Car2X Communication: Securing the Last Meter

A cost-effective approach for ensuring trust in Car2X applications using in-vehicle symmetric cryptography

Hendrik Schweppe*, Yves Roudier*, Benjamin Weyl†, Ludovic Apvrille‡, and Dirk Scheuermann§

*EURECOM
Sophia-Antipolis, France
{schweppe, roudier}@eurecom.fr

†BMW Group Research & Technology
München, Germany
benjamin.weyl@bmw.de

‡Telecom ParisTech - LTCI CNRS
Sophia-Antipolis, France
ludovic.apvrille@telecom-paristech.fr

§ Fraunhofer SIT
Darmstadt, Germany
scheuermann@sit.fhg.de

***Abstract*—The effectiveness of Car2X communication strongly relies on trust in received data. Securing in-vehicle communication is an essential yet so far overlooked step to achieve this objective. We present an approach based on the use of symmetric key material protected with inexpensive hardware. We modeled the involved cryptographic and network protocols, showed their applicability to automotive bus systems and conclude about their soundness with analytical and simulation methods. A prototype realization in real vehicles is envisaged as part of an ongoing project.**

On-Board Protocols; Automotive; Security; Bus Systems; Embedded Communication; Protocols

I. Introduction

Car2X communication has been a major research topic for safety and traffic efficiency scenarios within recent years, and security and privacy are critical to those scenarios. Most solutions have been focusing on the communication between nodes, in particular vehicles. However, as pointed out by [1], the need for in-node security and privacy is only beginning to be covered for Car2X based applications. Recent research we conducted in the European-funded project EVITA has been specifically addressing those requirements and appropriate solutions for vehicular on-board networks.

This paper introduces a new communication security architecture for Car2X actors. We notably describe a key distribution protocol, which serves as a basis for integrity and authenticity within automotive on-board networks, and a secure transport protocol for in-vehicle communication. Our approach has been tailored for on-board security, and can be tuned to reach a certain security level depending on the application requirements and on the vehicles' on-board topology. It aims at securing one-to-one as well as one-to-many communication between ECUs, and ensures the trustworthiness of information and the in-vehicle data aggregation process before such data are being sent to other vehicles.

The vehicular on-board communication architecture has been designed towards functional requirements, such as error-recovery, low latency, and high tolerance to electromagnetic interference. The security of bus systems was, to a great extent, achieved through obscurity, i.e., by physically concealing wires and by preventing the dissemination of insider knowledge about data formats and computer interfaces. This means that security either has to be added "on top" of existing solutions or new communication systems and protocols have to be designed and deployed. The latter is virtually impossible for the automotive industry due to cost constraints.

There classically exist several approaches to establishing secure channels. They either rely on shared secrets or on asymmetric cryptography and a public key infrastructure. The latter can hardly be implemented inside vehicles, while the former is too weak for trust establishment in many Car2X scenarios, despite the fact that it makes maintenance difficult. Our security solution instead ensures an open yet symmetric cryptography based trust establishment between ECUs. It integrates into the existing infrastructure of vehicles, our only assumption being the introduction of security hardware modules, which are anyhow needed for cryptographic acceleration reasons.

The most commonly used communication bus in today's vehicles is the CAN bus that is operated at around 500 kbit/s and offers 8 bytes of data payload per packet. From a security perspective, a fraction of eight bytes for the security payload does not provide an adequate protection. On the other hand, including a larger payload field or a dedicated field for security is not feasible. We show how higher level protocols, which have been used for other purposes in the vehicle, can be employed in order to solve this particular issue.

