Automotive Security: Cryptography for Car2X Communication

Dr. Torsten Schütze[*][†]
Rohde & Schwarz SIT GmbH
Hemminger Str. 41
70499 Stuttgart, Germany

March 1, 2011

In this paper we report on security aspects of Car2Car and Car2Infrastructure communication (Car2X communication). We describe prominent use cases of Car2X communication and explain the communication scenarios. Typical security objectives and resulting security mechanisms will be discussed. In the focus of this paper is the IEEE 802.11p standard (Automotive WLAN) and its corresponding security standard, IEEE 1609.2. We explain in detail the cryptographic components of this standard, which is currently the most advanced automotive security standard worldwide. In addition we give a rationale on the algorithms selected and discuss alternatives. At the end of the paper we describe consequences for the practical realization in hard- and software.

Key words: Car2X communication, IEEE 802.11p, security, IEEE 1609.2, elliptic curve cryptography, electronic signatures, authenticated encryption

Acknowledgment: The author would like to thank his former colleague Dr. Jamshid Shokrollahi for many valuable discussions, especially on the hardware implementation aspects.

1 Introduction

1.1 Introduction and historical development

Safety has a long tradition in automotive industry. Almost every automotive engineer knows about Safety Integrity Levels (SIL); the automotive industry is actively pushing the new standard ISO 26262 *Road vehicles – Functional safety*. Security, on the other hand, the degree of protection against *malicious* attacks, is much more unknown to automotive manufacturers and suppliers. Only in the last ten to fifteen years, car industry started to worry about attacks on their systems. Cryptography plays a major role in securing these systems.

Cryptography in a car started with Remote Key-less Entry (RKE) and Key-less Go in the mid of the 1990s, followed by electronic immobilizers. In isolated systems, such as a single car, we have a lot of proprietary solutions. Relatively little is publicly known about the cryptographic mechanisms used except when news on weaknesses or complete breaks leak, for example, the break of the proprietary 40-bit encryption on Texas Instrument transponders, the break of NXP's proprietary HITAG2 encryption (see Karsten Nohl's talk at ESCAR 2010) or the BMW X5 LIN bus back-door. The RKE system KeeLoq, used in garage door openers and some cars, became rather infamous, see the attacks in [9, 16, 11]. After that series of attacks and the fact that after a 16-year decline, car theft in Germany rose again in 2009, industry

[*] `torsten.schuetze@rohde-schwarz.com`

[†] The author worked until September 2010 at Robert Bosch GmbH, Corporate Research and Advance Development on the topics automotive security, cryptography and Car2X standardization.

began to enforce their work on strong cryptography for authentication methods. For example, at last years Embedded World Atmel presented a new single-chip AES-128 immobilizer and remote key-less entry AVR micro-controller for combi-key applications, see also ESCAR 2007/2010. Even asymmetric solutions are being discussed now, for example the ECC RKE proposal from Fraunhofer SIT based on work by Siemens [8].

Other areas of cryptography in a car are secure access to multifunction instruments, securing odometers, and tuning protection. Although these uses cases lead to standard security problems, very often proprietary "cryptographic" solutions are still implemented. Standardization within automotive security began in Germany with Herstellerinitiative Software (HIS), an interest group of major car manufacturers. They define standards as the secure flashing standard.

In Germany, two large telematics infrastructure projects, namely Toll Collect and the Digital Tachograph, use cryptography and smart cards extensively. The first project, Toll Collect, for road tolling just uses certified smart cards, the system is, up to now, not open and not certified according to Common Criteria. Little was known about the system and security architecture, only recently due to European harmonization requirements cryptographic details have been published [21].

The digital tachograph project in Europe is the first security certified automotive project (ITsec E3 high or Common Criteria EAL 4+). The system started in May 2006, many of the security challenges were completely new to automotive industry. A nice overview on the system and security architecture is given in [12].

Nowadays special conferences like ESCAR or VDI Automotive Security on security and cryptography for automotive are established. Automotive security even becomes a topic at general security conferences, for example, the invited talk of Hovav Shacham at CHES 2010 [25] or the paper [17]. Currently, automotive manufacturers and suppliers built-up specialist know-how on security and cryptography or use highly qualified consultants.

1.2 Car2X communication

Many of above closed systems use proprietary security mechanisms. Automotive industry is just in the process of leaving the principle Security by Obscurity toward to open standards. Up to now, weaknesses of the systems had relatively little impact on other cars or passengers. But with Car2X communication we have a new quality. Information from the outside world over potentially insecure channels directly influences the behavior of your own car. In the future, when driving your car you will leave data footprints in the world, may be even personal data. Car2Car and Car2Infrastructure communication connect cars to a completely new communication system, a hopefully Intelligent Transportation System (ITS).

Every car will act as a receiver and as a sender. In addition, a car will act as a relay and will route messages to other cars not directly within reach of the original sender. Weaknesses in the security architecture of Car2X communication will directly influence a great number of other cars. Safety and privacy can only be assured when establishing a sound security system. Therefore, very early in the beginning of the concept studies on Car2X communication it became clear that security is an essential part of Car2X communication. Without sound security architecture there will be no successful Car2X communication, neither technologically nor politically through acceptance by the users.

Consequently, the approach taken was different from older approaches: universities and research institutes were participating right from the start in designing a secure system.

1.3 Overview on project landscape and standardization work

There are a large number of publicly founded projects on the topic secure Car2X communication. For Europe we just cite the projects SEVECOM[1] (Secure Vehicular Communication, 2006–2008), EVITA[2] (E-safety vehicle intrusion protected applications, 2008–2011), and simTD[3] (Sichere Intelligente Mobilität Testfeld Deutschland – Safe and Intelligent Mobility Test Field Germany, 2008–2012).

In SEVECOM the threat, vulnerability and risk analysis followed by a general security architecture has been in focus. SEVECOM started to define specific security mechanisms for Car2X communication.

[1] http://www.sevecom.org

[2] http://www.evita-project.org

[3] http://www.simtd.de

EVITA is focused on the design of a secure automotive on-board network, i.e., secure hard- and software components as well as protocols for communication *inside the car* have been developed. As a result we have three different types of security controllers (light, medium and full hardware security modules corresponding to sensor, ECU and ECU Car2X communication), which perform symmetric and/or asymmetric cryptography.

simTD is the first large test field for Car2X functionalities. Communication and application layer are definitely the subject of this test, security is only of minor importance.

Besides these publicly funded projects in Germany/Europe there are at least three further European interest groups dedicated to Car2X security

- Car2Car Communication Consortium (C2C CC), an industrial community lead by European automotive manufacturers complemented by suppliers and research institutes, especially WG Security,
- eSafety Security Working Group of the eSafety Forum / European Union, and
- Working Group 5 Security of ETSI (European Telecommunications Standards Institute), Technical Committee (TC), Intelligent Transport Systems (ITS).

The core standardization in the area of Car2X security in Europe takes places at ETSI, it received the official mandate to standardize Car2X. Essential input comes from C2C CC Security WG, C2C CC WGs act as preparatory platforms and discussion forums.

