Automotive Security: Cryptography for Car2X Communication*

Torsten Schütze†
Rohde & Schwarz SIT GmbH
Stuttgart/Germany

March 1, 2011

*With special thanks to Jamshid Shokrollahi for many valuable discussions.
†The author worked until 09/2010 at Robert Bosch GmbH / Corporate Research on Car2X security standardization.

1. Introduction

History of Automotive Security and Cryptography

- started with Remote Key-less Entry and Key-less Go (mid of 1990s)
- immobilizers, tuning detection/protection, feature activation, etc.
- $\Longrightarrow$ **many proprietary systems, many successful attacks recently, for example,**
  - KeeLoq, Texas Instruments 40-bit encryption, NXP's HITAG2 encryption, etc.
  - after 16-year decline, car theft in Germany rose again in 2009
  - *Experimental security analysis of a modern automobile*, 2010, IEEE Security and Privacy + Hovav Shacham's invited talk at CHES 2010: *Cars and voting machines: Embedded systems in the field.*
- $\Longrightarrow$ **currently, attacks have only limited influence (one car, one brand); they do not scale**
  - only one security certified system (Digital tachograph)
  - automotive manufacturers + suppliers start to catch up

Car2Car and Car2Infrastructure communication

- information/data from the outside will directly influence your car, **attacks scale**
- $\Longrightarrow$ **Security for Car2X communication right from the start is essential**

Introduction — project landscape and standardization work

Selected projects

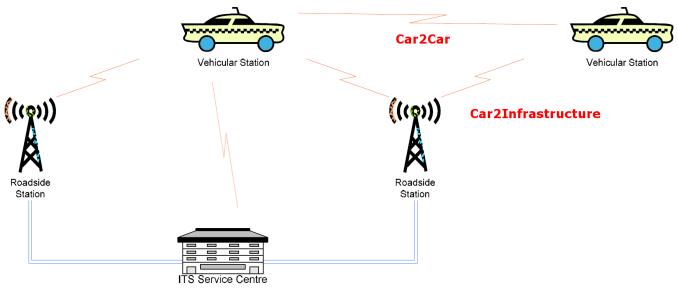
- SEVECOM (Secure Vehicular Communication, 2006–2008),
- EVITA (E-safety vehicle intrusion protected applications, 2008–2011), and
- simTD (Sichere Intelligente Mobilität Testfeld Deutschland – Safe and Intelligent Mobility Test Field Germany, 2008–2012).

Car2X security standardization (Europe)

- Car2Car Communication Consortium (C2C CC), esp. WG Security
  = industrial community by European automotive manufacturers + suppliers + research institutes, preparatory platform and discussion forum, industry
- eSafety Security Working Group of eSafety Forum / European Union,
  political and scientific support
- Working Group 5 Security of ETSI (European Telecommunications Standards Institute), Technical Committee (TC), Intelligent Transport Systems (ITS).
  = core standardization work

US standardization (IEEE 1609.2) is advanced concerning security mechanisms
Goal of this talk: show cryptographic mechanisms of IEEE 1609.2

2. Car2X communication — communication network



Communicating parties within a Car2X system

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,
- communication between car and network infrastructure.

Interesting: Car2Car + Car2Infrastructure = Car2X

- very short time and ultra low latency
- cannot (should not) rely on existing fixed infrastructure (router, access points)
- multi-hop routing using cars as relays
- $\Longrightarrow$ **Vehicular Ad Hoc Network (VANET)**

Car2X communication — communication protocol

1. mobile phone network, esp. Long Term Evolution (LTE) (purely symmetric)
   - existing infrastructure + broad coverage (?) $\Longrightarrow$ introduction easy, but
   - requires infrastructure, no real ad hoc network
   - latency only sufficient for comfort functions and traffic efficiency, not for passive / active safety
   - $\Longrightarrow$ considered in some European Car2X groups as phase 1, no real decision
2. IEEE 802.11p = Automotive WLAN = WLAN-p = Wireless Access in Vehicular Networks (WAVE)
   - amendment to WLAN 802.11a/b/g/n standard specifically for Automotive
   - "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"
   - C2C CC WGs consider *four latency categories*[1], i. e., time between triggering event (crash detection) and resulting action (brake or warning):
     a) $T \leq 50\,\mathrm{ms}$, b) $T \leq 100\,\mathrm{ms}$, c) $T \leq 200\,\mathrm{ms}$, d) best effort

[1] Time includes communication stack down and up plus air transmission!

