Securing Car2X Applications with effective Hardware-Software Co-Design for Vehicular On-Board Networks

Dipl.-Ing. **Hendrik Schweppe**, EURECOM, Sophia-Antipolis, Frankreich
Dipl.-Ing. **Timo Gendrullis**, escrypt GmbH, München
M.S. **M.-Sabir Idrees**, EURECOM, Sophia-Antipolis, Frankreich
Prof. Dr. **Yves Roudier**, EURECOM, Sophia-Antipolis, Frankreich
Dr.-Ing. **Benjamin Weyl**, BMW Forschung und Technik GmbH, München
Dr.-Ing. **Marko Wolf**, escrypt GmbH, München

Kurzfassung

In diesem Paper werden die Ergebnisse des EVITA Projekts am Beispiel des *Active Brake* Szenarios vorgestellt. Konkret wurde in EVITA ein Software Security Framework in Kombination mit passenden Hardware Security Modulen (HSM) entwickelt und umgesetzt. Dieses Framework ermöglicht automobilen Applikationen Angriffsschutz und über kryptographische Absicherung der Controllerplattform und Kommunikationswege ein erhöhtes Vertrauensniveau. Solch ein hohes Vertrauensniveau wird für sicherheitskritische Anwendungen wie das vorgestellte *Active Brake* Szenario benötigt.

Abstract

In the scope of the EVITA project, we present an approach to secure the vehicular on-board communication architecture. Computing devices range from simple low-cost sensor control units on low-speed buses to costly multimedia units on high-speed buses, which allow for customized applications. Our architectural approach combines hardware- and software measures and integrates them with cryptographic protocols. These are designed for compatibility with today's vehicular bus systems like CAN or FlexRay networks, as well as upcoming systems such as IP/Ethernet. In this paper, we present the security architecture and protocols defined in EVITA, explain these on the vehicle-to-vehicle Active Brake scenario and present results of our prototype implementation and simulations.

1. Introduction and Motivation

Two years after we introduced the approach taken in the EVITA project at VDI 2009 [1], the project has now not only specified a security architecture for vehicles, but also implemented a demonstrable prototype system. This paper presents one of the major scenarios ad-

dressed, the *Active Brake*, that involves interactions between two vehicles and their security infrastructure.

Our approach provides the link between hardware security anchors (integrated security modules) and security software, which is necessary to achieve an enhanced trust level for safety-critical applications like influencing the vehicle's behavior. To enable this scenario, we integrate our in-vehicle approach with existing Car2X technology that is currently being deployed in the simTD testbed [2]. While protocols and mechanisms that have so far been proposed take primarily the wireless channel into account, we show how end-to-end security is established by including in-vehicle components into the security concept. Additionally, cryptographic performance for signature generation and verification has always been a bottleneck for Car2X communication. We overcome this problem with hardware-accelerated cryptographic functions of our HSM and show that performance is adequate even in dense traffic.

2. Related Work

The IT security of automotive systems has been a long overlooked property. While vehicle systems undergo massive functional testing and safety checks, IT security has only been applied to limited domains, such as immobilizers in car keys.

Very recently, the area has gained additional interest and research showed that current implementations pose a major threat for passenger safety [3]. The survey [4] shows an overview about the, currently rather limited but growing, research activities in this field. We believe that the introduction of cooperative Car2X systems, security must inevitably be incorporated into vehicle systems.

3. Scenario Description

The Active Brake scenario involves a number of ECUs in two vehicles: a sending and a receiving vehicle. The sending vehicle provides information about the brake status up to the information, whether the vehicle performs an emergency brake and immediately notifies surrounding cars (i.e., the receiving vehicle in our case). We use this scenario in order to demonstrate the security protocols developed in EVITA in order to facilitate trust between entities.

The purpose of the Active Brake scenario is to provide the driver recommendations based on brake information received from a vehicle driving in front of the driver's vehicle. Depending on the relative position of the vehicles, their speed and current acceleration, a corresponding brake recommendation is deduced and provided as information to the driver. Depending on the security level, an automated brake is technically feasible.

