



State of the art of public key infrastructures for cooperative ITS

Deliverable 2.4.4.4

Activity 2: Studies

Sub-activity 2.4 > Specifications

Version 2.00

Publication date: 12/05/2017



Co-financed by the Connecting Europe Facility of the European Union

The contents of this publication are the sole responsibility of the SCOOP@F project consortium and do not necessarily reflect the opinion of the European Union.

Information about the document

Document: State of the art of public key infrastructures for cooperative ITS

Date of publication: 12/05/2017

Responsible, Entity: Houda LABIOD, Telecom ParisTech

Status: Version 2.00 – Approved

Publication history

Date	Version	Redactors	Principal modifications	Dissemination
16/10/2015	1.00	Houda LABIOD A.Sehrouchni P.Urien		Renault (ISE)
12/05/2017	2.00	Houda LABIOD A.Sehrouchni P.Urien	Brigitte LONC's modifications	Scoop@F Renault (ISE) Release 2

Reference to the version administration

Version number to be composed of 3 digits > vR.XY

- **R** corresponds to the release number : it is upgraded each time SC Studies validates the diffusion of a new release,
 - **X** is the major version number: it is upgraded each time SC Studies validates the deliverable,
 - **Y** is the minor version number: it is upgraded each time a contributor changes anything.
- Once the deliverable is approved, its version number is upgraded from vR.XY to vR.(X+1)0
Once the deliverable is release, its version number is upgraded from vR.XY to v(R+1).00

As illustration :

- 0.03 > Work in progress version
- 0.10 > Del. Approved by SC Studies but not released
- 2.00 > Del. approved & released (in release 2)
- 2.05 > Del. Updated - in progress version

Table of Contents

Abbreviations	5
1. Objective.....	7
2. Cryptographic Mechanisms	7
2.1 Encryption Algorithm	7
2.2 Digital signature.....	9
2.3 Certificates and Authentication	10
3. Cooperative ITS PKI architectures: State of the art	11
3.1 IEEE 1609.2 v2 architecture [7] [8]	11
3.2 ETSI architecture.....	13
3.3 Car 2 Car Communication Consortium architecture [13]	18
3.4 Vehicle-to-Vehicle Security Credential Management System [14].....	19
3.5 Security framework in Japan [15]	25
4. Certificates formats.....	26
4.1 IEEE 1609.2 [7]	26
4.2 ETSI certificates	28
5. Security profiles according to ETSI 103 097 standard [12]	30
5.1 Security profile for CAM.....	30
5.2 Security profile for DENM	33
Conclusion	35

Table of figures

Figure 1: Symmetric encryption mechanism [2]	7
Figure 2: Public key data encryption and decryption [4]	8
Figure 3: Creation and verification a digital signature [4]	9
Figure 4: IEEE 1609.2v2 PKI architecture	12
Figure 5: Mapping of OSI modelling layers to the ITS architectural layers	14
Figure 6: ETSI ITS PKI architecture	17
Figure 7: C2C-CC PKI structure	18
Figure 8: Simplified V2V security system	20
Figure 9: Current V2V security system design for deployment and operations	23
Figure 10: CA concept in Japan	25
Figure 11: IEEE 1609.2 Explicit and Implicit certificates	26
Figure 12: IEEE 1609.2 Certificate format	27
Figure 13: ETSI Certificate format	28

Table of tables

Table 1: 1609.2 v2 certificates	13
Table 2: Comparison between ETSI and C2C-CC types of Cas	18
Table 3: Example of the ECDSA signature generation for a SecuredMessage	32
Table 4: An example signed header for CAM	33
Table 5: Fields that shall be included in SecuredMessage structure for DENMs	34

Abbreviations

For the purpose of the present document, the following abbreviations apply:

AA	Authorization Authority
BSA	Basic Set of Applications
BSM	Basic Safety Message
C2C CC	Car 2 Car Communication Consortium
CA	Certificate Authority
CAM	Cooperative Awareness Message
CAMP	Crash Avoidance Metrics Partnership
CIA	Confidentiality, Integrity and Availability
CME	Certificate Management Entities
CRL	Certificate Revocation List
DCA	Device Configuration Manager
DENM	Decentralized Environmental Notification Message
DOT	Department of Transportation
EA	Enrollment Authority
ECA	Enrollment Certificate Authority
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
ETSI	European Telecommunications Standards Institute
ICS	ITS Central Station
IEEE	Institute of Electrical and Electronic Engineers
IRS	ITS Roadside Station
ITS G5A	ITS 5,9 GHz communications
ITS	Intelligent Transport System
ITS-AID	ITS Application ID
ITS-S	ITS Station
IVS	ITS Vehicle Station
LA	Linkage Authority

LOP	Location Obscurer Proxy
LTC	Long Term Certificate
LTCA	Long Term Certificate Authority
MA	Misbehavior Authority
NHTSA	National Highway Traffic Safety Administration
NISTP	National Institute of Standards and Technology
OBE	On-Board Unit
OSI	Open System Interconnect
PC	Pseudonym Certificate
PCA	Pseudonym Certificate Authority
PKI	Public Key Infrastructure
RA	Registration Authority
RCA	Root Certificate Authority
RFI	Request for Information
RSU	Roadside unit
Rx	Reception
SCMS	Security Credential Management System
SDE	Secure Data Exchange
SDEE	Secure Data Exchange Entity
SHA256	Secure Hash Algorithm with 256 bits hash value (digest)
SSP	Service Specific Permissions
TVRA	Threat, Vulnerability Risk Analysis
Tx	Transmission
V2I	Vehicle to Infrastructure Communication
V2V	Vehicle-to-Vehicle Communication
V2X	Vehicle to X Communication
VIIC	Vehicle Infrastructure Integration Consortium
WAVE	Wireless Access in Vehicular Environment
WSA	WAVE Service Advertisements

1. Objective

The main goal of this document is to present the state of the art of the Public Key Infrastructure (PKI) architectures proposed for cooperative Intelligent Transportation System (ITS), described in ETSI, IEEE 1609.2, C2C Communication Consortium, and in US Vehicle-to-Vehicle Security Credential Management System and in Japanese security framework.

2. Cryptographic Mechanisms

Before giving details about supported security mechanisms, we provide in this section definitions about cryptographic mechanisms and its related data structures.

2.1 Encryption Algorithm

Encryption is the process of encoding messages in such a way that only authorized parties can read it. There are two types of encryption: symmetric encryption and asymmetric encryption.

2.1.1 Symmetric-key Algorithm

Symmetric-key algorithms are a class of algorithms for cryptography that use the same cryptographic keys for both encryption of plaintext and decryption of ciphertext.

The keys may be identical or there may be a simple transformation between the two keys. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link [1].