In the US, the Trial-Use Standard IEEE 1609.2 [23] is the relevant standard for Car2X security. In our opinion, the concrete technical standardization of security functions is most advanced in IEEE 1609.2. Within C2C CC and ETSI the American trial use standard is used as a good template for securing automotive WLAN 802.11p.

2 Car2X communication infrastructure

2.1 Communication network

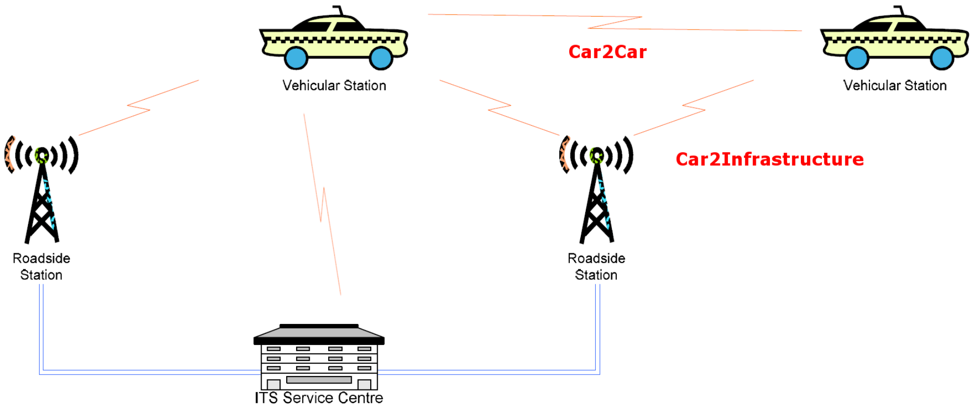

Figure 1: Communicating parties within a Car2X system

Within a Car2X system there are the following communicating parties: car (vehicle), roadside station (RSS) and network infrastructure, see Fig. 1. Therefore we have these typical communication scenarios

- short-time communication between car and car (Car2Car),
- short-time communication between car and roadside station (Car2Infrastructure),
- communication between roadside station and network infrastructure, and
- communication between car and network infrastructure.

The first two communication relations are especially important and interesting from a technical point of view. They have the most ambitious requirements since they are very short-time, consider for example two approaching cars with 140 km/h; they require extremely low latency; and they cannot rely on an existing fixed infrastructure as router or access points. Thus, Car2X communication is a typical example of a Mobile Ad Hoc Network (MANET), a so-called Vehicular Ad Hoc Network (VANET).

The last point requires clarification: Since the communication between cars shall be possible without additional infrastructure, messages have to be routed over intermediate cars in case of insufficient message range. These cars work as relays (multi-hop routing).

2.2 Communication protocol

In some (European) Car2X groups, the mobile phone network has been considered as the underlying communication network. Of course, the reason for this is the existence of infrastructure and the necessary coverage right from the beginning. Due to the high latency requirements the future generation of mobile networks, Long Term Evolution (LTE), has been considered. Estimates show that mobile networks using LTE would be an option for a large part of applications, but not for all.

The variability in requirements for Car2X is shown by the following picture with application area and time scale.

traffic efficiency   post-crash   car autonomous systems   pre-crash   passive/active safety

⟵ minutes   ⟷   seconds   ⟷   milliseconds ⟶

It should be clear that applications from the traffic efficiency area and so-called comfort functions (e.g. infotainment) are not time critical and can therefore be realized over mobile phone networks. For applications from the area pre-crash and active/passive safety the timing requirements are much more extreme, therefore mobile phone networks are often not suitable.

Car2X communication in a narrower sense will use the communication standard IEEE 802.11p, also known as WLAN-p, Automotive WLAN or Wireless Access in Vehicular Networks (WAVE). IEEE 802.11p is an amendment to the well-known WLAN 802.11 a/b/g/n standards, which operates in the frequency range of 5.9 GHz. IEEE 802.11p has been designed for situations where rapidly changing communication environments require that transactions are completed in much less time than that would be possible with infrastructure or common 802.11 Ad hoc networks. To give an impression on the necessary latencies we show the four categories for latencies between triggering event (crash) and resulting action (alarm or brake) considered in C2C CC: a) $T \leq 50\,\mathrm{ms}$, b) $T \leq 100\,\mathrm{ms}$, c) $T \leq 200\,\mathrm{ms}$, d) best effort. Within the Car2Car 5.9 GHz band, there are dedicated bands for road safety (G5A) and non-safety applications (G5B). The band G5C has to be shared with other applications.

Other major differences to the core IEEE 802.11 standards are: there is no MAC multicast on the Control Channel, there is no MAC encryption (as WPA) or fragmentation. We have a Distributed Congestion Control, thus there exist mechanisms on all layers, e.g., CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) for layer 2, adjustment of packet rate for layer 3 or data rate on the application layer. The communication range of IEEE 802.11p is generally over line-of-sight distances of less than 1000 m.

2.3 Applications

There are a large number of applications for Car2X communication. In project simTD there are already twenty different applications, ranging from efficiency and comfort functions to car safety, which have to be implemented and tested. ETSI defined a so-called basic set of applications (BSA). In the beginning, BSA was mandatory for all Car2X stations, later-on this set was no longer mandatory but has the highest priority when implementing the standard. The basic set of applications consists of approx. 32 use cases, leading to the following ten applications, see [3]:

1. Stationary vehicle warning — accident vehicle problem
2. Traffic condition warning (includes traffic jam ahead warning)
3. Signal violation warning (includes stop sign violation)
4. Road work warning
5. Collision risk warning from RSS
6. Decentralized floating car data — precipitations, road adhesion, visibility, wind
7. Regulatory/Contextual speed limits
8. Traffic information & recommended itinerary

9. Limited access, detour notification
10. In-vehicle signage

2.4 Fundamental message types

Despite these different use cases and applications there are only two fundamental message types:

- Cooperative Awareness Message (CAM): periodically broadcast information about the station to other nodes in the vicinity (beaconing), will not be routed in general
- Decentralized Environmental Notification Message (DENM): generated upon detecting an event and contains information about this event. DENMs are addressed via geo-positioning since they are relevant for a defined geographic area

In this paper we cannot go into detail on CAM and DENM, we refer to [4, 5] instead.

For the security considerations it suffices to know that CAM contain a time stamp, car or RSS specific fixed data (identity) as well as position, velocity and direction. Using special flags, emergency vehicles can signal Siren in Use or braking activities can be sent. The length of a CAM block is about 64 bytes. A CAM will be sent in message broadcast mode, sometimes in unicast mode.

The most important difference between CAM and DENM is, that DENMs contain geographical data: about the event area, about the target area, about the relevance area and the networking area in which resending of messages shall take place. DENM will be transported using geo-networking or infrastructure relays. The application information in DENM are event management information, priority, generation time, validity period, etc. and of course information on the event itself. For example, the type of event, the severity of the event, the reliability of its detection and a human readable description of the data. Every DEN message has an action ID which is unique for each event and contains information about the originator (privacy concerns!). DEN messages are typically much longer than CAMs, their length depends very much on the event type. DENM will be transported in broadcast or unicast mode.