II. Deployment Architecture

A. Asymmetric Usage of Symmetric Keys

We make use of shared secrets (i.e., symmetric keys) due to cost and embedded constraints. Deploying those keys requires a small hardware component, the Hardware Security Module (HSM) at each ECU. The HSM implements accelerated cryptographic primitives and also offers secure (i.e., tamper proof) key storage, thereby preventing any ECU compromise to spread to our key infrastructure. The HSM also mediates the access to cryptographic keys. The latter carry metadata, such as an *expiration date*, an *authentication code* or so-called *use-flags*. These use-flags enable the HSM to distinguish for which functionality a specific key can or cannot be used. Typically, the HSM will enforce some usage control on the symmetric keys it stores, based on those use-flags: for instance, a symmetric key may be tagged for *sign*ing (i.e., generating a MAC) at a single HSM while the exact same key resides at several other HSMs, although tagged for *verify*ing only.

This usage control makes it possible to enforce an asymmetric use of key material, while the computation itself is efficiently based on symmetric cryptography. In the following, we speak of keys $k_{n,g}$ as sending (generating) key for the session key for ECU_n and $k_{l,v}$ for the receiving (verifying) ECU_l.

Two other use-flags used throughout this paper are the *export* and *transport* flags. The former allows a key to be exported in encrypted form and the latter allows it to be used to encrypt another key for export (that other key must be exportable). A cryptographic key can under no circumstances leave the HSM unencrypted.

The EVITA project differentiates between three different types of HSMs. While full and medium HSMs support and accelerate asymmetric cryptography, we focus on the use of the light HSM, which provides a hardware based implementation of symmetric cryptography primitives only. This type of module is most interesting for integration into low-cost sensors and actuators. A comprehensive and detailed specification, including a comparison with the HIS SHE module, can be found in [2].

B. Dynamic Key Exchanges

Globally shared keys that would, if revealed, definitely compromise the overall system security are avoided in our solution. The distribution of keys between communication partners is done through an entity called the Key Master (KM). Every ECU e_i on the vehicle network shares two secret keys with the KM. The first key $k_{i,a}$ is used to *authenticate* the entities against each other, while the second key $k_{i,t}$ is used for *transport encryption* of generated keys. A KM is a logical entity, which can be present on one or multiple ECUs. It is possible to have multiple KMs (e.g., one for every domain), so that keys are only shared for a limited number of hosts. For reasons of brevity, we discuss a simplified deployment with only one central KM present in the vehicle network (see Fig. 1). For multiple KMs, an additional connection between the involved key masters must be established within the key distribution protocol. We use a distributed policy system for authorization of such domain-spanning connections [3].

C. Multi-Criteria Setting of Secure Communication

Our work is integrated into a security framework, which notably provides API abstractions for (i) entity authentication, (ii) secure communication (integrity, confidentiality), and (iii) access control along with policy management. The application programmer can open a communication channel from A to B with a given *security-property-set* SP and *security-level* SL. The receiver B may be a group of n recipient nodes, thus creating a $1 : n$ communication channel. The security property set contains channel attributes such as `authentic` or `confidential`, while the security-level refers to the selection of a particular MAC strength, as explained in Section III-C. The communication module initiates the key exchange and establishes a secure channel according to those attributes. Direct and group communication are managed in the access control framework, which interacts with the communication API as well as with the key distribution protocol.

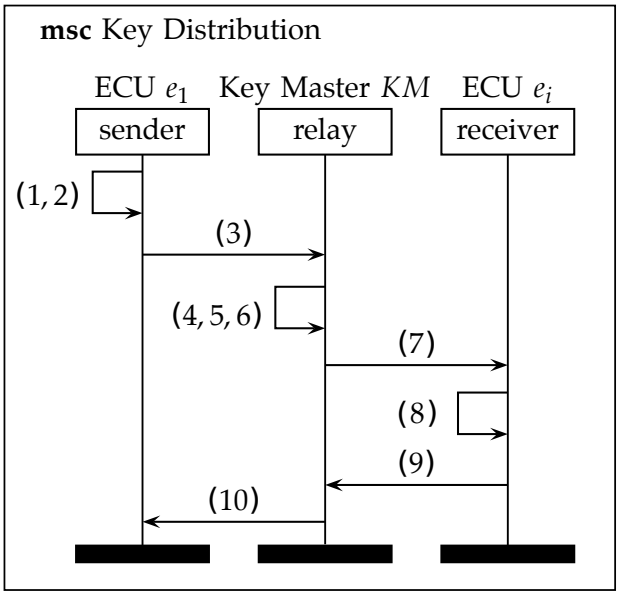

Fig. 2. Key Distribution via Key Master. Steps (6—9) are executed for all i ECUs in the receiver group. The numbers correspond to the steps defined in III-A.