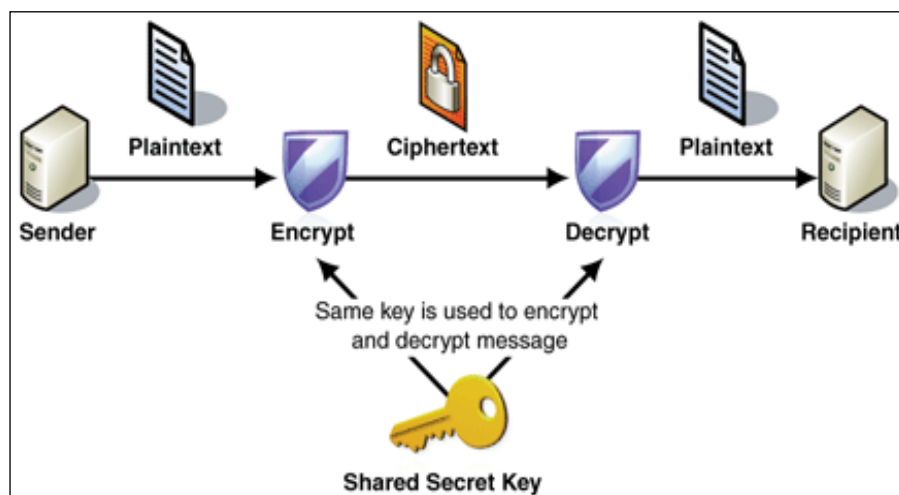


Figure 1: Symmetric encryption mechanism [2]

There are different types of symmetric-key encryption mechanisms that can be used:

- Stream ciphers encrypt the digits (typically bytes) of a message one at a time.
- Block ciphers take a number of bits (64 bits, 128 bits ...) and encrypt them as a single unit, padding the plaintext so that a multiple of the block size is composed.

An example of symmetric key encryption system which can be used to provide secured vehicular network communications is the AES (Advanced Encryption Standard) in CCM mode (Counter Mode Block Chaining Message Authentication Code as specified in [3]). CCM mode combines two cryptographic operations: the CBC-MAC with the counter mode of encryption. These two operations are applied in an "authenticate-then-encrypt" manner, i.e. CBC-MAC is first computed on the message to obtain a digest; the message and the digest are then encrypted using counter mode.

2.1.2 Asymmetric Algorithm

Public key encryption, also known as asymmetric encryption, is based on a public/private key pair. The keys are mathematically linked, so that data encrypted with the public key can only be decrypted with the corresponding private key. With public key encryption, the sender converts the plaintext message into ciphertext by encrypting it with the public key. The message recipient converts the ciphertext back into the plaintext message by decrypting it with the corresponding private key. By using public key encryption, a message sender has assurance that only the recipient will be able to read the message [4].

Asymmetric algorithms are important because they can be used for transmitting encryption keys or other data securely even when the parties have no opportunity to agree on a secret key in private. Asymmetric algorithms are based on mathematical functions (integer factorization, discrete logarithm, and elliptic curve relationships) which guarantee that it is computationally infeasible to derive the private key from the public key. ECIES (Elliptic Curve Integrated Encryption Scheme) is an example of asymmetric key encryption system based on elliptic curve cryptography, which is used to encrypt vehicular network communications.

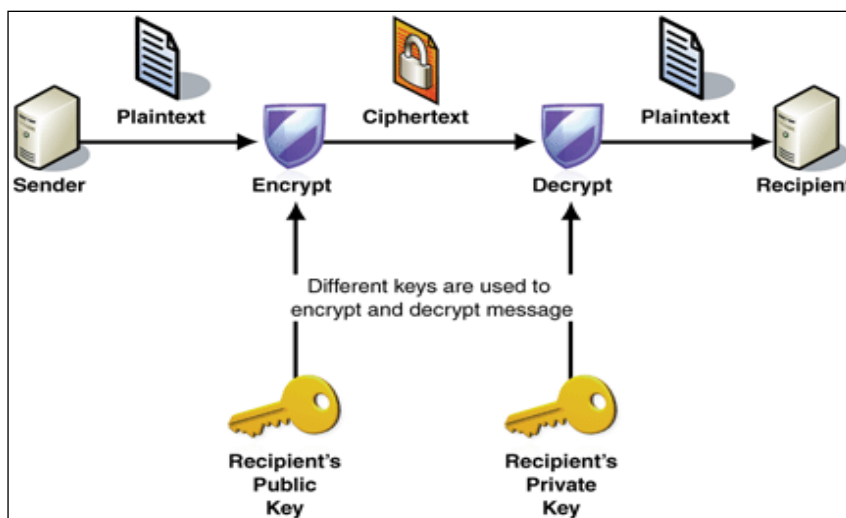


Figure 2: Public key data encryption and decryption [4]

2.2 Digital signature

A digital signature aims at binding message data of the sender to the sender's identity and to provide a means of verifying the integrity of the message to detect tampering. The digital signature ensures authenticity, integrity and non-repudiation of a digital message or a document.

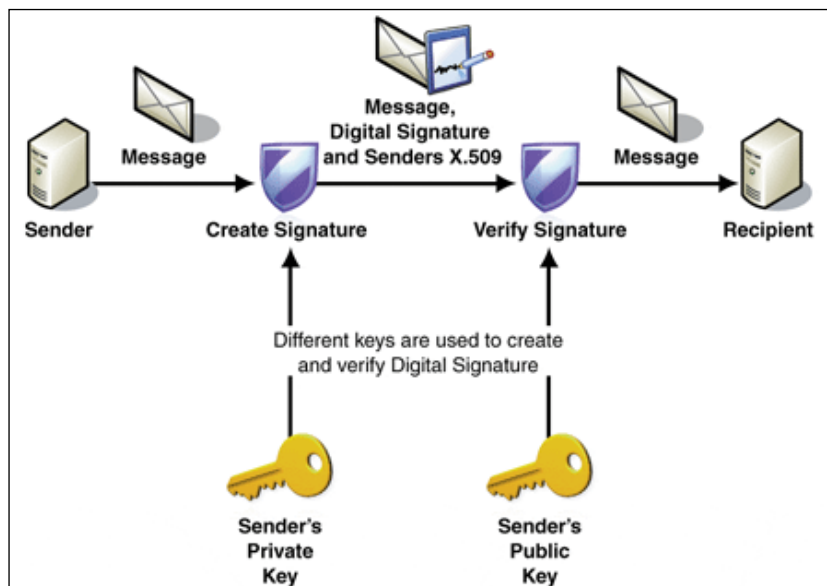


Figure 3: Creation and verification a digital signature [4]

In the figure above (figure 3), the private key of the message sender is used to create the digital signature. The corresponding public key (which is found in the sender's X.509 certificate) is used to verify the signature. Digital signatures are used to assure the message recipient that the message originated from the identified sender, and that the message contents have not been altered since they have been signed by the sender.

