by the interrupts.

Again, the time required by the smartphone for executing the security layer code can be neglected while working on the synthesis of the control loop. The measurement results for the AES decryption of 64 bytes payloads on the mobile device are shown in Fig. A.6.

A.4.4.3 Impact of our Security Protocol on Execution Time:

Fig. A.7 shows the instantaneous Bluetooth sending frequency, which provides a concise view of the impact of the security layer on the real-time exchange of data. We derived this frequency values by first measuring the time interval ΔT between two received data frames on the smartphone. Therefore, the instantaneous frequency f_b is equal to:

$$f_b = \frac{1}{\Delta T} = \frac{1}{\Delta T_d + \Delta T_r + \Delta T_e + \Delta T_b}$$

ΔT_d and ΔT_e are the computational time of the decryption and of the encryption, respectively. ΔT_r is a random time interval between two sent messages, and ΔT_b is the time needed by the Bluetooth stack to send and receive data.

The average values of the Bluetooth frequency with and without the security layer are 4.83 Hz and 5.01 Hz, respectively. The cause of this slight discrepancy is twofold. On the one hand, the security layer introduces a delay because the terms ΔT_d and ΔT_e are significant, as shown in Table A.3. On the other hand, the size of the message sent via Bluetooth is 40% larger compared to the case where the security layer is disabled. Therefore, different payload sizes lead to different behaviors. In general, the increased size of the message decreases the Bluetooth frequency due to the low-level mechanisms implemented in the Bluetooth stack. Despite this slight decrease of sending frequency, the performance of the closed-loop system is not affected by the security routines when the high-level control strategies equipped with this additional layer are tested on the electric PTW.

A.5 Discussion and Future Work

While our reference implementation is capable of providing a security session layer that ensures end-to-end security transparently, there are three aspects that need further investigation in the future. First, in this work we concentrated on one symmetric encryption algorithm (i.e., AES/FIPS 197) and an elliptic curve key establishment protocol. Depending on the specific needs of the application domain or case study, other algorithms may be implemented and tested. For example, if a security session layer should be established for an ECU with even less computational power than the Gateway ECU, lightweight cryptographic algorithms like PRESENT [9] might be more suitable. However, embedding other algorithms in our system only requires implementation—in assembly, as we did for AES/FIPS 197 and ECDH/NIST P-192—and integration.

The second and third aspect that could be investigated further both regard the evaluation of our approach. As discussed in §A.4.1, we already collected some initial feedback from real-world users, which helped us in the

design of the user interactions. This initial feedback met the qualitative evaluation needs of this paper, whereas an extensive usability study could help improving the user-interaction aspects of our approach—although these are slightly out of scope for this paper.

Furthermore, the impact of our security layer on battery life should be measured, although we expect no remarkable results. More precisely, as we discussed in §A.4.4.3, our security layer barely affects the execution time; consequently, the computational resources of the mobile device are little affected as well. Therefore, we expect that the battery life is also not affected significantly. These conclusions are also substantiated by a series of short (e.g., 10 to 20 minutes) test drives that we performed while we monitored the battery discharge: We noticed no discrepancy when driving with and without the security layer enabled.

A.6 Conclusions

In this appendix, we highlight some of the research being done within the SysSec consortium to improve the security of the connected car. In the main part of the report, we emphasized breadth of coverage over depth. Contrary, in this appendix we have described one particular research direction in detail and it thus complements the main part of the report.

We proposed a security layer that sits on top of the Bluetooth standard (or actually, any other communication layer), ensuring a secure communication between smartphones and in-vehicle networks. This enables modern automotive services to interact with vehicles in a secure manner. Our proposed approach can be applied to real-world cases, as shown by our practical evaluation, because (1) it has very *low impact* on the (often small) computational resources available on the vehicle and the smartphone, (2) it requires *no hardware modifications* (i.e., it is agnostic with respect to the adopted wireless communication standard), and (3) it requires no complex user interactions.

We implemented our proposed system on an electric vehicle and an iPhone application that actively monitors the vehicle's battery and controls the driving speed, so that the battery lasts longer. This case study is suitable for our proposed system, because the mobile device and the vehicles exchange sensitive control data, which may affect the vehicle driveability. Our tests on this case study confirm that our system meets both the security and the real-time performance requirements.

We conclude that our approach effectively mitigates the security threats that commonly affect car-to-X applications. Furthermore, the recent attacks against cars [14, 33, 46, 60, 71] would be significantly harder if a security session layer would be used in vehicles since simple sniffing of protocol messages is not feasible anymore given our approach.