The demonstrator setup (cf. Figure 1) conveys a sending vehicle (SV) and a receiving vehicle (RV). The communication between SV and RV is done via simTD communication hardware [2] based on 802.11p and is represented as communication path AB_3.

The sending vehicle uses the simTD communication control unit (R-CU) running the 802.11p based communication stack as well as the security daemon (SD) securing the decentralized environmental notification message DENM. Here, the security daemon uses the EVITA security stack and the EVITA full HSM in order the sign the DENM based on EVITA security mechanisms. The DENM generation is triggered via the brake sensor S-BS, which provides information about the deceleration and with which force the brakes have been engaged. This "active brake" information is provided via communication path SV_AB_1 secured with EVITA to the EVITA brake ECU S-BC, where the information is processed and common safety algorithms[1] decide, whether DENM creation is triggered or not. If it is decided to trigger the DENM creation, the trigger information is sent via communication path SV_AB_2, secured with EVITA, to S-CU, where the actual DEN message is generated, secured, and then sent out via AB_3 to the receiving vehicle.

Vehicle	ECU description	Name
SV	Brake Sensor	S-BS
SV	Brake ECU	S-BC
SV	Communication Unit	S-CU
RV	Communication Unit	R-CU
RV	Application ECU	R-AU
RV	Instrument Cluster (display)	R-IC
RV	Brake ECU	R-BC
RV	Brake Actuator	R-BA

Table 1: ECUs in Active Brake Scenario.

The other vehicle receives the DENM via AB_3 and uses its security daemon to verify the security of the incoming message particularly regarding authenticity and integrity. The security daemon uses the EVITA security stack in order verify the signature of the DENM. After successful verification of the DENM, the "active brake" information is provided via communication path RV_AB_2 to R-BC. The link between RV_AB_2 is secured via the EVITA security protocols and corresponding HSM modules. When the security (i.e., authenticity and integrity) of the information provided via RV_AB_2 has been successfully verified, the brake ECU application further processes the data. If the application decides (again by using common safety algorithms) to provide the driver a recommendation, the decision is securely provided via R-AB_1 to R-IC, where a recommendation is displayed on the HMI to the driver.

[1] These represent reuse of standard safety decision algorithms not developed in this project.