Table 1: Six C2X communication patterns with typical use cases

Communication pattern	Type	Use cases
C2C cooperative awareness	broadcast	intersection collision warning, left turn assistant, cooperative forward collision warning, blind spot warning
C2C unicast exchange	unicast	pre-crash sensing (to share more information at a faster rate)
C2C decentralized environmental notification	broadcast/unicast	road collision warning, post-crash warning
Infrastructure-to-car (one way)	broadcast	post-crash warning, low bridge warning
Local RSS connection	unicast	free-flow tolling, software updates
IP RSS connection	unicast	fleet management, in-route hotel management, web browsing

Besides these two message types there is another data object stored locally at each station called Local Dynamic Map which represents the complete information on the traffic and safety related circumstances. The Local Dynamic Map is updated from information received in CAM and DENM and from data acquired from its own sensors.

3 Security objectives

A detailed threat, vulnerability and risk analysis by ETSI can be found in the Technical Report [6]. As usual in Common Criteria the security objectives are mapped to functional security requirements. In Table 2 we show the resulting security objectives.

Table 2: Security objectives for Car2X communication, Source: [6]

Confidentiality	
Co1	Information sent to or from an authorized ITS user should not be revealed to any party not authorized to receive the information
Co2	Information held within the ITS-S should be protected from unauthorized access
Co3	Details relating to the identity and service capabilities of an ITS user should not be revealed to any unauthorized 3rd party
Co4	Management Information sent to or from an ITS-S should be protected from unauthorized access
Co5	Management Information held within an ITS-S should be protected from unauthorized access
Co6	It should not be possible for an unauthorized party to deduce the location or identity of an ITS user by analyzing communications traffic flows to and from the ITS user's vehicle
Co7	It should not be possible for an unauthorized party to deduce the route taken by an ITS end-user by analyzing communications traffic flows to and from the ITS end-user's vehicle
Integrity	
In1	Information held within an ITS-S should be protected from unauthorized modification and deletion
In2	Information sent to or from a registered ITS user should be protected against unauthorized or malicious modification or manipulation during transmission
In3	Management Information held within a ITS-S should be protected from unauthorized modification and deletion
In4	Management Information sent to or from an ITS-S should be protected against unauthorized or malicious modification or manipulation during transmission
Availability	
Av1	Access to and the operation of ITS services by authorized users should not be prevented by malicious activity within the ITS-S environment
Accountability	
Ac1	It should be possible to audit all changes to security parameters and applications (updates, additions and deletions)
Authenticity	
Au1	It should not be possible for an unauthorized user to pose as an ITS-S when communicating with another ITS-S
Au2	It should not be possible for an ITS-S to receive and process management and configuration information from an unauthorized user
Au3	Restricted ITS services should be available only to authorized users of the ITS

4 Possible security mechanisms

At first sight, there seems to be an antagonism with these security objectives: on one hand only authorized users should be able to participate in the system, on the other hand privacy aspects require anonymity or non-traceability.

Abstracting from the very specific security goals in Table 2, the main cryptographic problem in Car2X communication is to protect many, relatively short messages, which have to be authenticated, i.e., protect integrity of sender and of data, in a mobile Ad Hoc network. Sometimes the data has to be encrypted, too. The communication happens in broadcast or unicast mode, latencies are very critical.

Non-repudiation of C2X data in a cryptographic sense is discussed very controversial within C2X CC Security WG. From an application point of view only a very few applications, e.g., eCommerce applications with monetary value, require non-repudiation of data. But the solution selected, namely digital signatures on all data, gives this property for free. Strictly speaking, non-repudiation of data is not required for all use cases.

Confidentiality of data is not always required as well. Due to their very nature cooperative awareness messages (CAM) are not confidential per se. The situation is different with Decentralized Environmental Notification Messages (DENM). Depending on the event they can be confidential and therefore have to be encrypted. In addition, one has to check if any of these data is personal data in the sense of Federal Data Protection Act (BDSG) in Germany or can be made personal data, for example, long term identities.

Above standard security objectives (integrity of data and of sender, freshness of data, non-repudiation, optional confidentiality) can be achieved by different cryptographic means, see Table 3. As one can see, none of the mechanisms is optimal for all use cases.

Table 3: Security goals for C2X communication and possible security mechanisms

<table>
<tr><th>security mechanism →
↓ security goal</th><th>symmetric</th><th colspan="2">asymmetric</th><th>hybrid</th></tr>
<tr><td>integrity of data</td><td>Message Authentication Code (MAC) over data</td><td colspan="2">digital signature over data</td><td>MAC over data</td></tr>
<tr><td>integrity of sender</td><td></td><td colspan="2"></td><td></td></tr>
<tr><td>non-repudiation</td><td>–</td><td colspan="2"></td><td>digital signature</td></tr>
<tr><td>timeliness / freshness</td><td>MAC over data and time variant parameters[1]</td><td colspan="2">digital signature over data and time variant parameters</td><td>MAC over data and time variant parameters</td></tr>
<tr><td>confidentiality (optional)</td><td>encryption</td><td colspan="2">encryption of keys and small data only</td><td>encryption / decryption of ephemeral symmetric key with public / private key, encryption of bulk data with symmetric key</td></tr>
<tr><td>unicast communication</td><td>+</td><td colspan="2">+</td><td>+</td></tr>
<tr><td>broadcast communication</td><td>−</td><td colspan="2">+</td><td>?</td></tr>
<tr><td>performance</td><td>very fast</td><td colspan="2">slow – very slow</td><td>good</td></tr>
<tr><td></td><td></td><td>ECC</td><td>RSA</td><td></td></tr>
<tr><td></td><td></td><td>signature generation slow</td><td>signature generation very slow</td><td></td></tr>
<tr><td></td><td></td><td>signature verification ≈ 2 × signature generation</td><td>signature verification fast (for $e = 3$ and $e = F_4 = 2^{16} + 1$)</td><td></td></tr>
<tr><td>size of MAC / signature (bit)</td><td>64–96</td><td>ECC</td><td>RSA</td><td>64–96 MAC, 448–512–2048 signature</td></tr>
<tr><td></td><td></td><td>448–512</td><td>512–2048</td><td></td></tr>
<tr><td>applications / standards / examples</td><td>GSM, UMTS, banking</td><td colspan="2">IEEE 1609.2 WAVE Security, C2X CC Security WG, SEVECOM architecture</td><td>eMail security (OpenPGP, S/MIME), IEEE 1609.2 + TESLA</td></tr>
</table>

[1] time-stamp, sequence counter, or nonce

4.1 Discussion of possible methods

Asymmetric methods At first and foremost in the European groups as C2X CC Security WG and SEVECOM, a purely asymmetric solution in the form of digital signatures has been discussed, i.e., all parties of the system obtain digital certificates and sign all their messages with their private key. The recipient verifies the integrity of sender and of data by signature verification.

In all Car2X systems we need long-time security, i.e., methods and parameters have to be selected in such a way, that these systems remain secure over a long time period, typically development time + production time + warranty time / run-time, i.e., 25–30 years. The lifetime of a Car2X message is rather limited, but the certificates have to stay secure for a long time.