The public key can be distributed openly to encrypt messages and to verify digital signatures, but the private key in a key pair should be carefully guarded by its owner. This is necessary because it is used to prove the identity of the certificate subject and to decrypt messages that are intended for that subject [4]. ECDSA (Elliptic Curve Digital Signature Algorithm) is the algorithm used to provide digital signature in vehicular networks.

2.3 Certificates and Authentication

Authentication is the process of insuring that both entities of the communication are in fact who they say they are. It is the way to confirm the identity of an entity by means of digital signatures.

A certificate is used to identify an entity, in order to ensure that a certain public key is indeed from the expected source. A certificate is composed of three main parts, the owner's identification, the associated public key and the digital signature of an entity which has verified that the certificate's contents are correct. A specific certificate for vehicular communications is needed to avoid a huge overhead and protect users against potential tracking.

2.3.1 Public key certificates [4] [5] [6]

A public key certificate, also known as a digital certificate or identity certificate, is an electronic document used to prove the ownership of a public key. Generally, a certificate includes information about the key, information about the owner's identity, and the digital signature of the issuer of the certificate (the entity that has verified the certificate's contents are correct). If the signature is valid and the person examining the certificate trusts the signer, then he can use that key to communicate with its owner.

In a typical public-key infrastructure (PKI) scheme, the signer is a certificate authority (CA), usually a company such as VeriSign which charges customers to issue certificates for them. In a web of trust scheme, the signer is either the key's owner (a self-signed certificate) or other users ("endorsements") whom the person examining the certificate might know and trust.

The contents of a typical digital certificate are:

- Version number: The number version of the certificate,
- Serial Number: A unique identifier of the certificate,
- Subject: The person, or entity identified,
- Signature Algorithm: The algorithm used to create the signature,
- Signature: The actual signature to verify that the certificate belongs to the issuer,
- Issuer: The entity that verified the information and issued the certificate,
- Valid-From: The date the certificate is first valid from,
- Valid-To: The expiration date,
- Key-Usage: Purpose of the public key (e.g. encipherment, signature, ...),
- Public Key: The public key,
- Thumbprint Algorithm: The algorithm used to hash the public key certificate,
- Thumbprint (also known as fingerprint): The hash itself, used as an abbreviated form of the public key certificate.

3. Cooperative ITS PKI architectures: State of the art

The following section provides a brief description of the C-ITS PKI architectures.

3.1 IEEE 1609.2 v2 architecture [7] [8]

The IEEE 1609.2 standard specifies a set of security services for supporting vehicular communications. It defines secure message formats and processing for use by Wireless Access in Vehicular Environments (WAVE) devices, including methods to secure WAVE management messages and methods to secure application messages. It also describes administrative functions necessary to support the core security functions. The standard classifies all the entities that provide or use IEEE 1609.2 security services into two categories:

- Certificate authority entities (CA entities)
- End entities

CA entities issue certificates and Certificate Revocation Lists (CRLs). All other entities that use IEEE 1609.2 certificates, but cannot issue certificates or CRLs, are end entities. The IEEE 1609.2 defines two types of end entities: Secure Data Exchange Entity (SDEE) and secure provider service entity. It includes vehicles, roadside units (RSUs), application servers, and applications.

The IEEE 1609.2 standard defines three types of CA entities:

- **Root CAs:** Root CAs are trusted to issue certificates to all other CA entities and all end entities. The public keys of a Root CA are trusted by end entities. A Root CA issues certificates to other CA entities to authorize them to issue certificates or CRLs to end entities.
- **Secure Data Exchange CAs:** SDE_CAs issue certificates to end entities that send application messages secured with IEEE 1609.2.
- **WAVE Service Advertisements (WSA) CAs:** WSA_CAs issue certificates to end entities that send WSA. An end entity uses WSAs to broadcast what WSAs it provides.

The CRL Signers are CRLs distribution centers, which are entities that store and distribute certificates revocation lists (CRLs).

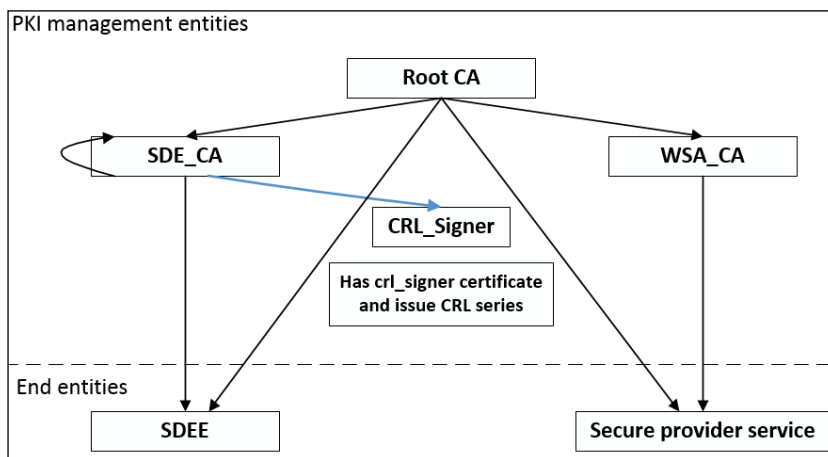


Figure 4: IEEE 1609.2v2 PKI architecture

A Root CA issues certificates to both CA and end entities within a defined region. This area is specified by the region field in the Root CA certificate and can indicate that the Root CA is worldwide.

A Secure Data Exchange CA (SDE_CA) is responsible for issuing certificates to SDEE and SDE_CA. The types of certificates that a SDE_CA is authorized to issue are:

- sde_ca,
- sde_enrolment,
- sde_identified_localized,
- sde_identified_not_localized,
- sde_anonymous
- crl_signer.

A SDEE can have three certificates types to secure its V2X communications:

- sde_identified_localized certificate,
- sde_identified_not_localized certificate, and
- sde_anonymous certificate.

These certificates are named communication certificates. The sde_enrolment certificate is used to request new certificates.

Wave Service Announcement CA (WSA_CA) is authorized to issue certificates for a secure provider service that broadcasts WSAs advertising a specific set of services.

The table below presents the different 1609.2v2 certificates.

1609.2v2 certificates of end entities		
	Communication certificates	Enrolment certificates
SDEE	sde_identified_localized, sde_identified_not_localized, sde_anonymous	sde_enrolment
Secure provider service	wsa	wsa_enrolment

Table 1: 1609.2 v2 certificates

For user privacy protection, the IEEE 1609.2v2 standard defines anonymous certificates issued by Root CA or SDE_CA to a SDEE. The IEEE 1609.2v2 anonymous certificates are communication certificates without the identifying information.