Table A.3: Summary of the average values that we obtained over 5000 samples collected by running each routine on the mobile device and on the Gateway ECU in both simulation mode (S) and while driving (D).

MODE	PHASE	DEVICE	AVERAGE VALUE		TEST
Runtime	Data encryption (64-bytes payload)	Gateway ECU	50.52	kClocks	S
			51.41	kClocks	D
		Mobile device	33.98	μs	S
			34.5	μs	D
Pairing	Key establishment protocol	Gateway ECU	2626.21	kClocks	S
			2628.67	kClocks	D
		Mobile device	7161.2	μs	D
	EC key generation	Mobile device	6939.8	μs	D

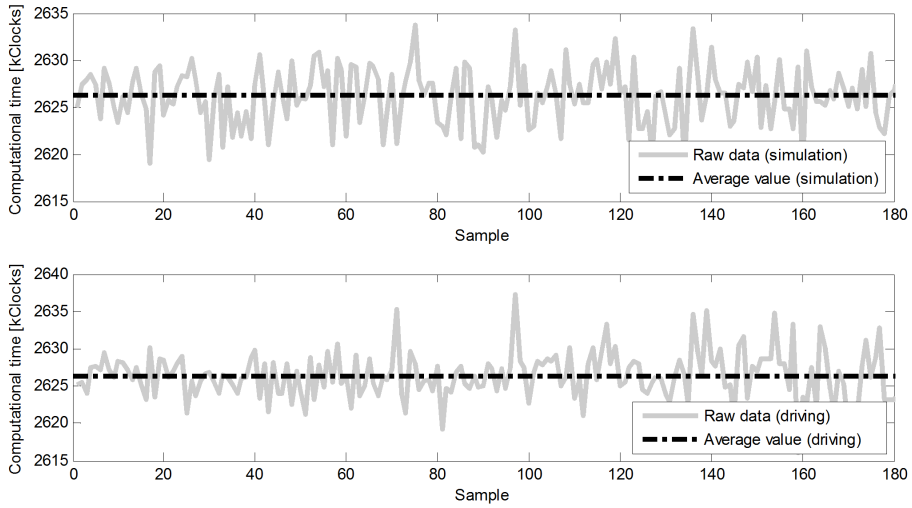
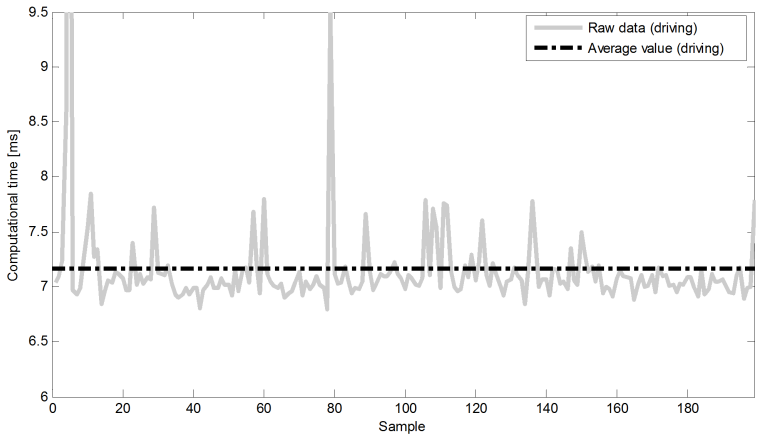
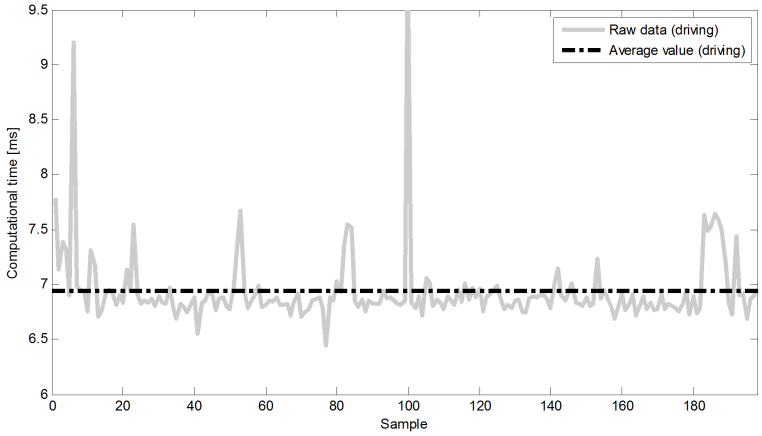


Figure A.2: Execution time for the computation of the ECDH protocol measured on the Gateway ECU. Top plot: simulation. Bottom plot: on-vehicle tests.