III. Protocols

We present a protocol for key distribution and another one to secure intra-vehicular communication using the exchanged session keys. In order to explain the protocols, a simplified scenario of multicast communication is assumed: one node sends messages to a group of receivers, as it is commonly done in today's vehicles[1].

A. Key Distribution and Entity Authentication

In the following we show how keys for secure communication are exchanged via the key master node. The example also applies to session-keys for encrypting communication, although authentication code generation and verification flags are used subsequently.

A secure communication session s between an ECU e_1 and a group g_x of i ECUs e_i is established by the following steps. All cryptographic operations such as generation or import/export take place inside the respective HSMs.

1) *At* e_1: Local generation of a key pair $k_{s,g}$ and $k_{s,v}$, where only $k_{s,v}$ is exportable. The key has a limited time of validity.
2) Exporting $k_{s,v}$, encrypted with transport key $k_{1,t}$ and authenticated with $k_{1,a}$ We call this the key blob b_1.
3) Opening an unauthenticated communication channel between e_1 and KM.
4) *At* KM: Authorizing communication of e_1 to group g_x based on policies.
5) Importing authenticated key blob $b_1 (= k_{s,v})$ into HSM of KM.

[1]On the CAN bus data are broadcasted on the medium and received according to a device's acceptance filter bit-mask, similar to IP multicast.

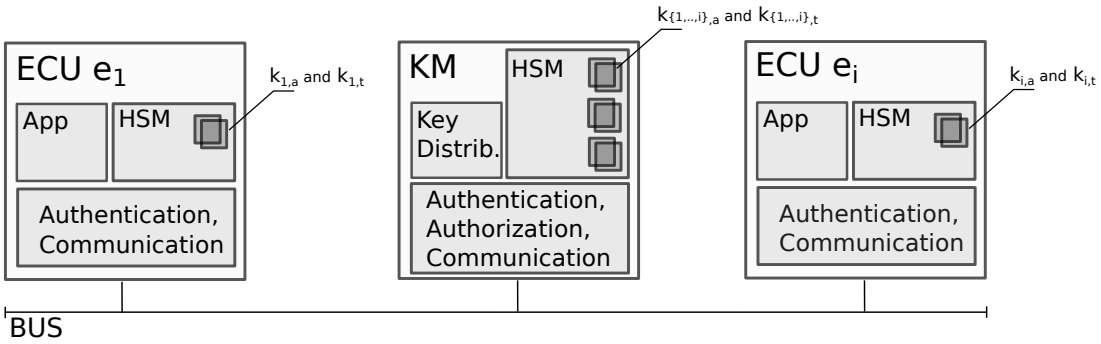

Fig. 1. Simplified deployment architecture. One ECU acts as Key Master node (KM). Symmetric keys are shared with every other ECU (e_1, e_i shown). All keys reside in the corresponding HSM. System wide policies are evaluated at the key master (Authorization).

6) Re-exporting the imported key $k_{s,v}$ for all ECUs e_i in g_x with the according transport and authentication keys $k_{i,t}$ and $k_{i,a}$ as blobs b_i.
7) Opening unauthenticated communication channels from KM to all e_i.
8) *At every* e_i: Importing authenticated key blob b_i ($= k_{s,v}$) into HSM of e_i.
9) Acknowledging successful import to KM.
10) *At* KM: acknowledging successful distribution to initiator e_1.