Due to the long-time security aspect and due to very bad performance in signature generation as well as prohibitive long signature length, RSA-based signatures have been discarded very early.[4]

Instead of RSA-based methods Elliptic Curve Cryptography (ECC) has been considered right from the start. Elliptic curves are given by the solution of the equation $y^2 = x^3 + ax + b$ (or similar equations) over a suitable finite field together with the point at infinity $\mathcal{O}$. From the three possible variants $\mathbb{F}_p$ with

[4] Unfortunately, due to several constraints, in simTD RSA PKCS#1 signatures with 512 bit and small public exponent will be used. Hopefully, this algorithm and signature length will never go live.

$p > 3$ prime, $\mathbb{F}_{2^m}$, and $\mathbb{F}_{p^m}$, $p > 3$ prime, the finite field $\mathbb{F}_p$ has been selected. The Elliptic Curve Digital Signature Algorithm (ECDSA) [2] will be used as signature algorithm.

ECC-based methods have moderate key and signature length. For medium to high security requirements they are favorable over RSA-based methods. Only RSA-signature verification with small exponent ($e = 3$ or $e = F_4 = 2^{16} + 1$) is competitive, but cars not only have to verify, sometimes they have to sign messages by themselves. But RSA signature generation for 2048 bit length or longer is prohibitive expensive.

Special methods, for example, from Post Quantum Cryptography, specially designed for long-time security, e.g., Merkle Hash Tree methods, have very good security properties, but extreme length of public key and signature. Therefore they seem unsuitable for this use case.

Symmetric methods When the author heard for the first time of the signature proposal in Car2X, he thought that this would be the largest purely asymmetric system in the world. Mobile phone and banking systems, currently probably the largest cryptographic systems in the world, are based on symmetric methods only.

A sophisticated security system can be established with key derivation (from a secure master key) and key encryption/transport. All security objectives, except non-repudiation — which was not required in all use cases — can be achieved with symmetric methods only. However, symmetric methods are not optimal with broadcast messages when only one or a few messages will be sent and no previous security association exists.

Hybrid methods Hybrid methods combine both symmetric and asymmetric methods. In first generation standards like OpenPGP or S/MIME this combination is done ad hoc. A temporal symmetric key is encrypted by a public key encryption algorithm. Integrity and encryption of mass data is done by symmetric schemes.

Modern hybrid encryption methods like those in [1] perform the combination of symmetric and asymmetric schemes in a more systematic way. Thus, results of provably security apply. For example, the scheme ECIES is proven IND-CCA2 secure in random oracle model.

In Section 5 we will see that the WAVE Security standard consists of a purely asymmetric digital signature scheme, a purely symmetric authenticated encryption scheme and a hybrid encryption scheme, so the best of all worlds.

4.2 Performance requirements

For DSA-like methods, signature verification is the most time consuming part of the security mechanisms. The sender has to generate one signature per message (in case we do not have a pseudonym change at this time). But the receiver has to verify two signatures (message and certificate). The message length will be increased by the length of the signature and the length of the certificate.

Now we have to estimate how many signature verifications we have to do in a car in a worst case scenario. In [24] we find: “Assuming a neighborhood density of 200 vehicles and beaconing rates between 1 and 10 Hz, each vehicle needs to generate between 1 and 10 signatures per second and needs to verify between 400 and 4000 signatures per second.” Prof. Paar of Bochum University and escrypt GmbH often cite the requirement of up to 5000 signature verifications per second.

The worst case load scenario, i.e., the maximum number of messages received, probably occurs for secure beaconing of CAM at a large highway inter-junction during congestion. Assuming four times three lanes and an operating distance of WLAN 802.11p of 500 m we obtain a neighborhood density of about 400 cars. With 10 Hz beaconing frequency this gives 8000 signature verifications per second. With congestion control for CAM at the network layer and congestion control at the application layer (there is no need to send 10 nearly identical messages while at rest), see Distributed Congestion Control in previous sections, the estimated number of 4000 signature verifications per second could be very realistic for this scenario,.

simTD assumes 20 incoming and 2 outgoing messages per second, i.e., 40 signature verifications and 2 signature generation per second for their test-field with a limited number of cars and RSS.

For comparison we cite the current figures of the FPGA box built by escrypt GmbH for CAMP/VST and EVITA, see [14]: They get up to 400 signature verifications per second using NIST $\mathbb{F}_p$ curves.

Summarizing, the performance requirements are really huge, there is no off-the-shelf crypto co-processor capable of such a signature verification load. Even with customized hardware we still have a gap of factor ten.

4.3 Privacy requirements

Privacy is of special importance in the Car2X communication scenario. Security objectives Co6 and Co7 state that it should not be possible for unauthorized parties to deduce the past or future location, route and identity of an ITS participant based on Car2X communication data.

This security goal is, strictly speaking, much stronger than those of current mobile phone systems. With current mobile phone networks such as GSM it is possible for unauthorized parties to deduce the location and identity (IMSI), see the talks at recent Chaos Communication Congresses.

In jurisdiction of Germany it is widely assumed that IP numbers belong to personal data protected under Federal Data Protection Act (BDSG) or at least they can be made personal when joined with other data. The same would occur with long-time identities, e.g., in CAM or DENM. To protect drivers against such track and trace we need strict privacy rules. This means especially that personal data, for example, in some DENM messages, have to be encrypted.

But this is not sufficient: encrypted data can still be traced and assigned to certain identities. Therefore, in Car2X a pseudonym approach has been selected. Every participant of the system obtains a number of different pseudonyms, i.e., certificates with different pseudonyms belonging all to the same person, which he changes in regular intervals. If these changes occur often enough the tracked data cannot be assigned to a single person or car. It is clear that with this solution we do not have a full anonymous Car2X network, just pseudonyms. It depends very much on the concrete technical realization of the system to fulfill the security objectives. Note, that it is not sufficient to reach pseudonym just at one communication layer. If host-names are made pseudonymous, data can still be tracked based on MAC or IP addresses. Similar occurs for Car2X networking.

The author sees privacy of Car2X data as one of the most challenging topics within Car2X security. But as there is no success for Car2X without security, there will be no success (acceptance) for Car2X security without privacy.

It is very interesting to note that current investigations on privacy in the US, where there is traditionally no such strict privacy jurisdiction, are much deeper than in European circles.

To give an impression of the political dimension of privacy, we cite from a discussion in the eSafety WG:

> - Privacy: One of the major consumer concerns about the Vehicle Communication (VC) technology is its potential influence on privacy. Although there are solutions that can provide vehicle and driver anonymity, this may negatively affect the liability of the network. In fact, if vehicles are totally anonymous, the drivers involved in an accident who flee the scene may not be easily identified. Hence, a balance should be kept between the privacy and liability of drivers. [internal working document]
>
> *It seems to me what is being nonchalantly proposed here is very wrong and very worrying from the point of view of European human rights. . . .*
>
> *I am afraid there is no way to sugar this message – unless you have a cast iron solution for anonymity, the entire concept of the system is totally illegal – it is amazing it should have progressed so far without anybody noticing this fact. – Caspar Bowden, Chief Privacy Adviser Microsoft, June 14, 2010*

5 Cryptographic mechanisms of IEEE 1609.2

5.1 Description of cryptographic components

The IEEE 1609.2 security standard basically consists of three components

a) digital signatures using ECC over $\mathbb{F}_p$, specifically Elliptic Curve Digital Signature Standard (ECDSA) with NIST $\mathbb{F}_p$ curves [2]

b) asymmetric encryption with ECC, specifically Elliptic Curve Integrated Encryption Scheme (ECIES) [1, 22]

c) purely symmetric scheme: authenticated encryption, specifically counter mode encryption and CBC-MAC with AES (AES-CCM) [10]

A special compact form of certificates is defined, the so-called WAVE certificate. The secured messages are of type unsecured, signed or encrypted.