(a) Key establishment protocol.



(b) EC key generation.

Figure A.3: Execution time for the pairing phase measured on the mobile device. The measurements have been taken independently from each other.

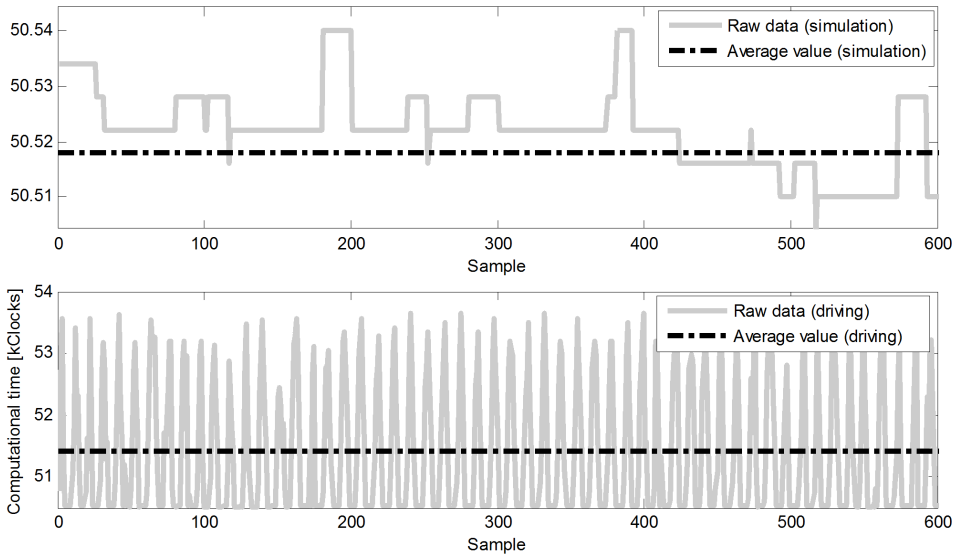


Figure A.4: Measurements acquired for the encryption of 64 bytes payloads on the Gateway ECU. Top plot: Simulation. Bottom plot: On vehicle tests.

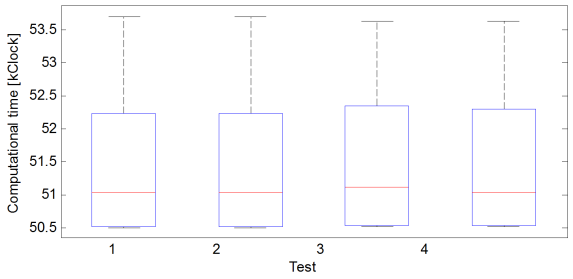


Figure A.5: Performance of the encryption of 64 bytes payloads. Boxplot of four different acquisitions on the Gateway ECU while driving the electric PTW.

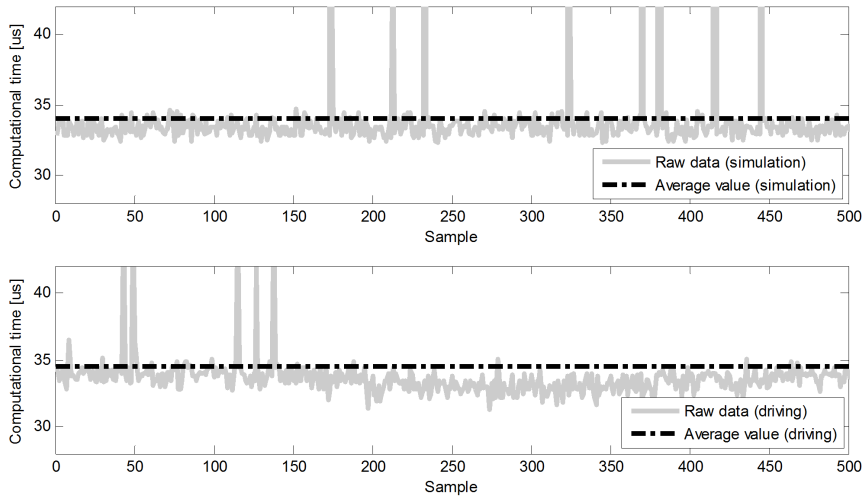


Figure A.6: Measured execution time acquired for the decryption of 64 bytes payloads on the mobile device. Top plot: Simulation. Bottom plot: On vehicle tests.