More details can be found in [8].

3.2 ETSI architecture

The ETSI ITS Technical Committee Working Group 5 is responsible for the ITS security architecture, providing security standards as also guidance on the use of security standards to protect and secure the ITS applications.

3.2.1 ETSI TS 102 940 [9]

ETSI TS 102 940 standard specifies a security architecture for ITS communications. It identifies:

1. Functional entities required to support security in an ITS environment.
2. Relationships that exist between the entities themselves and the elements of the ITS reference architecture.
3. Roles and locations of a range of security services for the protection of transmitted information and the management of essential security parameters. These include identifier and certificate management, PKI processes and interfaces as well as basic policies and guidelines for trust establishment.

Firstly, the standard discusses the ITS reference architecture which is based upon 4 processing layers identified as follows:

- Access Layer
- Networking Layer & Transport Layer
- Facilities Layer
- Application Layer

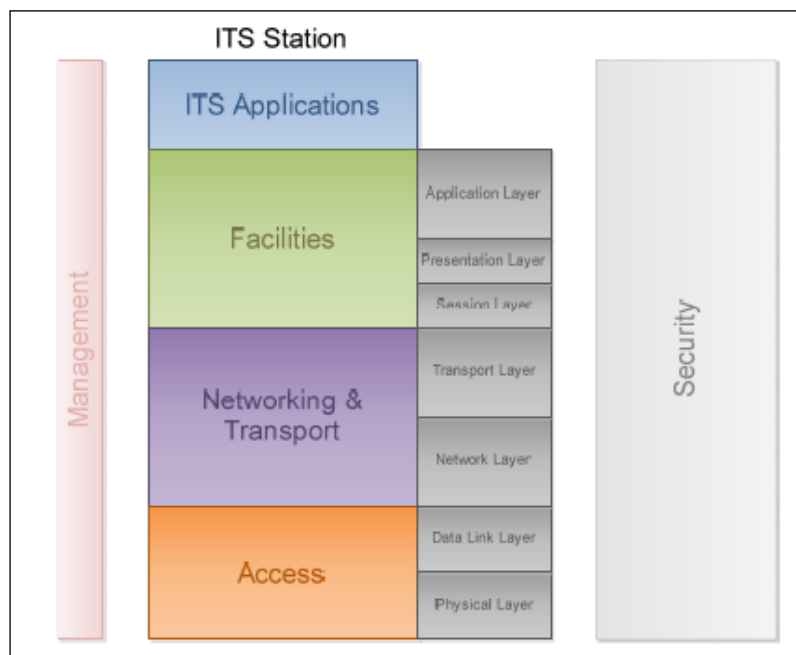


Figure 5: Mapping of OSI modelling layers to the ITS architectural layers

For each layer of the ITS station architecture, Management services and Security services are associated.

The expected functionality of the ITS station architecture layers can be mapped to OSI model. For example, Facilities layer is mapped to Application layer, Presentation layer and Session layer of the OSI model, Networking and Transport layer is mapped to the Transport layer and Network layer of the OSI model, and finally Access layer is mapped to Data Link layer and Physical layer of the OSI model. Having mapped the OSI protocol layers to the ITS station architecture, can be extended into an ITS communications architecture in which the protocol layers communicate on a peer-to-peer basis.

Secondly, the document presents the basic set of ITS applications which are represented by groups according to the functionality provided. It also presents, the communication behavior (addressing, frequency, direction...) for each use case of the ITS applications.

Thirdly, in order to provide communications security between ITS station and other stations, a range of security services supported by the ITS station are presented. Different categories of security services are defined such as enrollment services, authorization services, integrity services, plausibility validation services...Security services are provided on a layer-by-layer basis, in the manner that each of the security services operates within one or several ITS architectural layers, or within the Security Management layer.

Communications security services require more than one element within their functional model. Principal elements are:

- **Enrolment Authority:** authenticates an ITS Station (ITS-S) and grants its access to ITS communications.
- **Authorization Authority:** provides an ITS-S with authoritative proof that it may use specific ITS services.
- **Sending ITS-S:** acquires rights to access ITS communications from Enrolment authority, negotiates rights to invoke ITS services from Authorization Authority, and sends single-hop and relayed broadcast messages.
- **Relaying ITS-S:** receives broadcast messages from the sending ITS-S and forwards them to the receiving ITS-S if required.
- **Receiving ITS-S:** receives broadcast messages from the sending or relaying ITS-S.

The document also presents ITS security reference points through which information are exchanged, the types of information carried across these security reference points (CAM, DENM, authorization parameters, request for permissions...), and security services supported by each security reference point.

Fourthly, the standard presents security management supported by ITS stations. It is necessary for an ITS-S to provide secure access to common resources such as services, information and protocols. These security requirements can be separated into two parts: external security and internal security. External security represents the security related to the behavior of the ITS-S as a communication end-point, while internal security represents the security related to the ITS-S as a processing platform and application host.

The document talks also about how ITS communication system relies on indirect trust relationships built using certification by trusted third parties such as the Enrolment Authority (EA). EA allows an ITS Station to be a part of the ITS communications by providing access control and permissions.

Finally, the standard explains how ITS communications should support trust, privacy, access control, and confidentiality regarding ITS stations.

- **Trust** is supported by provisioning ITS stations with certificates allowing it to assert their permission to use the ITS system and to use specific ITS services and applications.
- **Privacy** is supported by using pseudonyms that can be used in place of a more meaningful and traceable identifier.
- **Access Control** is assured by giving ITS stations cryptographically signed certificates from the Authorization Authority (AA), which allows it to use specific services, or send particular information.
- **Confidentiality** of transmitted information in a unicast communication is protected by the encryption of messages within an established security association.

3.2.2 ETSI TS 102 941 [10]

The ETSI TS 102 941 standard specifies the trust and privacy management for ITS communications. It identifies trust establishment and privacy management required to support security in ITS environment and the relationships that exist between the entities themselves and the elements of the ITS reference architecture. The document starts by presenting ITS authority hierarchy, which is a PKI composed of an Enrolment Authority, Authorization Authority and a Root CA, and used for distribution and maintenance of trust relationships between ITS stations and authorities or other ITS stations (see figure 6).

Enrolment Authority

The EA issues a proof of identity to authenticate the canonical identifier of the ITS-S by delivering an enrolment certificate. This proof of identity allows to not revealing the canonical identifier to a third party and may be used by the ITS-S to request authorization of services from an Authorization Authority.