The security level of IEEE 1609.2 WAVE secured messages is 112 bit for short-time messages, 128 bit else. This leads to the following requirements: ECDSA with NIST P-224 curve and SHA-224, ECDSA with NIST P-256 curve and SHA-256, ECIES with NIST P-256 curve, AES-128 encryption.

```
ECDSASign( message $m$, private key $d$, generator $G$, order $n$)
1. Compute $H(m)$ and convert the bit string to integer $e$.
2. Select random number $k \in \mathbb{Z}$ with $0 < k < n$, ephemeral key, and compute $kG$.
3. Compute $r = \pi(kG) \bmod n$. If $r = 0$, goto 2.
4. Compute $k^{-1} \bmod n$.
5. Compute $s = k^{-1}(e + dr) \bmod n$. If $s = 0$, goto 2.
6. Signature for message $m$ is the pair $(r, s) \in F_n \times F_n$.
```

```
ECDSAverify( signature $(r, s)$, message $m$, generator $G$, order $n$, public key $Q$)
1. If $r \notin [1, n-1]$ or $s \notin [1, n-1]$, deny signature as invalid.
2. Compute $H(m)$ and convert the bit string to integer $e$.
3. Compute $w = s^{-1} \bmod n$.
4. Compute $u_1 = ew \bmod n$ and $u_2 = rw \bmod n$.
5. Compute $X = u_1 G + u_2 Q$. If $X = \mathcal{O}$, deny signature as invalid.
6. Compute $\nu = \pi(X) \bmod n$.
7. Accept signature iff $\nu = r$.
```

Figure 2: Pseudocode of ECDSA sign and verify with private key $d \in \mathbb{Z}$, $0 < d < n$, and public key Q, $Q = dG$, hash function H and projector $\pi(P) = \pi(x_P, y_P) = x_P$.

```
AES_CCM_Generate_Encrypt( key $K$, payload $P$)
1. (Nonce $N$, Associated Data $A$, Payload $P$) $\longrightarrow B_0, B_1, \ldots, B_r$ /* Nonce 96 bit, Assoc. data $\emptyset$ */
2. $Y_0 = \mathrm{Enc}_K(B_0)$
3. for $i := 1$ to $r$ do $Y_i := \mathrm{Enc}_K(B_i) \oplus Y_{i-1}$
4. $T := \mathrm{MSB}_{\mathrm{Tlen}}(Y_R)$; /* Tlen=128, 2-4 CBC-MAC */
5. Counter generation $Ctr_0$, $Ctr_1$,...,$Ctr_m$ with $m = \lceil \mathrm{Plen}/128 \rceil$
6. for $j := 0$ to $m$ do $S_j := \mathrm{Enc}_K(Ctr_j)$
7. $S = S_1 || S_2 || \ldots || S_m$ /* 5-8 CTR mode encryption */
8. return $C = (P \oplus \mathrm{MSB}_{\mathrm{Plen}}(S) \,||\, T \oplus \mathrm{MSB}_{\mathrm{Tlen}}(S_0))$
```

```
AES_CCM_Decrypt_Verify( key $K$, cipher-text $P$, Nonce $N$, Assoc. data $A$)
1. if Clen $\leq$ Tlen, then return INVALID
2. Counter generation $Ctr_0$, $Ctr_1$,...,$Ctr_m$ with $m = (\mathrm{Clen} - \mathrm{Tlen})/128$
3. for $j := 0$ to $m$ do $S_j := \mathrm{Enc}_K(Ctr_j)$
4. $S = S_1 || S_2 || \ldots || S_m$ /* 2-5 CTR mode decryption */
5. $P = \mathrm{MSB}_{\mathrm{Clen-Tlen}}(C) \oplus \mathrm{MSB}_{\mathrm{Clen-Tlen}}(S)$
6. $T = \mathrm{LSB}_{\mathrm{Tlen}}(C) \oplus \mathrm{MSB}_{\mathrm{Tlen}}(S_0)$
7. if N, A, or P is not valid, then return INVALID else (Nonce $N$, Associated Data $A$, Payload $P$) $\longrightarrow B_0, B_1, \ldots, B_r$
8. $Y_0 = \mathrm{Enc}_K(B_0)$
9. for $i := 1$ to $r$ do $Y_i := \mathrm{Enc}_K(B_i) \oplus Y_{i-1}$
10. if $T \neq \mathrm{MSB}_{\mathrm{Tlen}}(Y_R)$, then return INVALID, else return $P$
```

Figure 3: AES-CCM with AES-128, a nonce length of 96 bit and a tag length of 128 bit is used for authenticated encryption.

Modern hybrid encryption schemes consist of a Data Encapsulation Mechanism (DEM) and of a Key Encapsulation Mechanism (KEM). For ECIES, ElGamal-based encryption is used as KEM. Two keys, encryption and MAC key, are derived for the Data Encapsulation Mechanism which uses symmetric schemes. There are various sources for the definition of ECIES, for example [22, 1, 26], unfortunately they differ in subtle ways. The schematics of ECIES is shown in multiplicative form in Figure 4, taken from [7].

5.2 Rationale

As far as the author knows, no such rationale on the security mechanisms in IEEE 1609.2 has been given so far. Some of the following has been discussed with the Technical Editor of 1609.2, William Whyte, who prepares the Rationale for the forthcoming edition of the standard.

a) The digital signature algorithm ECDSA comprises the main mechanism. Every message, for example in broadcast mode, will be signed with the private key of the sender. The corresponding public key may refer to a pseudonym. The certificate is sent together with the signed message. The receiver first verifies the public key in the certificate and then the message, thus he can verify the integrity of sender and data without prior security relation with the sender. Because of the time stamp inside the message, freshness will be secured as well.