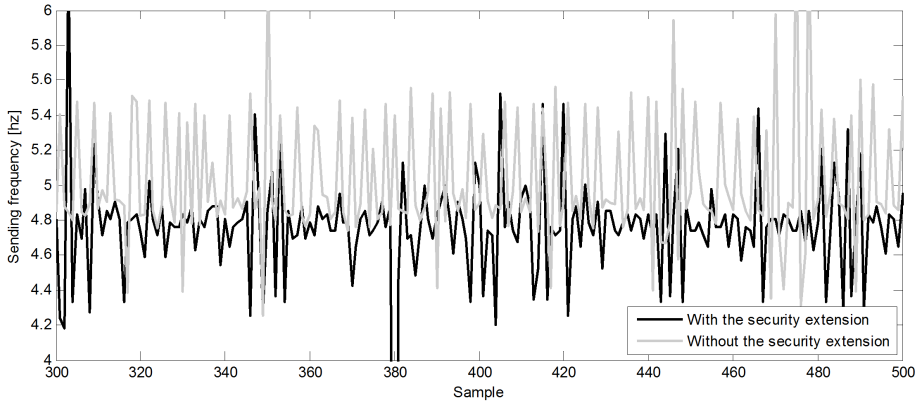


Figure A.7: Instantaneous Bluetooth sending frequency estimated with and without the security layer.

Bibliography

[1] E-safety Vehicle Intrusion Protected Applications (EVITA). URL http://www.evita-project.org/.2011-08-06.

[2] Intelligent Transport Systems (ITS); Communications Architecture. European Standard EN 302 665, v1.1.1, ETSI, 650 Route des Locioles, F-06921 Sophia Antipolis Cedex, France, September 2010.

[3] Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA). Technical Report TR 102 893, v1.1.1, ETSI, 650 Route des Locioles, F-06921 Sophia Antipolis Cedex, France, March 2010.

[4] Intelligent Transport Systems (ITS); Security; Security Services and Architecture. Technical Specification TS 102 731, v1.1.1, ETSI, 650 Route des Locioles, F-06921 Sophia Antipolis Cedex, France, September 2010.

[5] Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Methods and protocols; Part 1: Method and proforma for Threat, Risk, Vulnerability Analysis. Technical Specification TS 102 165-1, v4.2.3, ETSI, 650 Route des Locioles, F-06921 Sophia Antipolis Cedex, France, March 2011.

[6] Microchip Technology Inc. 16-bit dsPIC® Digital Signal Controllers.

[7] Audi at the CES 2012. Intelligent networking with Audi connect™. http://www.audi-mediaservices.com, 2012.

[8] Algirdas Avižienis, Jean-Claude Laprie, Brian Randell, and Carl Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33, 2004. doi: 10.1109/TDSC.2004.2.

[9] Andrey Bogdanov, Gregor Leander, Lars R. Knudsen, Christof Paar, Axel Poschmann, Matthew J.B. Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. PRESENT—An Ultra-Lightweight Block Cipher. In *International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, number 4727 in LNCS. Springer, 2007.

[10] R.R. Brooks, S. Sander, Juan Deng, and J. Taiber. Automobile Security Concerns. *Vehicular Technology Magazine, IEEE*, 4(2):52–64, June 2009. ISSN 1556-6072. doi: 10.1109/MVT.2009.932539.

[11] Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, and Antonio Lioy. Efficient and robust pseudonymous authentication in VANET. In *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, VANET '07, pages 19–28, Montréal, Québec, Canada, September 10 2007. ACM. ISBN 978-1-59593-739-1. doi: 10.1145/1287748.1287752. URL http://doi.acm.org/10.1145/1287748.1287752.

[12] *C2C-CC Manifesto*. CAR 2 CAR Communication Consortium, v1.1 edition, August 2007. URL http://www.car-to-car.org/.2011-08-06.

[13] Miguel León Chávez, Carlos Hernández Rosete, and Francisco Rodríguez Henríguez. Achieving Confidentiality Security Service for CAN. In *Proceedings of the 15th International Conference on Electronics, Communications and Computers, CONIELECOMP 2005*, pages 166–170, February 2005. doi: 10.1109/CONIEL.2005.13.