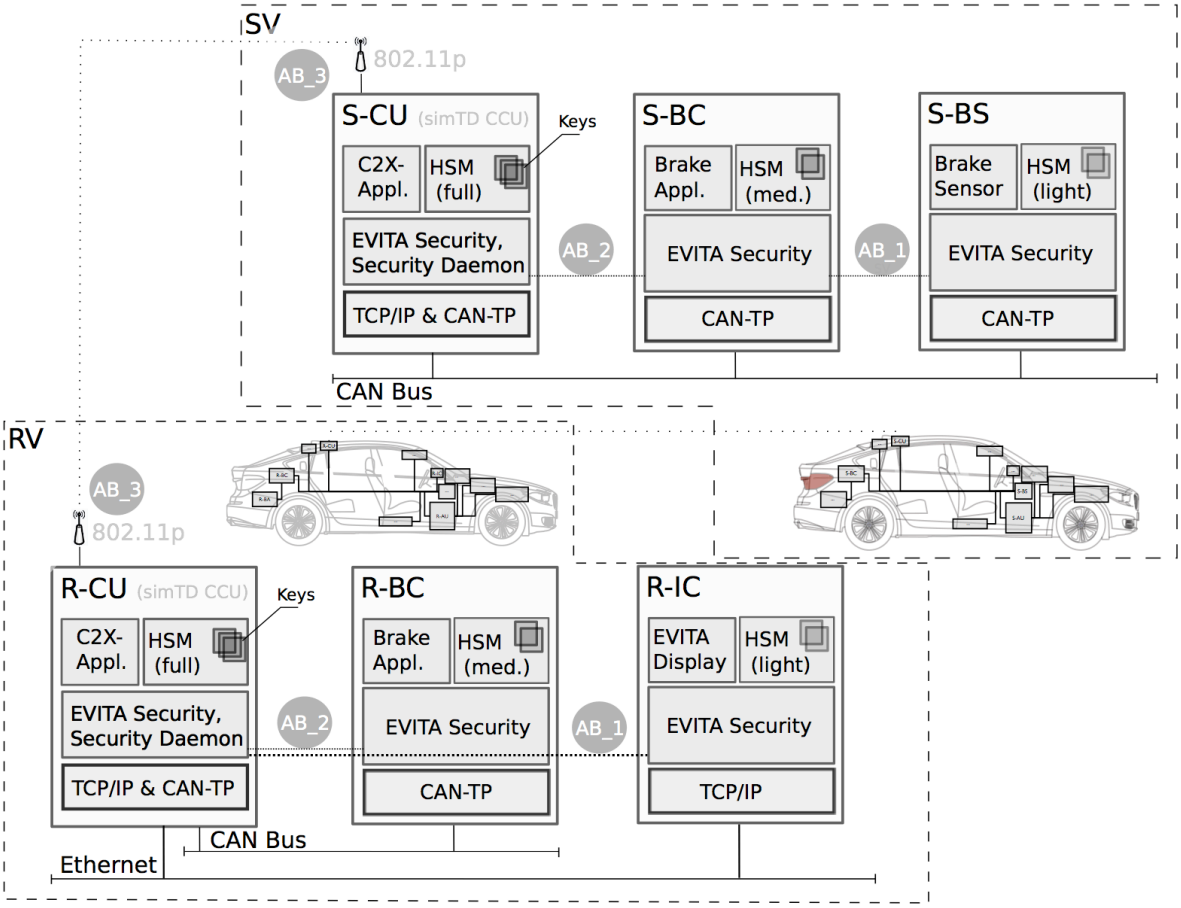

Figure 1: Vehicle Demonstrator Setup with sending (upper) and receiving vehicle (lower). AU for Key Distribution and optional Brake Actuator are omitted and discussed in the following.

In Figure 2 an abstract functional sequence chart is depicted, showing the Active Brake scenario and involved components. In addition to recommendations provided as information to the driver (e.g. via R-IC), more active control may be foreseen, e.g. via actuator R-BA. It is intuitively clear that functionality involving the semi-automated or even automated control of vehicles–especially in emergency situations–must be secured adequately. While automated emergency braking based on purely external information can barely be imagined today, tomorrow's systems will provide the intelligence and capabilities to handle such delicate situations. The appropriate protocols to ensure authenticity and integrity of data exchanged between ECUs in the EVITA on-board networks provides for these security requirements.

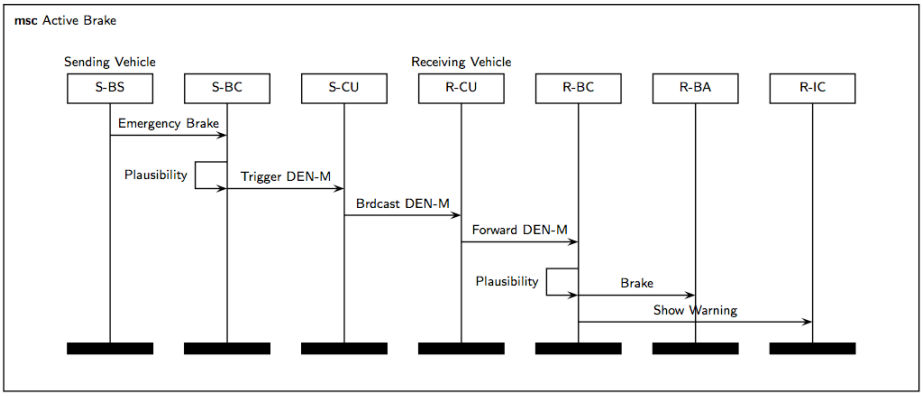

Figure 2: Sequence Chart: Functional view of Active Brake scenario with involved components. Each step (with the exception of S-CU to R-CU) is secured by EVITA protocols.