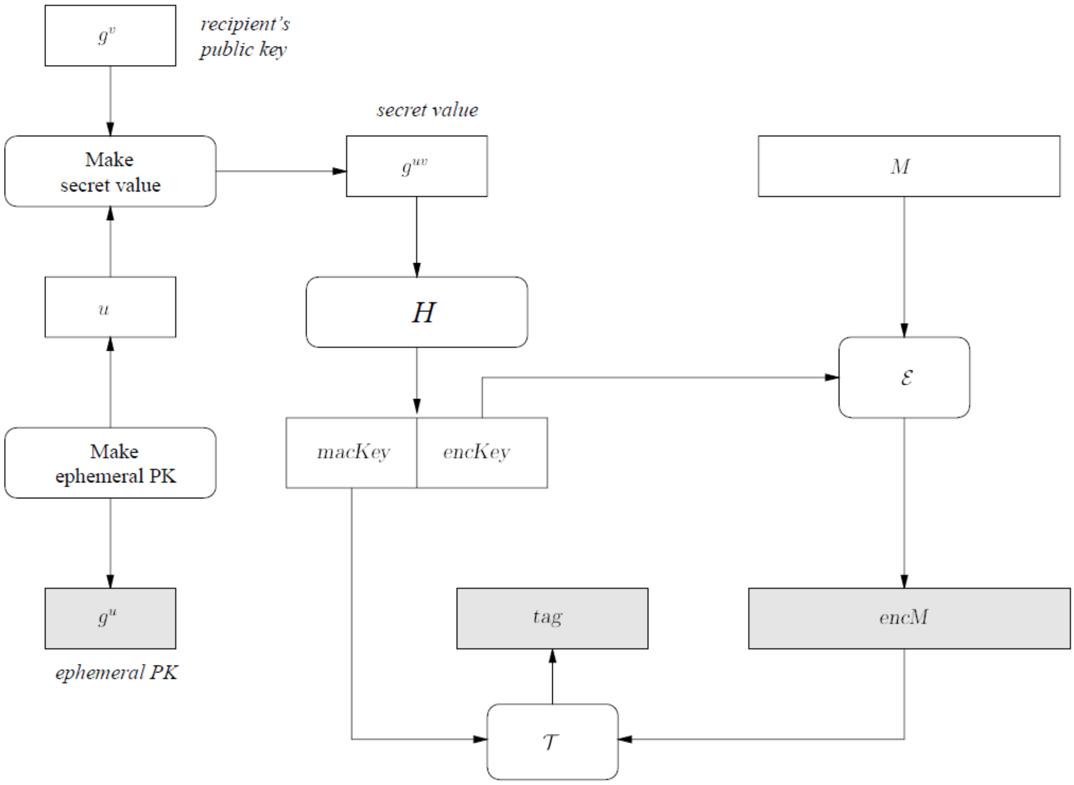

Figure 4: Encrypting with ECIES scheme: To encrypt a message M, we choose a random u and compute an ephemeral public key g^u (or uG in additive form). The value g^u is part of the cipher-text (grey rectangles), consisting of the ephemeral public key g^u, the authentication tag *tag* and the symmetrically encrypted message *encM*. This makes g^{uv} to an implicit Diffie-Hellman key exchange: both sender and receiver are able to compute this secret value. The value g^{uv} is passed to a key derivation function (hash function H). We obtain two symmetric keys, one for the symmetric encryption algorithm $\mathcal{E}$ and one for the MAC scheme $\mathcal{T}$.

b) Asymmetric encryption using ECIES is used for key transport of a symmetric key, thus, in general no mass data will be transfered using this method. With RSA PKCS#1 v1.5 encryption this would be a cryptographic requirement (do not encrypt large blocks of data with this method). The hybrid scheme ECIES would, in principle, allow to encrypt large blocks of data. So, although possible in principle with ECIES, but not with `RSAES-PKCS1-V1_5`, to encrypt and transfer the payload C2X data, in the IEEE 1609.2 setting, ECIES will be used only to transport a symmetric key. Afterward, the purely symmetric scheme authenticated encryption is used. The reason for this construction is that repeated transfers with authenticated encryption using a key transferred in the first exchange is much faster than repeated hybrid encryption using ECIES.

c) Authenticated encryption is used for efficient, integrity protected and, optional, encrypted transfer with least overhead. It will be used for example for unicast transfers of software downloads.

5.3 Current discussions in standardization groups

We would like to shed some light on current discussions in C2C CC Security WG and ETSI. In one of the past meetings, the question on patents has been raised. Some members were in favor of the alleged patent-free ECGDSA alternative (G as German or General) instead of ECDSA. However, it is improbable that the algorithm will be changed to ECGDSA for the European ETSI standard. Mainly because the patents are often on certain implementation tricks rather than the algorithm itself.

A further issue, that the author of this paper raised in C2C CC Security WG, is the usage of the special

NIST curves in the American IEEE 1609.2 standard. The author opts very strongly for the so-called Brainpool curves, see [19, 20], instead of NIST curves. The Brainpool curves have some security advantages compared with NIST curves. NIST curves are more vulnerable to certain attacks (fault attacks, side channel attacks) than twist-secure Brainpool curves.

The exploitation of the Pseudo-Mersenne structure in implementations gives a small advantage in performance when using special hardware or software multipliers. However, with general long-number arithmetic units as the Crypto2000, see below, general primes are equally fast.

The usage of Brainpool curves is an open issue within C2X standardization. At least, general curves should be allowed, i.e., the implementations should not be restricted to the Pseudo-Mersenne structure of the NIST primes.

6 Implementation options of IEEE 1609.2 and consequences

As shown in Section 4, Car2X communication has rather high performance requirements. In principle, we see the following architectural possibilities

1. pure software implementation on automotive processors,
2. usage of standard smart cards,
3. usage of FPGAs or ASICs.

Of course, even combinations of these three could be implemented, e.g., automotive processors with special crypto co-processors or high-speed cryptographic co-processor cards from servers.

6.1 Pure software implementation on automotive processor

An implementation of IEEE 1609.2 in software only seems to be rather unrealistic due to performance constraints. To give a rough idea on the achievable performance and the security overhead we refer to Table 4.

A car usually signs with 224 bit key length. With a typical payload of 32 bytes, the size of the signed message (together with geodata) is 251 bytes, the size of an OBU signing certificate is 125 bytes. A RSS which signs and encrypts with 256 bit has a certificate length of 180 bytes, a CA signing certificate has 135 bytes, see [23, Appendix C]

Please note, that a typical automotive embedded processor has a clock rate of 400 MHz, i.e., the numbers have to be scaled by a factor of six at least. In reality the factor should be even higher due to the architectural advantages of a Core i5.

In addition to the performance problem, current automotive processors have an insufficient protection against malicious manipulations.

6.2 Smart cards

Smart cards are an alternative to pure software implementations. There are at least two reasons: Most smart cards have a long number arithmetic unit which accelerates public-key operations as for ECC considerably. In addition, the tamper protection of smart cards is designed for attackers with high attack potential, thus is sufficient for automotive use cases. In the past, smart cards were not qualified for automotive environmental conditions, but recently smart card companies began to produce special smart cards for use under automotive conditions, for example, Infineon SLI70 products with an extended temperature range or special SLM76 Machine to Machine cards.

In Table 5 we give the performance of two typical smart cards for ECC operations. Note that the results of the pure software implementation of Table 4 have been obtained with a much higher clock frequency than with smart cards in Table 5. Of course, the clock frequency of smart cards cannot be increased arbitrarily due to design constraints.