Car2X communication — communication protocol — IEEE 802.11p

Technical Specs:

- frequency band 5.9 GHz (reserved bands in US and Europe): G5A – Road safety, G5B – Non-safety applications, G5C – shared with other applications
- communication range: short range, line-of-sight distances of less than 1000 m, high-speed: up to 27 Mbit/s

Major differences to core 802.11

- no Medium Access Control (MAC) multi-cast on the Control Channel
- no MAC fragmentation
- Distributed Congestion Control, i.e., 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
- no MAC encryption (as WPA & WPA2)
- **IEEE 802.11p does not use 802.11i security!**

So, no security at all? No!
IEEE 1609.2 (WAVE Security) = dedicated security standard for 802.11p

Car2X communication — application requirements

ETSI defined **basic set of applications (BSA)**, e.g., stationary vehicle warning and signal violation warning

Two important message types

- **Cooperative Awareness Message (CAM)**: periodically broadcast information about station to other nodes in vicinity (beaconing), will not be routed
  - time stamp, car or RSS specific fixed data (identity); position, velocity, direction; special flags to signal braking activities, etc.
  - length of CAM: approx. 64 bytes; sent periodically in message broadcast mode
- **Decentralized Environmental Notification Message (DENM)**: generated upon detecting an event and contains information about this event; addressed via geo-positioning
  - geographical data: event, target, relevance, and networking area
  - application information: event management information, priority, generation time, validity period, etc.
  - event information: type, severity, reliability, etc.
  - action ID = unique for each event, contains info about originator
  - DENM much longer than CAMs, exact length depends on event type; transport: broadcast or unicast mode

Car2X communication — application requirements II

Every Car2X station builds **Local Dynamic Map** based on CAM, DENM, etc. (sensors)

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

3. Security objectives for Car2X communication, ETSI TVRA

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
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
Av1	Access to and the operation of ITS services by authorized users should not be prevented by malicious activity within the ITS-S environment
Ac1	It should be possible to audit all changes to security parameters and applications (updates, additions and deletions)
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

Security objectives — summary

- **only authorized users** should be able to participate in the system
- **anonymity** or **non-traceability** of users (pseudonym's) **Contradiction?**
- security of messages:
  - protect **integrity of sender** and **of data**; **freshness**
  - **confidentiality** of some (few) data
- **non-repudiation** of C2X data
  - discussed very controversial within C2X CC Security WG
  - only a very few applications, e.g., eCommerce applications with monetary value, require it
  - strictly speaking, non-repudiation of data is not required for most use cases
  - but: selected solution (digital signatures) gives it for free

$\Longrightarrow$ **standard security objectives**

- can be achieved by different cryptographic means, but none of mechanisms is optimal for all use cases
- problem: protection of many small packets with minor overhead but long-term security of the system

security mechanism → / ↓ security goal	symmetric	asymmetric	hybrid
integrity of data	Message Authentication Code (MAC) over data	digital signature over data	MAC over data
integrity of sender			
non-repudiation	–		digital signature
timeliness / freshness	MAC over data and time variant parameters	digital signature over data and time variant parameters	MAC over data and time variant parameters
confidentiality (optional)	encryption	encryption of keys and small data only	encryption / decryption of ephemeral symmetric key with public / private key, encryption of bulk data with symmetric key
unicast communication	+	+	+
broadcast communication	−	+	?
performance	very fast	slow – very slow	good
size of MAC / signature (bit)	64–96	ECC: 448–512 | RSA: 512–2048	64–96 MAC, 448–512–2048 signature
applications / standards / examples	GSM, UMTS, banking	IEEE 1609.2 WAVE Security, C2X CC Security WG, SEVECOM architecture	eMail security (OpenPGP, S/MIME), IEEE 1609.2 + TESLA