Authorization Authority

Having received the enrolment credentials, the ITS-S requests its authorization certificate(s) from the AA. These certificates allow the ITS-S to have specific permissions. Separation of enrolment and authorization is an essential component of privacy management and provides protection against attacks on a user's privacy.

Root CA

It issues certificates to all other Certificate Authorities. It is the root of trust for all certificates within that hierarchy. All certificates immediately below the root certificate inherit the trustworthiness of the root certificate. In order to trust an incoming message, an ITS-S must have access at least to the root certificate at the summit of the hierarchy for the authorization certificate attached to the message.

Four key attributes related to privacy (anonymity, pseudonymity, unlinkability, and unobservability) are cited. According to the standard, privacy is provided in two dimensions: privacy of ITS registration and authorization signaling, and privacy of communications between ITS stations.

After these definitions, the standard discusses the trust and privacy management by presenting the ITS station security lifecycle that begins with the manufacture phase, and passes to the enrolment phase, authorization phase and maintenance phase. At the **Manufacture phase** multiple information elements shall be established in the ITS-S using a secure process such as canonical identifier, contact information for EA and AA (network address and public key certificate), the set of current known trusted EA and AA that an ITS station may use to initiate the enrolment process and trust communications from other ITS-S respectively, a public/private key pair for cryptographic purpose as well as other multiple information. At the **Enrolment phase**, ITS-S requests its enrolment certificate from the EA at the **Authorization phase**, having received the enrolment credentials, the ITS-S requests its authorization certificates from the AA. And finally, at the **Maintenance phase**, ITS-S will be informed with any changes in EA and AA lists (adding or removing). The description of contents of request and response messages is presented in the document.

At the end of the document, security associations and key management between ITS stations, during broadcast, multicast or unicast communications are discussed. For broadcast communications, messages do not require confidentiality; CAMs and DENMs are signed using authorization certificates. Whereas for multicast and unicast applications communications shall be encrypted, and key management is required.

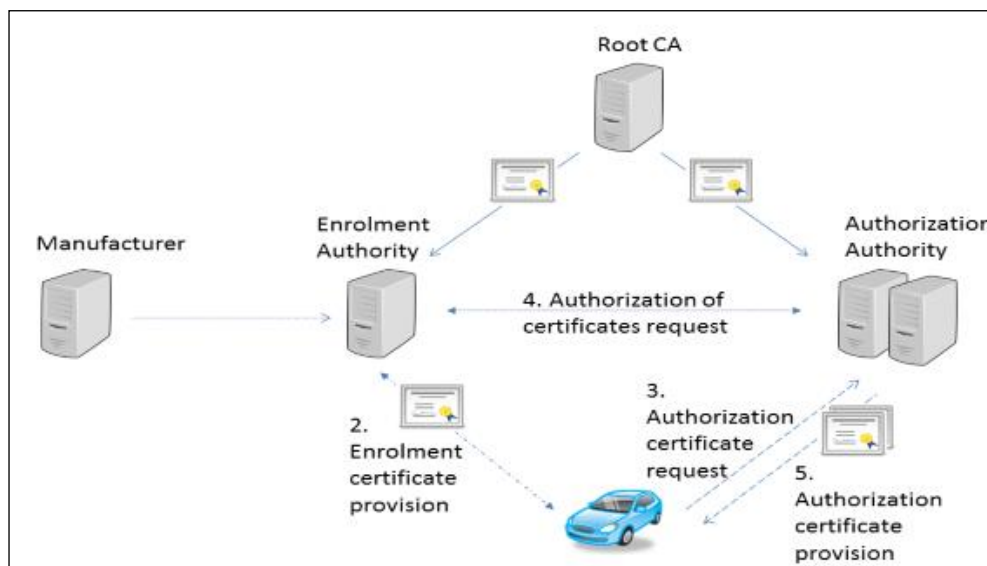


Figure 6: ETSI ITS PKI architecture

3.2.3 ETSI TS 102 731 [11]

ETSI defined a Threat, Vulnerability, Risk Analysis (TVRA) approach. TVRA consists of seven steps, where step 1 provides security objectives, step 2 provides security functional requirements, and TVRA step 7 provides detailed security requirements. TVRA step 4, 5 and 6 provide proof that links the detailed security requirements to the security requirements and security objectives. It contains argumentation for why the detailed security requirements are appropriate solutions to the objectives and functional requirements.

ETSI TS 102 731 standard provides descriptions of the security services and security architecture, but specifications in this document do not give deployment and implementation details. The document begins by describing the general ITS G5A security model, and presenting related security services for each countermeasure. These security services are divided into 2 level (First Level, and Lower Level). Security services identified as “First Level” are those that are invoked directly by applications or other components or layers in the ITS Basic Set of Application (BSA) [11]. Services identified as “Lower Level” are those that are invoked by other security services. The document mapped also countermeasures to CIA paradigm (Confidentiality, Integrity and Availability), and it represents ITS security services into 2 different groups: security service at transmission (Tx) and security service at reception (Rx). Then, an overview of the ITS security architecture is presented. It includes sending ITS Station, receiving ITS Station and the ITS Network. Connections, associations and interfaces between these 3 entities are also presented. After that, the document presents the ITS authoritative hierarchy composed

from the manufacturer, enrolment authority, and authorization authority. It gives also, the role of each of these entities, the different trust assumptions on which relies the security of ITS system, and ITS security parameter management such as identities and identifiers, and authorization and privacy with authorization tickets. The last part of the standard presents the ITS security services such as enrolment credentials, authorization tickets, security associations, single message services, integrity services, replay protection services, accountability services, plausibility validation, remote management, and report misbehaving ITS-S.

3.3 Car 2 Car Communication Consortium architecture [13]

The security working group of the C2C-CC defined the same PKI architecture as ETSI; however, names of ITS authorities are different.

ETSI types of CAs	C2C-CC types of CAs
Root Certificate Authority (RCA)	Root CA
Enrolment Authority (EA)	Long Term Certificate Authority (LTCA)
Authorization Authority (AA)	Pseudonym Certificate Authority (PCA)