In Figure 2 we show a sequence diagram of this protocol. All steps that affect cryptographic material (i.e., the generation, import and export steps) are executed in the hardware security module of the affected ECU.

Please note that thanks to the HSM's use-flags, no ECU can impersonate the sender of a group despite of symmetric keys. This is also the rationale for placing key generation and re-keying at the sender's HSM, as the key master would otherwise be able to impersonate sending ECUs.

In compliance with current standards, we anticipate a limited lifetime of key material. Specifically, a session is envisaged to last for one drive cycle or a maximum of 48 hours, after which it is not valid for use anymore.

B. Transport Protocol on CAN

As vehicle bus systems are restricted by a number of constraints such as message payload, we make use of transport protocols. The restriction of, for example, CAN buses at only eight bytes of payload makes it impossible to transmit cryptographic keys or signatures in one packet. A standard procedure to achieve the transfer of large chunks of data over carriers with packet sizes is a segmentation of data. Additional control information is necessary for the individual data packets. At least, a sequence number needs to be kept to address out-of-order transmission and potential packet loss.

To transmit data, a sequence of data packets is sent to the receiving node(s). The segmentation of data creates an additional delay until all data is received. For n segments, this is at least n times the packet transmission and handling time, as well as the arbitration latency before every packet is admitted to the bus. This latency heavily depends on the bus load and the outcome of arbitrations, as CAN uses a bitwise CSMA/CA medium access arbitration. The arbitration latency directly relates to the priorities of other messages on the CAN bus.

The standard ISO 15765-2, which is used mainly for diagnosis purposes, offers segmentation of the payload. The protocol has been enhanced with a generic addressing scheme in [4] as CTP (common transport protocol), which builds on CAN 2.0 B, that supports an extended addressing scheme of 29 bit identifiers compared to 11 typically used.

In order to enable secure communication between ECUs, we enhance the protocol with a mandatory security header that supplies information about the types of cryptographic payload used for the current data (see [3] for further details).

C. Truncation of Cryptographic Authentication Codes

Not all communication of a vehicle demands for the same level of security and privacy. Most communication is never exposed and thus privacy infringing. Besides privacy requirements, integrity and authenticity are the most demanded requirements. In [5] a study was conducted, which includes a catalogue of over one hundred different requirements covering privacy, authenticity, and confidentiality for different automotive applications. These were combined with the risk and the attack potential, i.e., the likelihood of a security-related incident and the severity of a successfully mounted attack. In Figure 3 you can see an accumulation of low- and medium risk levels, which means that many application's security requirements can already be covered with basic security measures.

Given the fact that basic security can already protect most applications and that bandwidth is a scarce resource on automotive buses, we decided to allow MAC truncation. i.e. use of only fractions of a calculated MAC. According to NIST and FIPS recommendations [7] [8], cryptographic authentication codes should have a minimal length of 64 bits, when no additional measures to limit the validation rate are taken. Within our environment, we already have several environmental limitations, such as bus speed (low) and bus load (high). We furthermore set a hard limit for verification trials for any given key-handle to one hundred per second at the HSM. We can thus limit the size of authentication codes further and

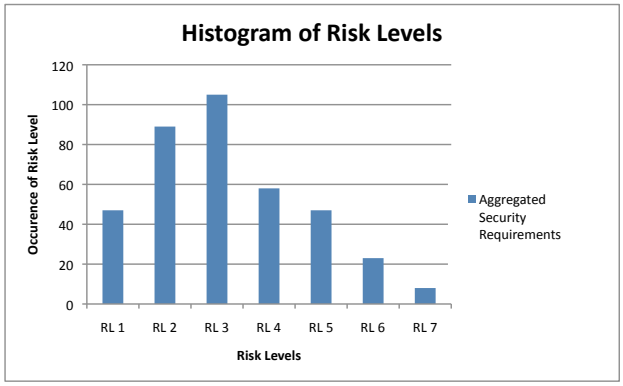

Fig. 3. Count (bins) of Risk Levels RL1—RL7 with regard to selected security requirements. A risk level is a combination of occurrence-probability and impact. The count shows an aggregate of security requirements for all use cases of EVITA (details in [5], [6], [2]).