4. Hardware Security Module

The central base of the EVITA vehicular security solution is a Hardware Security Module (HSM) specially designed for application in automotive security use cases. This HSM serves in particular for:

- secure storage of elementary security assets (e.g., cryptographic keys),
- secure processing of security assets (e.g., data encryption),
- a-priori trusted security anchor for all upper (software) security realizations (e.g., secure boot).

The HSM provides a small, controlled, non-volatile storage and processing environment executed *before* and *isolated from* the ECU's main processor and its main software applications. Thus, the HSM can protect security-critical assets and security-critical operations against attacks and vulnerabilities of the ECU's software. Software on the ECU is–in contrast to the rather small HSM implementation–very difficult to implement without security leaks due to its complexity and the sheer amount of code. Optionally, the HSM can be equipped with a range of hardware tamper-protection measures.

In order to enable a holistic but cost-efficient in-vehicle security architecture, the EVITA approach provides at least three different HSM variants – *full*, *medium*, and *light* – each focusing on a different application use case with different cost, functional and security requirements. The light and medium module, however, are fully compatible subsets of the full module. The EVITA HSM concretely provides hardware-protected security functionality for:

- asymmetric encryption and decryption,
- cryptographic hashing,
- random number generation,
- monotonic counters,
- symmetric encryption and decryption,
- key management ,
- secure real-time clock,
- bootstrap measuring,

together with a small internal processor and few kilobytes of volatile and non-volatile memories for hardware-protected security processing and hardware-protected storage of security assets such as cryptographic keys, certificates, and bootstrap references.

Key-Use-Flags

A distinctive feature of the EVITA HSM is the possibility for very fine-grained authorizations for the processing and the migration of protected security assets. In particular, a single security asset can have several individual authorizations that allow or forbid processing it in different HSM security functionalities. This is specified by so-called *use flags*. Thus, a symmetric cryptographic key, for instance, can have a use flag for Message Authentication Code (MAC) verifications but no use flag for the creation of MACs. Moreover, these use flags can have individual migration authorizations that specify their transport restrictions to locations outside the respective HSM as used by key exchange protocols. Thus, the use flag for signature verification of a certain key, for instance, can be allowed to become migrated to another HSM, while the use flag for *signature creation* of the same key cannot be moved to a location outside its local HSM. Lastly, each use flag can also have individual authorizations required for each invocation of HSM functions. These invocation authorizations can be simple passwords, but can also be based on the individual ECU platform configurations as measured during the ECU's boot process, or can even be a combination of both (i.e., the platform's state and password).

5. Runtime Environment and Software Security

The low-level hardware security functionalities of the hardware security module described in Section 4 provide the base or "security anchor" on which various high-level software security mechanisms can be built on. Within our approach we reused and extended the already existing software security framework named EMVY and coupled it tightly with the HSM. EMVY is an in-vehicle software security framework that allows easy integration and easy application of vehicular security mechanisms. Amongst others, EMVY provides security mechanisms for:

- flexible on-/off-board user/identity authentication and authorization (access control)
- communication protection (e.g., regarding authenticity, confidentiality) and filtering,
- secure storage using access authorizations for entities and platform configurations,
- intrusion detection, malware protection, and selective intrusion response,
- central, flexible security policy integration point and distributed policy enforcement,
- secure integration of backend services, third party applications, and user CE devices,
- bootstrap authentication, attestation, and bootstrap integrity enforcement, and
- cryptographic algorithms and security protocols.

In our prototype, EMVY uses a distributed master-client architecture with inherent communication protection that enables a vehicle-wide deployment and collaboration of all security-relevant ECUs within a vehicle. All clients request services such as key distribution, policy decision or remote entity authentication from the master in a “thin client” fashion, i.e., the service only needs to be implemented on the master. However, depending on the vehicular architecture, other deployment scenarios are possible, e.g. applying multiple security masters.