Differences in performance result, besides different clock frequencies and different cores (SLE88 – 32 bit processor, SLE78 – 16 bit processor), mainly from different cryptographic co-processors for performing modular multiplications. The Infineon SLE66 and some SLE88 possess ACE (Advanced Crypto Engine) with bit-length 1120 bit. The SLE88 considered here has its successor Crypto2000 or Crypto@1408/2304

Table 4: Performance[1] and security overhead for typical public key algorithms[2]

algorithm →	RSA			ECDSA[3] over $\mathbb{F}_p$			symmetric[5]
key length	512	1024	2048	160	224	256	AES-128
size of signature (bit)	512	1024	2048	320	448	512	64–96
security level (bit)	60	80	112	80	112	128	128
signature generation: time in ms	2.485	17.37	142.8	1.434	2.972	3.991	9.33 E-3
no. of signature generations, s^{-1}	402.3	57.55	7.001	697.5	336.4	250.5	107117
signature verification:[4] time in ms	0.076	0.252	0.956	1.676	3.673	5.201	8.46 E-3
no. of signature verifications, s^{-1}	13135	3959	1045.1	596.5	272.2	192.2	118137

[1] All timings have been performed on a Dell Latitude E4310 Notebook with Intel Core i5 520M@2.4 GHz processor with assembler optimized library MIRACL 5.44 under Windows XP SP3 and Cygwin, extra routines for signature generation / verification, no side-channel countermeasures

[2] Only most time-consuming functions, i.e., for RSA verification exponentiation with small exponent $m = s^e \bmod n$, for RSA signature generation full exponentiation $s = m^d \bmod n$, for ECDSA signature generation one scalar multiplication $Q = \alpha P$, for ECDSA signature verification two scalar multiplications (one multi-scalar exponentiation) $\alpha P + \beta Q$. This means no hashing and no padding is performed in the tests.

[3] ECDSA over NIST $\mathbb{F}_p$ curves

[4] signature verification for RSA with $e = F_4 = 2^{16} + 1$

[5] For comparison, we give the timing of the symmetric authenticated encryption scheme AES-GCM (similar to AES-CCM) with AES-128 to encrypt and decrypt 60 byte. Instead of a signature we have a MAC tag. Thus, signature generation here means encryption and tag generation, signature verification means decryption and tag verification.

to perform public key operations. Besides some minor architectural changes, Crypto@1408 has a sufficient number of registers for ECC, a so-called parallel mode and an adder of 1408 bit width (576 bit for ECC register which is sufficient). The successor of SLE88, SLE78, often uses the so-called Crypto@2304T (T as Tiny) which is a modified, cost optimized Crypto@2304. With Crypto@2304T, the parallel mode has been removed and the adder width is reduced, thus an addition has to use the adder for up to four times.

Both ECDSA implementation are secured against attacks, especially side-channel attacks. If costs play a minor role, a special unprotected multi-scalar multiplication for signature verification could be used instead of the standard protected scalar multiplication. This is done by the SLE78 implementation by IFX, therefore, verification is faster than generation.

Note that the design of a smart card long-number arithmetic unit follows completely different design goals (power consumption, area, costs) than that of a high-speed ECC arithmetic unit without resource constraints, see next subsection.

Table 5: Timing of ECDSA signatures over $\mathbb{F}_p$ with different smart card controllers for bit-length 256

	SLE88@66 MHz	SLE78@33 MHz
signature generation: time in ms	9	98
signature verification: time in ms	16	54

The implementation of the full IEEE 1609.2 standard on a smart card would be possible, but it would probably be as complex as, for example, Basic and Extended Access Control for RFID travel documents.

6.3 FPGAs and ASICs

Previous sections have shown that the performance required for Car2X communication probably cannot be done with pure software solutions. Even standard smart card controllers do not have enough performance. A promising alternative might be a special FPGA or ASIC design. Although costs for such FPGAs with high number of items are still way to high, they can be used as ASIC prototypes.

In this section, we investigate three different hardware implementations which have been designed for high performance ECC. Note, that usually high performance ECC designs are found for $\mathbb{F}_{2^m}$, not for $\mathbb{F}_p$!

The first design [14] has been implemented on a Virtex-4 FPGA from Xilinx and uses the special DSP units of this series heavily. With the move of the expensive modular long number multiplications to the 18×18 multipliers and a fast interconnect to the other arithmetic modules, the authors achieve a rather high performance for point multiplication on elliptic curves. The design works with different clocks for the DSP modules and the rest of the reconfigurable modules. This works because in this special FPGA chip, the DSP units are bare silicon and not reconfigurable modules.

Another point for high efficiency is due to the fact that [14] uses a special form of primes for the base field. They use the so-called NIST $\mathbb{F}_p$ curves, see [2]. NIST primes are the sum of a few powers of two, such that the binary representation has very few bits set, so-called Pseudo-Mersenne structure. Such a special structure of primes has been developed in the first place for efficient software implementations. Using such special primes is useful for high speed hardware, too.

When implementing this design as an ASIC, one would probably not directly map the DSPs on ASIC technology due to high area and costs. Instead, for a design targeted for ASICs only, one would probably choose a different modular multiplier.

The second design [13] is based on Residue Number Systems (RNS). To use RNS for arithmetic in $\mathbb{F}_p$, co-prime numbers $q_1, q_2, \ldots, q_m$ will be selected with $\prod_{i=1}^{m} q_i > p^2$. At the start of the computation all numbers will be reduced modulo q_i. The operations will be performed in parallel in rings $\mathbb{Z}_{q_i}$, at the end the main result is reconstructed using Chinese Remainder Theorem.

The main advantage of this method is its inherent parallelization of the computations in the smaller rings. Arithmetic in such smaller rings can be built much more efficiently with lower delays and higher clock frequencies. In addition, pipeline stages can be used.

The drawback of RNS is that all operands have to be reduced modulo p initially. Because of Chinese Remainder Theorem, the solution is only unique if the result is smaller than the product of all q_i. The initial reduction, transformation into RNS, is often very costly.

The author of [13] uses the fact, that operations on elliptic curves are of the form $AB + CD$. Thus, the modular reduction has to be performed only after adding the products, i.e., one reduction for two modular multiplications. Using this trick one achieves smaller area and higher performance through pipelining. The results in Table 6 are for a Stratix II FPGA of Altera.

The paper [18] does not present a new design, instead it focuses on a methodology for parallelization of different arithmetic modules for high-speed ECC. Due to the high performance requirements for Car2X communication this could be very useful.

Area and timing estimates for one scalar multiplication of the three implementations are shown in Table 6. The time for one signature generation is approximately the time of one scalar multiplication, the time of one signature verification is roughly the time of two scalar multiplications.

The Adaptive Logic Modules (ALMs) from Altera are equivalent to the slices from Xilinx. The results from [18] are originating from a simulation using 90 nm CMOS technology. The area of the design shown is the true silicon area.[5]

Unfortunately, the results are not directly comparable due to different technologies. But they show at least the principal feasibility of a hardware co-processor for Car2X communication.

7 Summary and discussion

Car2X communication will only be successful with a solid security system. Car2X security, on the other hand, will only be accepted by public if the privacy problems are solved.

[5] Note that a rather old Intel Pentium 4 processor in 90 nm technology has approx. $112\,\mathrm{mm}^2$!

Table 6: Area and timing of different hardware implementations for a scalar multiplication on elliptic curves in $\mathbb{F}_p$ for bit-length 256

Source	Area	Time in µs	Clock frequency in MHz
[14]	24574 Slices + 512 DSPs	40.49	375
[13]	9177 ALMs + 96 DSPs	680	157.2
[18]	1383 KGates or 5.5 mm^2	36.70	385

With the IEEE 1609.2 WAVE security trial-use standard we have a cryptographically solid base for a future secure Intelligent Transportation System. However, it depends on the manufacturers and suppliers to actually implement this standard in its full form. In European circles, mostly the signature part is being discussed, but this is only half of the story. Asymmetric encryption with ECIES and authentic encryption are required for efficiency reasons and for certain applications.