allow a minimum length of 32 bits: Given our limitations, the expectancy time of a brute force collision attack for 32 bit MACs corresponds to over 35 weeks. We think this is a reasonable value, especially since additional measures, such as plausibility checks involving multiple information sources are usually employed.

IV. Analysis and Simulation

We have analyzed the performance of the key distribution protocol as well as the transport protocol with a number of different MAC fractions (i.e., frame segments). This is especially important for automotive environments, where time is a critical issue. We anticipate that communication channels for recurring messages that are known at startup, are already initialized during the wakeup procedure of a vehicle. This limits the setup latency of new channels to dynamic communication.

A. Key Distribution

We modeled the simplified scenario described in III: An emitting ECU, a key master and a receiving ECU, all connected to the same CAN bus and equipped with an HSM. The model includes several assumptions about frequency and load of processors and network media. Specifically, we assume that the HSM's cryptographic engine is clocked at 100MHz and requires 60 clock cycles, including scheduler cycles, for the encryption of one block (AES-based). The CAN bus is loaded at 30% and the arbitration for every packet is won with a probability of $p = 0.5$. Messages transmission includes overhead on CAN (39 bits header, 25 bits trailer, 3 bits idle, bit stuffing omitted) and overhead for the segmentation (2 bytes for the first frame, 1 for consecutive frames). The HSM keys are constructed as defined in [2] and thus carry header fields of 86 bits in addition to the key length of 256 bits and an authentication code of 128 bits. Thus, for the successful transmission of an HSM keystructure via CTP on CAN nine frames are sent: $9 = 1 + \lceil (470 - 48)/(7 \times 8) \rceil$.[2] We assume the CAN bus speed at 500kbit/s, as this speed is commonly used in vehicles.

This system has been modeled with the DIPLODOCUS UML profile [9], and simulated with the simulation engine of TTool [10]. The DIPLODOCUS profile—and its simulation engine—explicitly take into account the hardware elements of the system, that is, CPUs, buses, memories, and hardware accelerators (HSM).

A complete key distribution cycle including acknowledgements is, under the load and speed assumptions mentioned above, performed within $9ms$ in average. While this is not adequate for hard real-time applications, it is certainly feasible to perform the key exchange at boot time for all statically known communication groups. The current HSM implementation allows for every ECU to participate in 64 groups.

B. Transport Protocol with MAC fractions

For simulating the transport protocol, we use a more in-depth simulation of the arbitration phase. We modeled three nodes on a CAN bus in TrueTIME 2.0 [11], a Matlab-Simulink toolkit, which supports the simulation of CAN networks. One node generates high-priority background noise at 60%, which will always be prioritized over our payload. The second and third node model a controller and a sensor/actor node. For communication between sensor and controller we encapsulate payload with the secure transport protocol and varied the MAC lengths. You can see the resulting latencies in the table in Fig. 4.

MAC/bits	0	32	64	128	256	512
Minimum	0.0004	0.0007	0.0011	0.0016	0.0027	0.0059
Average	0.0011	0.0018	0.0021	0.0030	0.0041	0.0073
Maximum	0.0030	0.0041	0.0043	0.0050	0.0060	0.0090

Fig. 4. Latency of CTP CAN packets with eight bytes payload in seconds. Bus is at 60% load with higher priority traffic.

In order to eliminate protocol overhead, a small 32 bit MAC can also be directly added for every 32 bits of payload (i.e., to fit payload and MAC into a single frame). The first column shows expected delays for a single frame transmission. The simulation model includes a processing overhead of 0.4ms per send and 0.2ms per receive operation on each ECU.