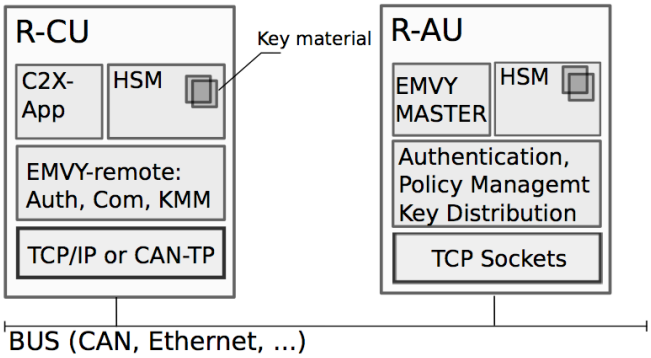

Figure 3: Distributed Architecture of EMVY Remote Client and EMVY Master. Policy decisions are requested by clients and taken by the master. Authenticity and trusted state of the ECUs are guaranteed by an HSM, which is used to sign data exchanged between EMVY nodes.

6. Protocols

The protocols used within the vehicle are integrated with the hardware security anchor (section 4) and software security architecture (section 5). We exemplarily show how session keys for communication are generated and distributed (“KMM” for Key Management Module) and a policy system (“PDM” for Policy Decision Module, detailed in section 7), which is queried for access control enforcement.

External Communication

While internal communication between ECUs is EVITA specific, the external communication link between vehicles is protected by protocols defined within the simTD project. The EVITA HSM accelerates cryptographic operations (e.g., RSA signatures and certificate handling for IEEE1609.2 signatures as used in simTD), where possible.

Internal Communication

Communication sessions for periodically sent data, such as “brake pedal status” values, are established at boot time. Communication groups are defined inside the vehicle’s policy management system. The group session itself is established by the sending ECU. Thanks to the HSM use-flags introduced above, session keys (even symmetric keys) may be used asymmetrically. This is enforced by export limitations of use-flags inside the HSM, i.e., exported

keys k_e may not be used for the same operations as the original key k_o. An example is the restriction to only verify a MAC with the exported key, despite the fact that the same cryptographic material is used inside the HSM.

After a session key is generated by an ECU, a request is issued to the network's master node. This request is handled by a number of EMVY modules (see section 5): authentication, policy decision and key management. If authentication and policy decision are positive, the exported key material is submitted to the master's KMM, which is located on the AU in our example scenario. The HSM export of the session key uses a pre-shared transport and authentication key. The transport encryption happens inside the HSM and comprises an authenticity code (using a separate key), timestamp, and validity interval. We assume session keys to be valid for one drive-cycle or a maximum of 24 hours. We chose this time to accommodate for MAC truncation limitations that limit the overhead on communication buses but also the cryptographic robustness against brute force attacks. Further details on the key distribution protocol can be found in the EVITA deliverable documents [5], e.g., D3.3 and [6].

The CAN buses used in today's vehicles only provide little space (8 bytes) for payload of individual data packets. Thus, we use the ISO transport protocol (ISO 15765-2) in order to exchange larger payloads such as key material, signed payload, or complex queries. We implemented a gateway between the CAN bus and TCP/IP connections, which does not imply payload restrictions despite the 4095 bytes allowed for ISO-TP fragmentation. We implemented this gateway by using the `SocketCAN` API available in the latest Linux kernels.

Ethernet/CAN Gateway

The gateway can be used in a listening mode, in which it opens an IP socket on the gateway's Ethernet/IP interface, or it can be used in an actively connecting mode, opening a connection to a remote host on request. Payload data from these sockets are segmented into ISO-TP frames and forwarded to the CAN bus using the associated identifier address pair. Incoming data from this address pair is forwarded to the IP sockets accordingly. We use TCP connections for unicast connections, e.g., between EMVY remote clients and the EMVY master, and UDP for group communication between EMVY clients. Each ECU is equipped with the EMVY client library, which is used for programming secured applications.

The CAN-TP gateway is necessary as we have two ECUs (R-BC and R-BS) in our setup, which are only connected via CAN. An adapted EMVY remote library is used in combination with the AUTOSAR runtime environment on an Infineon TriCore microcontroller.