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

4. Possible security mechanisms — discussion

- **Asymmetric methods:** Digital signatures fulfill all requirements, especially in Ad Hoc networks, but not efficient enough as exclusive mechanism
  - RSA much too slow for long-term security
  - ECC possible, 256 bit sufficient for medium/long-term security
  - biggest drawback: size of security payload, performance for verification
- **Symmetric methods:** Message Authentication Codes or Authenticated encryption
  - much smaller footprint in security payload and performance
  - but: not suitable for highly dynamic MANET scenario
- **Hybrid methods:** combines advantages of symmetric and asymmetric schemes
  - medium size footprint, more complicated scheme

What does this mean for IEEE 1609.2? $\Longrightarrow$ **use all three mechanisms!**

Possible security mechanisms — performance and privacy

Performance requirements

- most crucial for DSA-like methods: signature verification
- E. Schoch/F. Kargl [2010]: *"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**."*
- Chr. Paar/escrypt: **up to 5000 signature verifications per second**
- worst case load: 8000 signatures per second + congestion control for CAMs on network layer[2] + cong. control on application layer $\Longrightarrow$ **4000 signatures per second**
  - FPGA box by Escrypt Inc.: up to 400 sig. verifications per second
  - simTD assumes 40 signature verifications per second

Privacy requirements

- highly political topic which is discussed very controversial in WGs $\rightarrow$ requires own talk
- current solution: use pseudonyms on layer three $\Longrightarrow$ This may not be enough!
- planned frequency of pseudonym changes currently higher in US than in Europe

[2] parameters not fixed yet

5. Cryptography of IEEE 1609.2

- IEEE 1609.2-2006 Trial Use Standard for Wireless Access in Vehicular Environments (WAVE) – Security Services for Applications and Management Messages
- = most advanced security standard for Car2X communication
- currently development of draft standard with rationale for mechanism selection
- 112 bit security for short-time messages, 128 bit security else
- based on modern ECC methods

Components of IEEE 1609.2

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

b) asymmetric encryption with ECC, specifically Elliptic Curve Integrated Encryption Scheme (ECIES)

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

Cryptography of IEEE 1609.2 — digital signature (ECDSA)

- **main mechanism:** every message will be signed with private key of sender, public key may refer to pseudonym → privacy; receiver verifies certificate and message → integrity of data and of sender, non-repudiation; time-stamp in message → freshness
- ECDSA NIST P-224 curve + SHA-224; ECDSA NIST P-256 curve + SHA-256

ECDSASign(message m, private key d, generator G, order n)