[14] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. Comprehensive Experimental Analyses of Automotive Attack Surfaces. In *Proceedings of the 20th USENIX Security Symposium*, pages 77–92, San Francisco, CA, USA, August 8–12, 2011.

[15] Dimitrios Damopoulos, Georgios Kambourakis, and Stefanos Gritzalis. iSAM: An iPhone Stealth Airborne Malware. In *Future Challenges in Security and Privacy for Academia and Industry*, volume 354 of *IFIP Advances in Information and Communication Technology*, chapter 2. Springer Boston, 2011.

[16] Kashif Dar, Mohamed Bakhouya, Jaafar Gaber, Maxime Wack, and Pascal Lorenz. Wireless Communication Technologies for ITS Applications. *IEEE Communications Magazine*, 48(5):156–162, 2010. doi: 10.1109/MCOM.2010.5458377.

[17] A. Dardanelli, M. Tanelli, B. Picasso, S. M. Savaresi, O. Di Tanna, and M. Santucci. A smartphone-in-the-loop active state-of-charge manager for electric vehicles . *IEEE Transactions on Mechatronics*, 2012. To appear.

[18] A. Dardanelli, M. Tanelli, and S. M. Savaresi. Active energy management of electric vehicles with cartographic data. In *2012 IEEE International Electric Vehicle Conference*, 2012. To appear.

[19] Andrea Dardanelli, Mara Tanelli, Bruno Picasso, Sergio M. Savaresi, Onorino di Tanna, and Mario Santucci. Speed and Acceleration Controllers for a Light Electric Two-wheeled Vehicle. In *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, Dec. 2011. doi: 10.1109/CDC.2011.6160749.

[20] F. Dressler, F. Kargl, J. Ott, O.K. Tonguz, and L. Wischhof. Research Challenges in Intervehicular Communication: Lessons of the 2010 Dagstuhl Seminar. *Communications Magazine, IEEE*, 49(5):158–164, May 2011. ISSN 0163-6804. doi: 10.1109/MCOM.2011.5762813.

[21] W. Enck, D. Octeau, P. McDaniel, and S. Chaudhuri. A Study of Android Application Security. In *Proceedings of the 20th USENIX Security Symposium*. USENIX Association, 2011.

[22] FIPS-186-3. Digital Signature Standard (DSS). NIST, 2009.

[23] FIPS-197. Advanced Encryption Standard (AES). NIST, 2001.

[24] Matthias Gerlach, Andreas Festag, Tim Leinmüller, Gabriele Goldacker, and Charles Harsch. Security Architecture for Vehicular Communication. In *4th International Workshop on Intelligent Transportation (WIT)*, Hamburg, Germany, March 2007.

[25] André Groll and Christoph Ruland. Secure and Authentic Communication on Existing In-Vehicle Networks. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1093–1097, June 3–5, 2009. doi: 10.1109/IVS.2009.5164434.

[26] Keijo Haataja and Pekka Toivanen. Two Practical Man-in-the-Middle Attacks on Bluetooth Secure Simple Pairing and Countermeasures. *Trans. Wireless. Comm.*, 9(1): 384–392, January 2010. ISSN 1536-1276. doi: 10.1109/TWC.2010.01.090935. URL http://dx.doi.org/10.1109/TWC.2010.01.090935.

[27] K.M.J. Haataja and K. Hypponen. Man-in-the-Middle Attacks on Bluetooth: A Comparative Analysis, a Novel Attack, and Countermeasures. In *Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on*, pages 1096–1102. IEEE, 2008.

[28] Sheikh Habib, Cyril Jacob, and Tomas Olovsson. An Analysis of the Robustness and Stability of the Network Stack in Symbian-based Smartphones. *Journal of Networks*, 4(10):968–975, 2009. doi: 10.4304/jnw.4.10.968-975. URL http://ojs.academypublisher.com/index.php/jnw/article/view/0410968975.

[29] C.T. Hager and S.F. MidKiff. An Analysis of Bluetooth Security Vulnerabilities. In *Proceedings of Wireless Communications and Networking Conference*, WCNC'03. IEEE, 2003.

[30] C.T. Hager and S.F. Midkiff. Demonstrating Vulnerabilities in Bluetooth Security. In *Proceedings of Global Telecommunications Conference*, volume 3 of *GLOBECOM'03*. IEEE, 2003.