V. Related Work

The vehicle domain has for a long time been driven by functional requirements. Security features have only been added to selected functions, such as immobilizers or remote door unlocking. Recent analyses have shown that the risk of attacks on vehicle bus systems is not anymore of theoretic nature [12]. Even supposedly well-secured functions, such as remotely unlocking the vehicle, have been proven faulty [13].

With the integration of new wirelelss access technologies in order to enable applications based on Car2X and internet communication, new security threats are posed against vehicular on-board networks. Furthermore, in order to enable certain communication-based safety functions, the trust in data from a source has to be assured with dedicated security measures within the vehicular on-board network. While most of research

[2]The first frame offers six bytes of payload; thus 48 bits are subtracted from the HSM key of 470 bits. Consecutive frames offer 7×8 bits of usable payload.

has focussed on securing external communication, the security of the on-board network has been only partly addressed, yet.

For upcoming Car2X communication, several approaches have been discussed in research. Most rely on asymmetric cryptography [14], but also group-cryptography [15] and timed-symmetric [16] mechanisms are taken into account.

For the in-vehicle architecture, numerous authors mention the need for security [17], [18], [19]. However, very few solutions have been proposed. A basis for security within the on-board network is the establishment of trust relations by pre-deploying keys, such as the in-vehicle key distribution mechanism which has been proposed in [20]. Their approach is based on a so-called Key Predistribution System. They use an $n \times n$ generation matrix, which is stored at a dedicated master ECU and parts (the rows) at all n ECUs. As ours, their system uses symmetric cryptography and trusted hardware. However, we see that the practicability of their approach is largely limited by the fact that the configuration must be static over the complete vehicle lifetime. Despite the fact that vehicle upgrades aren't possible without changing the complete keying matrix, maintenance work such as the replacement of ECUs is very complex. Another approach for in-vehicle security is discussed in [21]. The presented security framework also relies on a master/slave partitioning. There, session keys are only generated at the master.

Both approaches are fitted to embedded systems, but ignore important aspects of vehicle networks. Neither supports communication to multiple receivers, which is the core use case within current on-board networks, where data is sent to multiple receivers. In addition, the constraints of embedded vehicular networks, i.e., vehicular busses, and the practical implementation of the proposed approaches, has so far been left out.

VI. Conclusions and Future Work

With the vehicle being more and more inter-connected to external devices, vehicles and infrastructure services, new security threats are posed against the on-board network. Moreover, in order to ensure trust in information sent from a vehicle, on-board security measures need to provide assurance in the provided data. While recent research has widely discussed the security on the wireless link between the entities, on-board security has been only partly addressed and constraints of on-board networks merely considered. We have applied well-known key-distribution protocols to vehicular on-board architectures and proposed a solution for scaling the intended security level against specific application requirements. Based on a simulative approach, we have shown that most of the applications can be successfully deployed for on-board communication.

We are currently prototypically implementing our approach in a vehicular on-board environment in order to provide further results and analysis with respect to the tradeoff between security, bus overload and latency.

From a conceptual point of view, we are currently investigating the required assurance level for enabling Car2X communication applications and the impact on the required security-configuration within the on-board network.

Acknowledgements

This work was done in the scope of the EVITA FP7 EU Project under grant agreement FP7-ICT-224275. We want to thank everyone involved for their contributions. Special thanks goes to Marko Wolf and Timo Gendrullis from escrypt GmbH.

References

[1] F. Kargl, L. Buttyan, D. Eckhoff, P. Papadimitratos, and E. Schoch, "10402 Report – Working Group on Security and Privacy," in *Inter-Vehicular Communication*, ser. Dagstuhl Seminar Proceedings, Leibniz-Zentrum fuer Informatik, Germany, 2011.