Table 2: Comparison between ETSI and C2C-CC types of CAs

3.3.1 ITS authorities

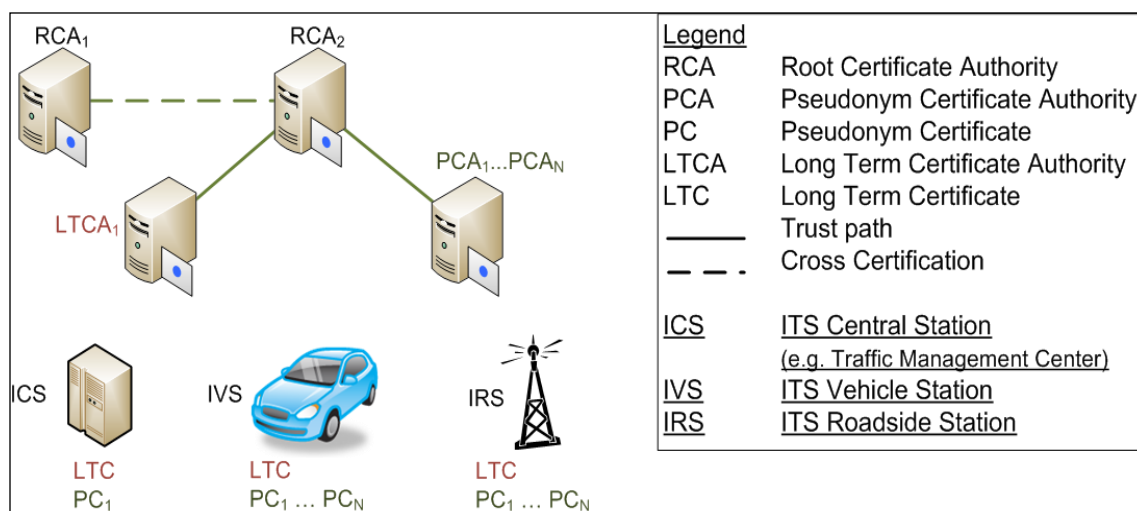


Figure 7: C2C-CC PKI structure

Root CA

The Root CA issues certificates for LTCA and PCA. It also defines and controls policies among all subordinate certificate issuers.

The Root CA is only required once a new LTCA or PCA shall be created, or when the lifetime of an LTCA or PCA certificate expires.

LTCA

The LTCA issues for each ITS-Station a Long-Term certificate that is valid for a long period. This Long-Term certificate is only used to identify and authenticate the ITS station (ITS-S) within the PKI, and never used in V2X communication for privacy reasons. It also enables ITS-S to request pseudonym certificates.

PCA

The PCA issues a short lifetime certificates called Pseudonym certificate, which are used in V2X communications. The PCA guarantees privacy of requesting ITS Stations since it is technically and operationally separated from the LTCA, which is the only authority that knows the real identity of the ITS-S.

3.4 Vehicle-to-Vehicle Security Credential Management System [14]

On the 15th October 2014, National Highway Traffic Safety Administration (NHTSA), Department of Transportation (DOT) published a Request for Information (RFI) named as Vehicle-to-Vehicle Security Credential Management System (V2V SCMS). The purpose of this RFI, is to seek responses concerning the establishment of an SCMS, security approaches for a V2V environment, and technical and organizational aspects of the SCMS.

In the following, we present a brief description of the V2V security system considered by NHTSA. According to the RFI, three primary elements of the V2V system requires security, which are:

- The V2V communication such as the medium, messages, data, certificates, and any other element that supports message exchange,
- V2V devices (cars),
- V2V security system itself through organizational, operational, and physical controls.

For this reason, different security technologies were assumed to be effective in providing trusted message exchange and secure communications. These technologies are: symmetric encryption, signature group, and PKI. Since it offers the most effective approach to achieving communications security and trusted messaging for a very large set of users in V2V system, asymmetric public key infrastructure (PKI) using the signature method, was selected by DOT and NHTSA, along with Crash Avoidance Metrics Partnership (CAMP) security experts.

3.4.1 V2V security design concept: functions, components, communications

The figure 8 presents a simplified V2V security system, with components and functions which are similar to the basic functions of any Public Key Infrastructure (PKI) security system.

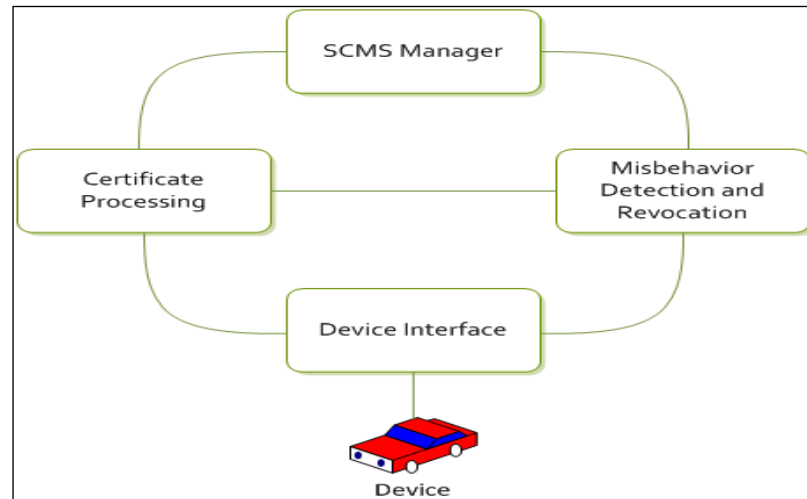


Figure 8: Simplified V2V security system

In figure 8, it is clear that the SCMS Manager is responsible for all other entities, and functions including certificate processing for devices, misbehavior detection and revocation of certificates. Figure 9 shows security, privacy operations and components used to accomplish the distribution of certificates to protect user's privacy.

As we see in the figure 9, entities of the V2V system are grouped into 4 classes:

- Overall Management,
- Registration and Enrollment,
- Certificate Management,
- Misbehavior Management.

SCMS is an integral part of V2V security design, it encompasses all technical, organizational, and operational aspects of the V2V security system that is needed to support trusted, safe /secure V2V communications and to protect driver privacy appropriately. Fundamental SCMS operating functions categories are:

1. Pseudonym functions,
2. Bootstrap functions.

1. Pseudonym functions/certificates

Since V2V communications relies on sending and receiving Basic Safety Messages (BSMs), short-term certificates become necessary to authenticate and validate these messages. A valid short-term certificate indicates that the BSM was transmitted from a valid and trusted source, in contrast a revoked certificate implies that the messages will be rejected by other V2V devices.

In order to create, manage, distribute, monitor and revoke short-term certificates, pseudonym functions were identified and defined as follow:

Intermediate Certificate Authority (Intermediate CA)

It is considered as an extension of the Root CA. Its main roles are:

- Authorize other CMEs and possibly Enrollment CA, using authority from the Root CA,
- Protect Root CA from direct access to the internet,
- Provide flexibility by removing needs to connect to RCA each time a new SCMS entity is added to the system.

However, Intermediate CA does not hold the same authority as the Root CA; it cannot self-sign a certificate.

Linkage Authority (LA)

Linkage values helps PCA calculating a certificate ID in a way to connect all short-term certificates from a specific device for ease of revocation in the event of misbehavior.

Linkage Authority is responsible for:

- Generating linkage values as response to RA and PCA requests,
- Communicate only with RA to provide these values.

The figure 9 shows a pair of LAs (LA1 and LA2); it provides more privacy to the system.