[31] James Hoagland, Ollie Whitehouse, Tim Newsham, Matt Conover, and Oliver Friedrichs. Vista's Network Attack Surface. Presented at CanSecWest., April 2007. URL http://hoagland.org/presentations/CanSecWest07-Vista-Ntw-Attack-Surface.pdf.

[32] Tobias Hoppe and Jana Dittmann. Sniffing/Replay Attacks on CAN Buses: A simulated attack on the electric window lift classified using an adapted CERT taxonomy. In *Proceedings of the 2nd Workshop on Embedded Systems Security (WESS)*, Salzburg, Austria, 2007.

[33] Tobias Hoppe, Stefan Kiltz, and Jana Dittmann. Security Threats to Automotive CAN Networks – Practical Examples and Selected Short-Term Countermeasures. In *Proceedings of the 27th International Conference on Computer Safety, Reliability, and Security (SAFECOMP '08)*, pages 235–248, Newcastle upon Tyne, UK, September 22–25, 2008. Springer-Verlag. ISBN 978-3-540-87697-7. doi: http://dx.doi.org/10.1007/978-3-540-87698-4_21. URL http://dx.doi.org/10.1007/978-3-540-87698-4_21. Springer-Verlag, Berlin, Heidelberg.

[34] Tobias Hoppe, Stefan Kiltz, and Jana Dittmann. Adaptive Dynamic Reaction to Automotive IT Security Incidents Using Multimedia Car Environment. In *Proceedings of the 4th International Conference on Information Assurance and Security (ISIAS '08)*, pages 295–298, September 2008. doi: 10.1109/IAS.2008.45.

[35] Tobias Hoppe, Stefan Kiltz, and Jana Dittmann. Automotive IT-Security as a Challenge: Basic Attacks from the Black Box Perspective on the Example of Privacy

Threats. In *Proceedings of the 28th International Conference on Computer Safety, Reliability, and Security (SAFECOMP '09)*, SAFECOMP '09, pages 145–158, Hamburg, Germany, 2009. Springer-Verlag. ISBN 978-3-642-04467-0. doi: http://dx.doi.org/10.1007/978-3-642-04468-7_13. URL http://dx.doi.org/10.1007/978-3-642-04468-7_13. Springer-Verlag, Berlin, Heidelberg.

[36] Tobias Hoppe, Stefan Kiltz, and Jana Dittmann. Applying Intrusion Detection to Automotive IT — Early Insights and Remaining Challenges. *Journal of Information Assurance and Security*, 4(3):226–235, 2009. URL http://www.mirlabs.org/jias/hoppe.pdf.

[37] John D. Howard and Thomas A. Longstaff. A Common Language for Computer Security Incidents. (Sandia Report: SAND98-8667), 1998. URL http://www.cert.org/research/taxonomy_988667.pdf.

[38] International Standard Organization. Road Vehicles - Controller Area Network (CAN). ISO 11898:2003, 2003.

[39] Michael Jenkins and Syed Masud Mahmud. Security Needs for the Future Intelligent Vehicles. In *2006 SAE World Congress*, Detroit, Michigan, USA, April 3–6, 2006. SAE International. doi: 10.4271/2006-01-1426.

[40] Wolfgang John and Tomas Olovsson. Detection of malicious traffic on back-bone links via packet header analysis. *Campus-Wide Information Systems*, 25(5):342–358, 2008. doi: 10.1108/10650740810921484.

[41] Georgios Karagiannis, Onur Altintas, Eylem Ekici, Geert Heijenk, Boangoat Jarupan, Kenneth Lin, and Timothy Weil. Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions. *IEEE Communications Surveys & Tutorials*, (99):1–33, 2011. doi: 10.1109/SURV.2011.061411.00019. Early Access.

[42] Frank Kargl, Panagiotis Papadimitratos, Levente Buttyan, Müter Muter, Elmar Schoch, Björn Wiedersheim, Ta-Vinh Thong, Giorgio Calandriello, Albert Held, Antonio Kung, and Jean-Pierre Hubaux. Secure Vehicular Communication Systems: Implementation, Performance, and Research Challenges. *IEEE Communications Magazine*, 46(11):110–118, November 2008. doi: 10.1109/MCOM.2008.4689253.

[43] Pierre Kleberger, Asrin Javaheri, Tomas Olovsson, and Erland Jonsson. A Framework for Assessing the Security of the Connected Car Infrastructure. In *Proceedings of the Sixth International Conference on Systems and Networks Communications (ICSNC 2011)*, Barcelona, Spain, October 23-28 2011. IARIA.