7. Security Policies

The design and enforcement of security policies, and in particular the access control policy, is central in connecting applications with the underlying middleware and security mechanisms and protocols. We propose a modular architecture for managing and enforcing security policies that can be deployed in different ways, and in particular distributed if need be, thus allowing policies in different areas of the vehicle (like safety-critical subsystems and entertainment subsystems) to be controlled separately. The rest of this section focuses on access control policies, which specify to which extent and under which conditions an ECU (physical or virtual) is allowed to access and use a specific resource. They mainly consist in two classes: communications access control policies, and application access control policies depending if the resource is defined in the EMVY framework or in an application specific manner.

Communications Access Control Policies

Defining which entities or groups of entities are allowed to communicate together is a basic requirement in the EVITA architecture. Figure 4 depicts for instance how communication between ECUs is set up in the active brake scenario.

```
<Group name="ActiveBrake-Receivers">
        <type>1:n</type>
        <initiatorEntity>R-CU</initiatorEntity>
        <members> R-CU, R-IC, R-CB </members>
</Group>
<Group name="Brake Activator">
        <type>1:1</type>
        <initiatorEntity>R-BC </initiatorEntity>
        <members> R-BC, R-BA </members>
</Group>
```

Figure 4: Group Communication Defined through the Communications Access Control Policy

In particular, this policy determines which ECUs the KMM component will distribute keys to in the vehicle. In this respect, communications access control policies management is integrated into the EMVY framework, which it extends with the necessary configuration functions. This policy also makes it possible to define message filtering that may be performed at a gateway. Such a filtering might for instance prevent some traffic from entering a bus and abusively consuming bandwidth.

Application Access Control Policies

Authorizations in on-board applications use authenticated attributes of the subjects and objects, which we define by building upon the XACML language with some extensions. In that manner, application programmers have access to a well-known language, which they can extend for expressing their authorizations and diverse application specific resources, and which supports fine grained and conditional authorizations, policy combination, and conflict resolution. Policies expressed in XACML and containing different "*PolicySets"* may however

become too long to interpret and the policy expressed very complex. Due to the limited resources of ECUs in vehicular embedded systems, we thus implemented a translation from XML encoding into a binary ASN.1 based format easier to interpret by the policy decision module (see below).

Policy Decision and Enforcement Architecture

Security policies are managed by dedicated components akin to the XACML policy decision and enforcement points.

The policy decision module (PDM) determines based on a local policy database whether access to a particular resource is granted. The main PDM is typically deployed in a central ECU, together with the KMM. Yet, a PDM may additionally delegate decisions to other PDMs, typically as defined by application programmers on specific ECUs.

Policy enforcement points (PEP) are integrated in various applications in ECUs, as well as in the EMVY framework for generic policies. A PEP effectively controls the application of the security policy to a given resource. For instance, the KMM constitutes a hardcoded PEP for the enforcement of the communications access control policy.

8. Quantitative Evaluation

Within the scope of the EVITA project, the security hardware architecture was transferred into a proof of concept implementation and integrated with the implementations of the software security framework and protocols.

For prototyping purposes the HSM full version was implemented in a hardware and software co-design approach on an off-the-shelf available FPGA platform (i.e., *Xilinx Virtex-5FX70T* on an *ML507* evaluation board). Exemplary instances of cryptographic algorithms – AES-128, ECC-256, and WHIRLPOOL – covering the HSM's security functionality for symmetric and asymmetric encryption and decryption, cryptographic hashing, and cryptographic signatures were implemented in hardware in the configurable logic of the FPGA. The embedded dedicated microprocessor (i.e., a PowerPC) of the FPGA hosts the HSM's internal processor with access to the configurable logic – and thus to the cryptographic algorithms in hardware. High-level cryptographic protocols based on those available algorithms as well as the communication and data handling are implemented in software on the internal processor. Furthermore, the FPGA evaluation board provides all necessary resources such as volatile and non-volatile memory, external interfaces, and debugging facilities to allow an easy integration into the overall EVITA design.