The proposed mechanisms in IEEE 1609.2 are State-of-the-Art cryptography. We have to admit that the cryptographic details are not easy for non-cryptographers and non-mathematicians. However, this standard gives the unique chance to implement standard cryptography right from the start, not based on Security by Obscurity. It is hoped by the author, that industry will actually implement this standard and will not implement again proprietary solutions (let's fix the bugs later).

The usage of pseudonyms is a first step for the privacy problem. However, it remains to be seen if this solution is sufficient. There is the danger, that privacy can be broken by means of additional information from other layers or sources. In addition, we have the usual legal requirements for de-anonymizing or decryption.

The performance requirements for Car2X security are rather high. There is no common of the shelf product that can deliver the required performance for scalar multiplications. However, with custom built hardware the performance problems can be solved.

Current high-speed ECC co-processors are either designed for F_{2^m} or for resource constrained smart cards. Only a few high-speed F_p co-processors exist (as FPGA). It remains to be seen if parallel architectures are of further help. In the end, the market will decide how much our security and privacy in intelligent transport systems may cost.

8 References

[1] ISO/IEC 18033-2-2006. *Information technology — Security techniques — Encryption algorithms — Part 2: Asymmetric ciphers.* ISO/IEC, 2006.

[2] FIPS 186-3. *Digital Signature Standard (DSS).* Federal Information Processing Standards Publication 186-3, 2009. June 2009. Available at `http://csrc.nist.gov/`.

[3] ETSI TS 102 637-1. *Intelligent Transport Systems (ITS); Vehicular Communication; Basic Set of Applications; Part 1: Functional Requirements*, 2010.

[4] ETSI TS 102 637-2. *Intelligent Transport Systems (ITS); Vehicular Communication; Basic Set of Applications; Part 2: Specification of Co-operative Awareness Basic Service*, 2010.

[5] ETSI TS 102 637-3. *Intelligent Transport Systems (ITS); Vehicular Communication; Basic Set of Applications; Part 3: Specification of Decentralized Environmental Notification Basic Service*, 2010.

[6] ETSI TR 102 893. *Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA)*, 2010.

[7] M. Abdalla, M. Bellare, and P. Rogaway. DHIES: An encryption scheme based on the Diffie-Hellman problem. In D. Naccache, editor, *Topics in Cryptology – CT-RSA 2001*, number 2020 in Lecture Notes in Computer Science, pages 143–158. Springer-Verlag, 2001.

[8] Holger Bock, Michael Braun, Markus Dichtl, Erwin Hess, Johann Heyszl, Walter Kargl, Helmut Koroschetz, Bernd Meyer, and Hermann Seuschek. A milestone towards RFID products offering asymmetric authentication based on elliptic curve cryptography. In *Proceedings of RFIDSec 2008, the 4th Workshop on RFID Security, Budapest, Hungary, July 2008*, 2008.

[9] Andrey Bogdanov. Linear slide attacks on the KeeLoq block cipher. In Dingyi Pei, Moti Yung, Dongdai Lin, and Chuankun Wu, editors, *Information Security and Cryptology, Third SKLOIS Conference, Inscrypt 2007, Xining, China, August 31 - September 5, 2007*, volume 4990 of *Lecture Notes in Computer Science*, pages 66–80. Springer, 2007.

[10] Morris Dworkin. Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality. NIST Special Publication 800-38c, National Institute of Standards and Technology (NIST), May 2004. See `http://csrc.nist.gov/publications/nistpubs/800-38c/sp800-38c.pdf`.

[11] Thomas Eisenbarth, Timo Kasper, Amir Moradi, Christof Paar, Mahmoud Salmasizadeh, and Mohammad T. Manzuri Shalmani. On the power of power analysis in the real world: A complete break of the KeeLoq code hopping scheme. In D. Wagner, editor, *Proceedings of CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 203–220. Springer-Verlag, 2008.

[12] I. Furgl and K. Lemke. A review of the digital tachograph system. In K. Lemke, C. Paar, and M. Wolf, editors, *Embedded Security in Cars*, pages 69–94. Springer, 2006.

[13] Nicolas Guillermin. A high speed coprocessor for elliptic curve scalar multiplications over F_p. In *Cryptographic Hardware and Embedded Systems (CHES 2010)*, number 6225 in Lecture Notes in Computer Science, pages 48–64. Springer, 2010.

[14] Tim Güneysu and Christof Paar. Ultra high performance ECC over NIST prime fields on commercial FPGAs. In *Cryptographic Hardware and Embedded Systems (CHES 2008)*, volume 5154 of *Lecture Notes in Computer Science*, pages 62–78. Springer, 2008.

[15] E. Hess, B. Meyer, and N. Janssen. Design of long integer arithmetic units for public-key algorithms. In *Proceedings of EUROSMART Security Conference, June 2000, 13–15*, pages 325–334, Marseille, France, 2000.

[16] S. Indesteege, N. Keller, O. Dunkelman, E. Biham, and B. Preneel. A practical attack on KeeLoq. In N. Smart, editor, *Proceedings of EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*. Springer-Verlag, 2008.

[17] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental security analysis of a modern automobile. In D. Evans and G. Vigna, editors, *Proceedings of IEEE Security and Privacy*, pages 447–462. IEEE Computer Society, 2010. see `http://www.autosec.org/pubs/cars-oakland2010.pdf`.

[18] Jyu-Yuan Lai, Shuo-Hung Chen, and Chih-Tsun Huang. Methodology of design space exploration for high-performance elliptic curve cryptographic processors. *IEEE Transactions on Very Large Scale Integration Systems*, 2010. to appear.

[19] Manfred Lochter and Johannes Merkle. Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation. Internet Request for Comment RFC 5639, Internet Engineering Task Force, March 2010. see `http://www.rfc-archive.org/getrfc.php?rfc=5639`.

[20] Johannes Merkle and Manfred Lochter. Ein neuer Standard für Elliptische Kurven. In BSI, editor, *Sichere Wege in der vernetzten Welt, 11. Deutscher IT-Sicherheitskongress des BSI*, pages 527–536. SecuMedia Verlag, 2009.

[21] Daniel Ohst. Security aspects of road tolling – requirements from a toll service provider. In *Proceedings of escar 2007, Embedded Security in Cars, Munich, November 6, 7, 2007*, 2007.

[22] IEEE P1363. *Standard Specifications for Public Key Cryptography*, 2000.

[23] IEEE P1609.2. *Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) – Security Services for Applications and Management Messages*, 2006.

[24] E. Schoch and F. Kargl. On the efficiency of secure beaconing in VANETs. In *3rd ACM Conference on Wireless Network Security (WiSec 2010), Proceedings*, March 2010.

[25] Hovav Shacham. Cars and voting machines: Embedded systems in the field, 2010. Invited talk at CHES 2010, Santa Barbara.

[26] ANSI X9.63-2001. *Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography.* American Bankers Association, 2001.