[44] Pierre Kleberger, Tomas Olovsson, and Erland Jonsson. Security Aspects of the In-Vehicle Network in the Connected Car. In *Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 528–533, Baden-Baden, Germany, June 5-9 2011. IEEE. doi: IVS.2011.5940525.

[45] Roman Kochanek, Andrea Dardanelli, Federico Maggi, Stefano Zanero, Mara Tanelli, Sergio Savaresi, and Thorsten Holz. Secure integration of mobile devices for automotive services. Technical Report 2012-09, Politecnico di Milano, June 2012.

[46] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage. Experimental Security Analysis of a Modern Automobile. In *Proceedings of the 31st IEEE Symposium on Security and Privacy (SP)*, pages 447–462, 2010. doi: 10.1109/SP.2010.34.

[47] Andreas Lang, Jana Dittmann, Stefan Kiltz, and Tobias Hoppe. Future Perspectives: The Car and Its IP-Address — A Potential Safety and Security Risk Assessment. In *Proceedings of the 26th International Conference on Computer Safety, Reliability, and Security (SAFECOMP '07)*, SAFECOMP '07, pages 40–53, Nuremberg, Germany, September 18–21, 2007. Springer-Verlag. doi: 10.1007/978-3-540-75101-4_4.

[48] Ulf E. Larson and Dennis K. Nilsson. Securing Vehicles against Cyber Attacks. In *CSIIRW '08: Proceedings of the 4th annual workshop on Cyber security and information intelligence research*, CSIIRW '08, pages 30:1–30:3, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-098-2. doi: 10.1145/1413140.1413174. Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead.

[49] Ulf E. Larson, Dennis K. Nilsson, and Erland Jonsson. An Approach to Specification-based Attack Detection for In-Vehicle Networks. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 220–225, June 4–6, 2008. doi: 10.1109/IVS.2008.4621263.

[50] Gyesik Lee, Hisashi Oguma, Akira Yoshioka, Rie Shigetomi, Akira Otsuka, and Hideki Imai. Formally Verifiable Features in Embedded Vehicular Security Systems. In *Vehicular Networking Conference (VNC), IEEE*, pages 1–7, October 2009. doi: 10.1109/VNC.2009.5416378.

[51] Dennis K. Nilsson and Ulf E. Larson. Simulated Attacks on CAN Buses: Vehicle Virus. In *Proceedings of the 5th IASTED International Conference on Communication Systems and Networks*, AsiaCSN '08, pages 66–72. ACTA Press, 2008. ISBN 978-0-88986-758-1. URL http://portal.acm.org/citation.cfm?id=1713277.1713292. Anaheim, CA, USA.

[52] Dennis K. Nilsson and Ulf E. Larson. A Defense-in-Depth Approach to Securing the Wireless Vehicle Infrastructure. *Journal of Networks*, 4(7):552–564, September 2009. doi: 10.4304/jnw.4.7.552-564. URL http://academypublisher.com/jnw/vol04/no07/jnw0407552564.pdf.

[53] Dennis K. Nilsson, Ulf E. Larson, and Erland Jonsson. Efficient In-Vehicle Delayed Data Authentication Based on Compound Message Authentication Codes. In *Proceedings of the 68th IEEE Vehicular Technology Conference (VTC 2008-Fall)*, pages 1–5. IEEE, September 21–24, 2008. doi: 10.1109/VETECF.2008.259.

[54] Dennis K. Nilsson, Ulf E. Larson, Francesco Picasso, and Erland Jonsson. A First Simulation of Attacks in the Automotive Network Communications Protocol FlexRay. In Emilio Corchado, Rodolfo Zunino, Paolo Gastaldo, and Álvaro Herrero, editors, *Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems (CISIS'08)*, volume 53 of *Advances in Intelligent and Soft Computing*, pages 84–91. Springer Berlin / Heidelberg, 2009. URL http://dx.doi.org/10.1007/978-3-540-88181-0_11. 10.1007/978-3-540-88181-0_11.

[55] NIST Special Publication 800-121. Guide to Bluetooth Security: Recommendations of the National Institue of Standards and Technology, 2008.

[56] NIST Special Publication 800-38A. Recommendation for Block Cipher Modes of Operation - Methods and Techniques, 2001.