Although the HSM is physically connected via external interfaces and busses to the platforms hosting the ECUs (i.e., via SPI) or AUs (i.e., via Ethernet), the interconnection is still assumed to be secure against manipulations and eavesdropping. In later stages of the implementation process (e.g., for field operational tests with an ASIC design of the EVITA architecture, which was out of scope within this project), the HSM should be integrated into the same chip as the microcontroller of the ECU.

The prototypical HSM implementation was extensively tested and evaluated regarding its performance after deploying it on the target platform. In real world performance measurements on the internal HSM processor the AES-128 provides a data throughput of up to 80 Mbit/s in ECB encryption mode, of up to 60 Mbit/s in CMAC generation mode and the cryptographic hashing function WHIRLPOOL of up to 128 Mbit/s. Generating and verifying ECDSA signatures based on ECC-256 both exceed 400 signatures/s. The bottleneck of the prototypical HSM implementation is–compared to the values above–the relatively slow SPI connection in case of the ECU integration with a data throughput of only 22.5 Mbit/s.

We will have real-world performance measures including latency measurements of the radio communication using simTD's CCUs for the described Active Brake use case when the final demonstration integrations are performed in November 2011. In line with estimations from the Car2Car Communication Consortium, the performance of our HSM will be able to secure communication even for situations with very dense traffic.

9. Conclusions and Outlook

The EVITA project has developed solutions for securing automotive on-board networks and achieved following key results:

- EVITA enables authentic communication within well-established automotive on-board communication environments by scaling the security against the constraints, e.g. of a CAN-bus.
- EVITA is capable of extending the security level by deploying hardware security modules, which are specifically fit to the communication paradigm currently established in automotive networks. It provides secure group communication and keying based on an HSM infrastructure within the vehicle.
- In order to provide a cost-effective while secure solution, EVITA proposes three levels of HSMs with different security level as well as appropriate security protocols. These protocols use the available functionality of a deployment, which comprises at least one EVITA full HSM and a set of EVITA medium HSMs and EVITA light HSMs.

- EVITA lays the foundation for future Car2X based communication scenarios with very heterogeneous security requirements, as the EVITA architecture is designed to be scaled according to the specific needs of the actual on-board network. It provides appropriate security measures and hardware acceleration, e.g., ECC, for fast processing of signatures.

The security approach provided by the EVITA project foresees a generic and system-wide approach. The scalable architecture enables the application to use security measures according the requirements and constraints of dedicated use cases and given on-board networks. It is important to provide an approach, which protects against certain risks and which provides a certain security level. As with all security, it has to be clear, that though a security level with a certain protection level can be established at this point in time, there won't be a guarantee that this security level cannot be broken any time in the future.

We see the integration with the simTD project as a promising first step to show the real-world feasibility of our approach. In order to drive the research efforts of EVITA closer to an actual product, the FP7 project PRESERVE builds on our results and develops an ASIC design based on the EVITA HSM approach.

References

[1] O. Henniger et al.: *Securing Vehicular On-Board IT Systems: The EVITA Project*, VDI Automotive Security, Ingolstadt 2009.

[2] H. Stübing et al: *simTD: a car-to-X system architecture for field operational tests,* IEEE Communications Magazine, Volume 58, Issue 5, May 2010.

[3] K. Koscher et al.: *Experimental security analysis of a modern automobile*, 31st IEEE Symposium on Security and Privacy, Oakland 2010.

[4] P. Kleberger et al.: *Security Aspects of the In-Vehicle Network in the Connected Car*, IEEE Intelligent Vehicle Symposium, Baden-Baden 2011.

[5] The EVITA Project. *Homepage, Publications, and Deliverable Documents*, online, http://www.evita-project.org/publications.html, 2008-2011.

[6] H. Schweppe, et al.: *Car2X communication: securing the last meter*, 4th IEEE Symposium on Wireless Vehicular Communications WIVEC, San Francisco 2011.

[7] The PRESERVE Project. *Homepage, Publications, and Deliverable Documents,* online, http://www.preserve-project.eu/, 2011-2014.d