Location Obscurer Proxy (LOP)

Communications between OBE (on-board equipment) and SCMS components must pass through LOP.

The main roles of LOP are:

- Obscure the location of the OBE seeking to communicate with the SCMS functions,
- Shuffle misbehavior reports that are sent by OBEs to the MA (for more privacy purposes),
- Increases participant privacy.

Misbehavior Authority (MA)

This entity is responsible for detecting misbehavior in the system by performing plausibility checks to messages, or detecting potential malfunction or malfeasance within the system. Its main roles:

- Process misbehavior reports
- Produce and publish the certificate revocation list (CRL)
- Works with Pseudonym CA, Registration Authority (RA), and LA to acquire necessary information about a certificate and create entries to the CRL through CRL Generator.

Pseudonym Certificate Authority (PCA)

The main roles of this authority are:

- Issues short term certificates,
- Collaborates with the MA, RA, and LA to identify linkage values to place on the CRL if misbehavior has been detected.

Pseudonym certificates are used to authenticate messages (BSM) originating by a device. Their lifetime is no longer valid to a fixed period, it changes according to a variable length of time, which make them harder to track.

Registration Authority (RA)

The main roles of this authority are:

- Receives certificate requests from the OBE via LOP,
- Requests and receives linkage values from the LAs
- Performs the necessary key expansions before the PCA performs the final ones.
- Sends certificate requests to the PCA
-

RA receives requests from different OBEs, and in order to prevent correlating certificates IDs with users, it shuffles these requests before sending it to the PCA. Additionally, it maintains a blacklist of enrollment certificates to reject any request from a revoked OBE.

Request Coordination

In case of multiple RAs within the SCMS, Request coordination function role becomes critical. It collaborates with RAs in order to prevent an OBE from receiving multiple certificates from different RAs.

Root Certificate Authority (Root CA)

It represents the center of trust of the system, and produces a self-signed certificate verifying its own trustworthiness. The main role of this authority is to issue certificates to subordinate CAs such as MA, LAs, and RAs.

Root CA operates in offline environment to prevent any security threat which can have a critical impact on the security of the whole system.

SCMS Manager

SCMS Manager is the primary managerial component of the SCMS, it is responsible for managing all other component entities called Certificates Management Entities or CMEs. It provides the policy and technical standards for the V2V system, insures interoperability, security, privacy and auditing of the system, and manages the activities required for operation of the SCMS.

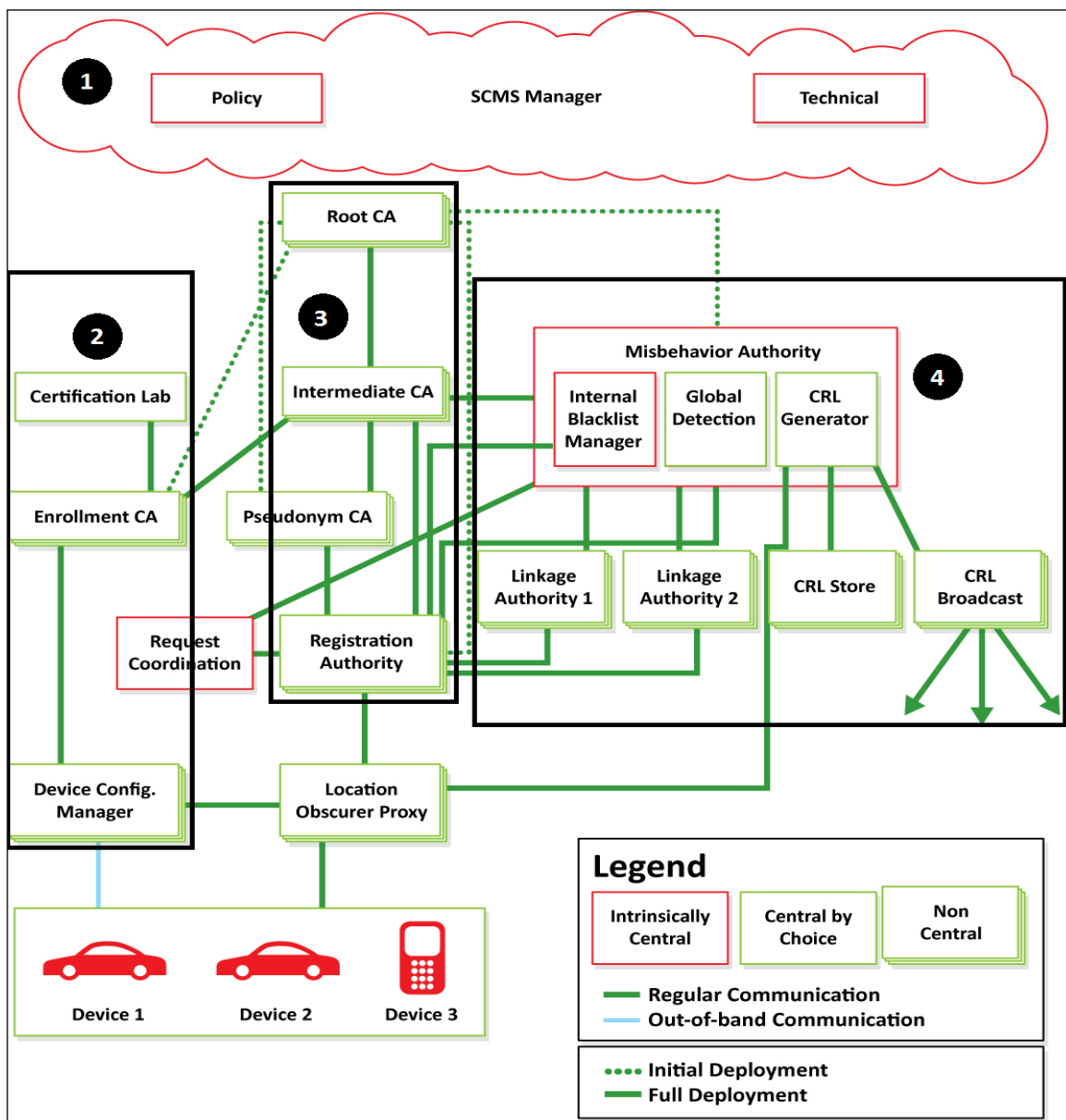


Figure 9: Current V2V security system design for deployment and operations

2. “Bootstrap”/initialization functions/enrollment certificate

In addition to pseudonym functions, the security design also includes bootstrap process. The Enrollment CA (ECA) is the functional component of this process, it assigns a long-term certificate to V2V devices at the first connection to the SCMS. Bootstrap process includes following functions:

Certification Lab

Provides ECA with policies and rules for issuing enrollment certificates. This is usually done when a new device is released to the market or if the SCMS Manager releases new rules and guidelines.