1. Compute $H(m)$ and convert 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 $(r, s) \in F_n \times F_n$.

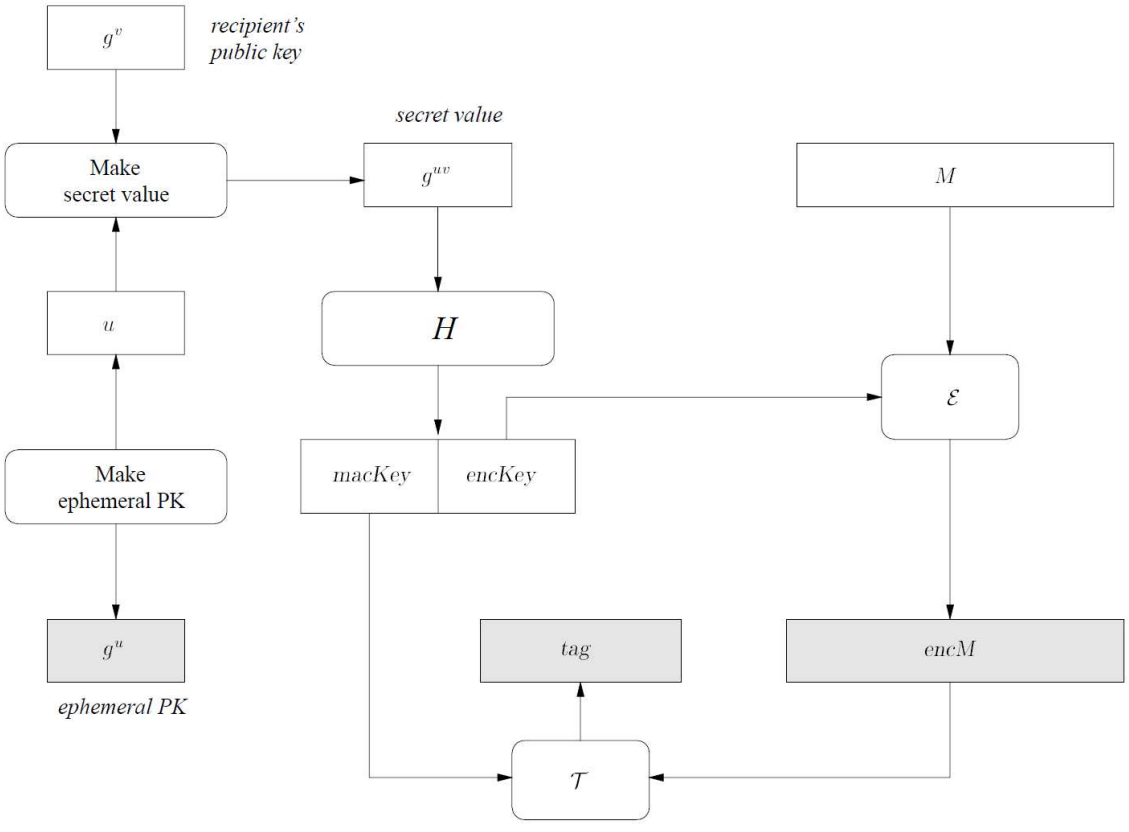
ECDSAverify(signature (r, s), message m, generator G, order n, public key Q)

1. If $r \notin [1, n-1]$ or $s \notin [1, n-1]$, deny signature as invalid.
2. Compute $H(m)$ and convert 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 $\nu = r$.

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 $\pi(P) = \pi(x_P, y_P) = x_P$.

Cryptography of IEEE 1609.2 — asymmetric encryption (ECIES)

- ECIES with NIST P-256 curve, used for key transport of symmetric keys (for authenticated encryption)
- `RSAES-PKCS1-V1_5` would not allow to encrypt and transfer (huge) payload C2X data
- hybrid method ECIES would allow, in principle, to transfer huge data, but is used only for keys (efficiency)
- Encrypting with ECIES, see next slide:
  - select random u and compute ephemeral public key g^u (or uG in additive form)
  - cipher-text (grey rectangles): ephemeral public key g^u, authentication tag *tag* and symmetrically encrypted message *encM*
  - Use g^v (public key of receiver) to compute g^{uv} = implicit Diffie-Hellman key exchange
  - key derivation with hash function H for $g^{uv} \rightarrow$ encryption key, MAC key
  - compute MAC tag *tag* and encrypted message *encM*



ECIES encryption.

Cryptography of IEEE 1609.2 — authenticated encryption

- AES-CCM = AES-128 CTR mode + AES-128 CBC-MAC
- used for efficient integrity protected and optional encrypted mass data transport, e.g., software download with unicast

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 $m =$ (Clen − 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. $P = \mathrm{MSB}_{\mathrm{Clen}-\mathrm{Tlen}}(C) \oplus \mathrm{MSB}_{\mathrm{Clen}-\mathrm{Tlen}}(S)$
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

AES-CCM with AES-128, a nonce length of 96 bit and a tag length of 128 bit.

6. Implementation options of IEEE 1609.2

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

1. Pure software implementation on automotive processors

- see Table on next slide to get timing estimates for all algorithms
- simTD uses RSA-512 bit (!!) with PKCS# 1 in software
  - **totally insecure**, 768 bit broken in 2010
  - **atypical**, RSA verify = very fast,
    DSA verify = slower than DSA generation (in general)
  - realistic RSA signature generation would be totally impossible!
- Don't compare apples to peaches![3]

[3] simTD: RSA-512 verify with ECDSA 256 bit generate

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] Dell Latitude E4310 Notebook, Intel Core i5 520M@2.4 GHz processor, assembler optimized library MIRACL 5.44, Windows XP SP3 + Cygwin, extra routines for signature generation/verification, no side-channel countermeasure

[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) $\alpha P + \beta Q$. No hashing, no padding.