[2] B. Weyl, M. Wolf, F. Zweers, T. Gendrullis, M. S. Idrees, Y. Roudier, et al., "Secure on-board architecture specification," EVITA Project, Tech. Rep. Deliverable D3.2, 2010.

[3] H. Schweppe, S. Idrees, Y. Roudier, B. Weyl, R. El Khayari, O. Henniger, et al., "Secure on-board protocols specification," EVITA Project, Tech. Rep. Deliverable D3.3, 2010.

[4] M. Busse and M. Pleil, "D1.2-10 data exchange concepts for gateways," EASIS, Tech. Rep., 2006.

[5] E. Kelling, M. Friedewald, T. Leimbach, M. Menzel, P. Säger, H. Seudié, and B. Weyl, "Specification and evaluation of e-security relevant use cases," EVITA Project, Tech. Rep. Deliverable D2.1, 2009.

[6] A. Ruddle, D. Ward, B. Weyl, S. Idrees, Y. Roudier, et al., "Security requirements for automotive on-board networks based on dark-side scenarios," EVITA Project, Tech. Rep. Deliverable D2.3, 2009.

[7] M. Dworkin, *Cipher Modes of Operation; the CMAC Mode for Authentication.* NIST Special Publication 800-38b. National Institute of Standards and Technology (NIST), May 2005.

[8] FIPS-198, *The Key-Hash Message Authentication Code (HMAC)*. FIPS Publication 198, Federal Information Processing Standards, March 2002.

[9] L. Apvrille, W. Muhammad, R. Ameur-Boulifa, S. Coudert, and R. Pacalet, "A UML-based environment for system design space exploration," in *13th IEEE ICECS*, Nice, France, Dec 2006, pp. 1272 –1275.

[10] D. Knorreck, L. Apvrille, and R. Pacalet, "Fast simulation techniques for design space exploration," in *47th International Conference Objects, Models, Components, Patterns*, vol. 33, Zurich, Jun. 2009, pp. 308–327.

[11] B. A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-e. Å rzén, "Analysis and Simulation of Timing," *IEEE Control Systems Magazine*, no. June, pp. 16–30, 2003.

[12] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, et al. "Experimental security analysis of a modern automobile," in *31st IEEE Symposium on Security and Privacy*, vol. 31, 2010.

[13] A. Francillon, B. Danev, and S. Capkun, "Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars," in *18th Annual Network & Distributed System Security Symposium*. Cryptology ePrint Archive, Report 2010/332, 2010.

[14] F. Kargl, P. Papadimitratos, L. Buttyan, M. Müter, E. Schoch, et al., "Secure vehicular communication systems: implementation, performance, and research challenges," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 110–118, Nov. 2008.

[15] J. Guo, J. Baugh, and S. Wang, "A Group Signature Based Secure and Privacy-Preserving Vehicular Communication Framework," *2007 Mobile Networking for Vehicular Environments*, pp. 103–108, 2007.

[16] Y.-c. Hu and K. P. Laberteaux, "Strong VANET Security on a Shoestring," in *escar*, 2006, pp. 1–9.

[17] A. Kung (Ed.), "Security architecture and mechanisms for V2V / V2I," Sevecom Project, Tech. Rep. Deliverable D2.1, 2008.

[18] T. Eymann and M. Busse, "Deliverable D1.2-12 Security and Firewall concepts for gateways," EASIS-Project, Tech. Rep., 2006.

[19] M. Gerlach, A. Festag, T. Leinmüller, G. Goldacker, and C. Harsch, "Security architecture for vehicular communication," in *Fourth International Workshop on Intelligent Transportation (WIT2007)*, 2007.

[20] H. Oguma, A. Yoshioka, M. Nishikawa, R. Shigetomi, A. Otsuka, and H. Imai, "New Attestation Based Security Architecture for In-Vehicle Communication," in *IEEE GLOBECOM 2008*, pp. 1–6.

[21] H. Bar-el, "Intra-Vehicle Information Security Framework," in *escar*, 2009.