[57] NIST Special Publication 800-56A. Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, 2007.

[58] Hisashi Oguma, Akira Yoshioka, Makoto Nishikawa, Rie Shigetomi, Akira Otsuka, and Hideki Imai. New Attestation-Based Security Architecture for In-Vehicle Communication. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–6, New Orleans, Louisiana, November 30–December 04 2008. IEEE. doi: 10.1109/GLOCOM.2008.ECP.369.

[59] QualComm. eCall Whitepaper, version 1.5, March 2009. URL http://ec.europa.eu/information_society/activities/esafety/doc/ecall/pos_papers_impact_assessm/qualcomm.pdf.

[60] Ishtiaq Rouf, Rob Miller, Hossen Mustafa, Travis Taylor, Sangho Oh, Wenyuan Xu, Marco Gruteser, Wade Trappe, and Ivan Seskar. Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Study. In *Proceedings of the 19th USENIX conference on Security*, USENIX Security'10, pages 21–21, Berkeley, CA, USA, 2010. USENIX Association. ISBN 888-7-6666-5555-4. URL http://dl.acm.org/citation.cfm?id=1929820.1929848.

[61] S. M. Savaresi, A. Dardanelli, M. Tanelli, B. Picasso, O. Di Tanna, and M. Santucci. System and method for the active management of the driving range of a vehicle, with particular reference to electric vehicles. Italian Patent n. MI2011A000393, filed on 11/03/2011. Applicant: Piaggio & C. S.p.A. and Politecnico di Milano.

[62] Sandro Schulze, Mario Pukall, Gunter Saake, Tobias Hoppe, and Jana Dittmann. On the Need of Data Management in Automotive Systems. In *13. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme" (DBIS)*, volume 144 of *Gesellschaft fr Informatik (GI)*, Münster, Germany, March 2-6 2009.

[63] Mihail L. Sichitiu and Maria Kihl. Inter-Vehicle Communication Systems: A Survey. *IEEE Communications Surveys & Tutorials*, 10(2):88–105, 2008. doi: 10.1109/COMST.2008.4564481.

[64] C. Spelta, V. Manzoni, A. Corti, A. Goggi, and S. M. Savaresi. Smartphone-Based Vehicle-to-Driver/Environment Interaction System for Motorcycles. *IEEE Embedded Systems Letters*, 2(2):39–42, 2010.

[65] Christof Szilagyi and Philip Koopman. A Flexible Approach to Embedded Network Multicast Authentication. In *2nd Workshop on Embedded Systems Security (WESS)*, 2008.

[66] Christopher Szilagyi and Philip Koopman. Flexible Multicast Authentication for Time-Triggered Embedded Control Network Applications. In *Dependable Systems Networks. IEEE/IFIP International Conference on*, pages 165–174, June 29–July 2 2009. doi: 10.1109/DSN.2009.5270342. Dependable Systems Networks, 2009. DSN '09. IEEE/IFIP International Conference on.

[67] Vilhelm Verendel, Dennis K. Nilsson, Ulf E. Larson, and Erland Jonsson. An Approach to using Honeypots in In-Vehicle Networks. In *Proceedings of the 68th IEEE Vehicular Technology Conference (VTC)*, pages 1–5, September 21–24, 2008. doi: 10.1109/VETECF.2008.260.

[68] E. Wenger and M. Werner. Evaluating 16-bit processors for elliptic curve cryptography. *Smart Card Research and Advanced Applications*, 2011.

[69] Theodore L. Willke, Patcharinee Tientrakool, and Nicholas F. Maxemchuk. A Survey of Inter-Vehicle Communication Protocols and Their Applications. *IEEE Communications Surveys & Tutorials*, 11(2):3–20, 2009. doi: 10.1109/SURV.2009.090202.

[70] Marko Wolf, André Weimerskirch, and Christof Paar. Security in Automotive Bus Systems. In *Workshop on Embedded IT-Security in Cars*, Bochum, Germany, November 2004.

[71] Alex Wright. Hacking Cars. *Commun. ACM*, 54(11):18–19, November 2011. ISSN 0001-0782. doi: 10.1145/2018396.2018403.

[72] Cui Xiang, Fang Binxing, Yin Lihua, Liu Xiaoyi, and Zang Tianning. Andbot: Towards Advanced Mobile Botnets. In *Proceedings of the 4th USENIX Workshop on Large-scale Exploits and Emergent Threats*, LEET'11. USENIX Association, 2011.