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

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

[5] timing of AES-GCM (instead of AES-CCM) to encrypt/decrypt 60 byte. MAC tag instead of signature.

Implementation options of IEEE 1609.2 — 2. Smart cards

- dedicated security devices with special arithmetic unit (public key coprocessor)
- Here: Infineon smartcards with coprocessor Crypto@2000 (successor of ACE)
- SLE88: 32-bit processor with Crypto@1408 bit coprocessor, parallel mode
- SLE78: 16-bit processor with Crypto@2304T, T as tiny, no parallel mode
- ECDSA implementations:
  - SLE88 by Siemens CT IC3 for BSI
  - SLE78 by IFX as standard ECC-Library
  - both implementations secured against attacks

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

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

Note: Timings for verification and generation with SLE78 swapped w.r.t. paper. IFX uses fast, unsecured, multi-scalar multiplication for verification, thus, it is faster than generation.

Implementation options of IEEE 1609.2 — 2. Smart cards

- dedicated security devices with special arithmetic unit (public key coprocessor)
- Here: Infineon smartcards with coprocessor Crypto@2000 (successor of ACE)
- SLE88: 32-bit processor with Crypto@1408 bit coprocessor, parallel mode
- SLE78: 16-bit processor with Crypto@2304T, T as tiny, no parallel mode
- ECDSA implementations:
  - SLE88 by Siemens CT IC3 for BSI
  - SLE78 by IFX as standard ECC-Library
  - both implementations secured against attacks

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

	SLE88@66 MHz	SLE78@33 MHz
signature generation: time [ms]	9	gap: factor 64 for SLE88@66 MHz
signature verification: time [ms]	16	

Note: Timings for verification and generation with SLE78 swapped w.r.t. paper. IFX uses fast, unsecured, multi-scalar multiplication for verification, thus, it is faster than generation.

Implementation options of IEEE 1609.2 — 3. FPGAs and ASICs

- most high-performance ECC designs for F_{2^m}, not for F_p
- smart card designs usually optimized with respect to area and power consumption, not performance

High-performance FPGA designs for scalar multiplication on ECC over 256-bit $\mathbb{F}_p$

Source	Area	Time [µs]	Clock frequency [MHz]
Gueneysu/Paar [2008]	24574 Slices + 512 DSPs	40.49	375
Guillermin [2010]	9177 ALMs + 96 DSPs	680	157.2
Lai/Chen/Huang [2010]	1383 KGates or 5.5 mm^2	36.70	385

FPGA solution **feasible**

Implementation options of IEEE 1609.2 — 3. FPGAs and ASICs

- most high-performance ECC designs for F_{2^m}, not for F_p
- smart card designs usually optimized with respect to area and power consumption, not performance

High-performance FPGA designs for scalar multiplication on ECC over 256-bit $\mathbb{F}_p$

Source	Area	Time [µs]	Clock frequency [MHz]
Gueneysu/Paar [2008]	24574 Slices + 512 DSPs	40.49	gap: closed
Guillermin [2010]	9177 ALMs + 96 DSPs	680	157.2
Lai/Chen/Huang [2010]	1383 KGates or $5.5\,\mathrm{mm}^2$	36.70	385

FPGA solution **feasible**, **but**:

- much too expensive for mass production,
- heat dissipation problems,
- Gueneysu/Paar [2008] only for NIST curves, no Brainpool curves

7. Summary

Without sound security architecture, there will be no successful Car2X communication. Without sound privacy mechanisms, there will be no sound security architecture and acceptance by customers.

Conclusions

- IEEE 1609.2 = sound security mechanisms for Car2X
- No Common-Off-The-Shelf hardware solution available yet, but (expensive) FPGA solutions exists
- Privacy issues (except pseudonymity) not completely solved
- It is to be hoped that automotive manufacturers and suppliers implement IEEE 1609.2 secure and complete, no weakened or proprietary solution.
- Hopefully, no manufacturer/supplier starts with weak/no security and tries to fix the bugs later.