Device Configuration Manager (DCM)

This entity is responsible of:

- Giving devices access to new trust information such as updates to authorities' certificates, policy decisions, and technical guidelines issued by SCMS Manager,
- Sending software updates to devices,
- Coordinating initial trust distribution with devices by passing on credentials for other SCMS entities,
- Providing devices with information it needs to request short-term certificates from RA,
- Providing secure channel to the ECA to communicate Enrollment certificates devices.

Two types of connections are used between devices and DCM, an in-band communication that passes through LOP, and an out-of-band communication that passes directly from the device to the ECA via DCM.

Enrollment Certificate Authority (ECA)

It produces the enrollment certificate and sends it to the OBE, but first it verifies the validity of the device type with the Certification Lab. The OBE uses the enrollment certificate to be able to request and receive certificates from the SCMS.

3.5 Security framework in Japan [15]

Japan has developed a threat and risk analysis based on 15 expected threats between V2I and V2V communication, and a list of countermeasures related to security issues was proposed, such as, using encryption technology for inter-vehicle and roadside to vehicle communication, verify authenticity of the sender, integrity checks and confidentiality maintenance. The digital signature method for V2V and V2I communication proposed by Japan is similar to European and US approach for public key infrastructure. It is also based on the concept of CAs that deliver certificates to different entities of the system. The figure 7 below presents the CA concept in Japan.

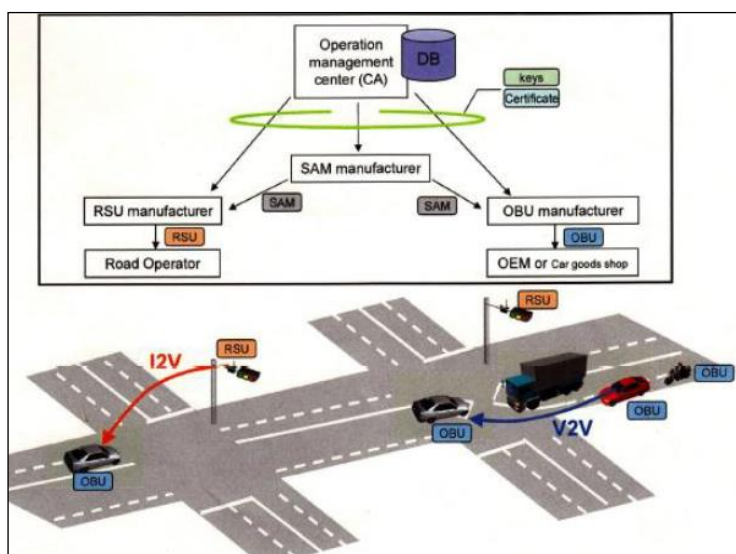


Figure 10: CA concept in Japan

Finally, Japanese industry and government organization are closely following the standards and deployment preparation in both Europe and USA in order to align both communication and security framework.

4. Certificates formats

4.1 IEEE 1609.2 [7]

The IEEE 1609.2 standard supports both explicit and implicit certificates.

Explicit certificate includes the public key certified by the certificate and the digital signature of the certificate issuer. A user can verify the certificate by verifying the signature of the issuer.

Implicit certificate is a variant of public key certificate. It does not explicitly include the public key certified by the certificate but instead allows the public key to be reconstructed from a reconstruction value and the certificate authority's public key. An implicit certificate does not include the signature of the certificate issuer.

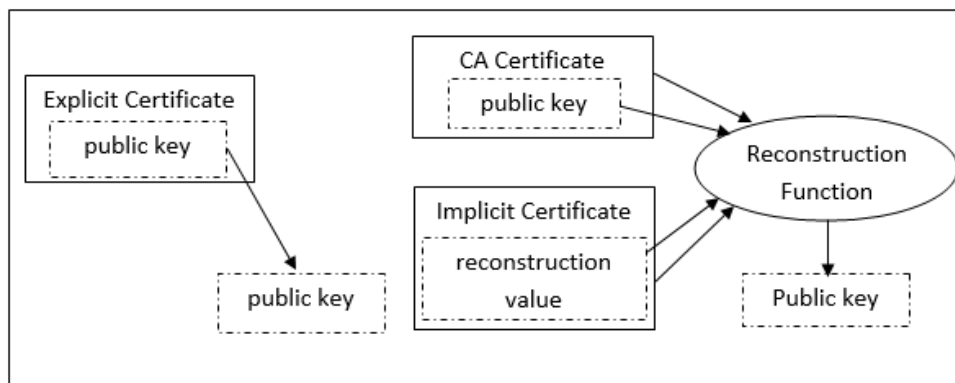


Figure 11: IEEE 1609.2 Explicit and Implicit certificates

Figure 13 shows the certificate format defined in IEEE 1609.2. It consists of three parts:

- A header field called **Version-And-Type**, contains the version of the certificate format and indicates whether the certificate is explicit or implicit.
- The unsigned certificate in a **To-Be-Signed-Certificate** format that contains the certificate contents.
- The **Signature** of the certificate issuer for explicit certificate or a reconstruction value for reconstructing the public key for an implicit certificate.

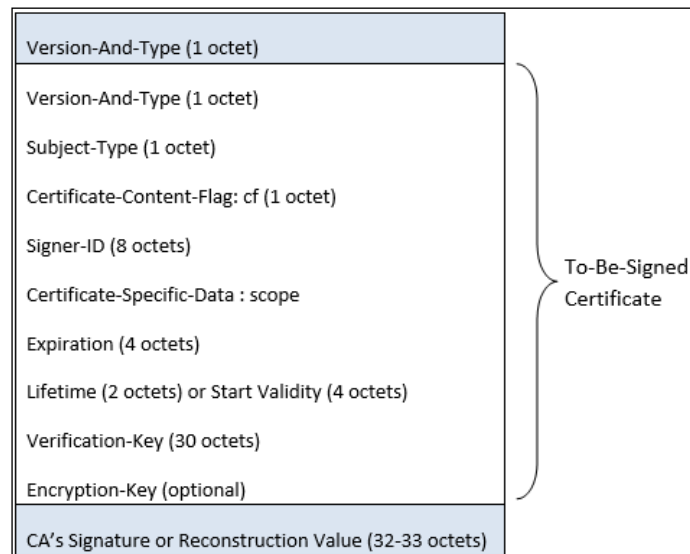


Figure 12: IEEE 1609.2 Certificate format

4.2 ETSI certificates

The ETSI TS 103 097 [12] specifies security header and certificate formats. These formats are defined specifically for securing G5 communication.

ETSI certificate format in the following, we give in detail ETSI certificate's elements.

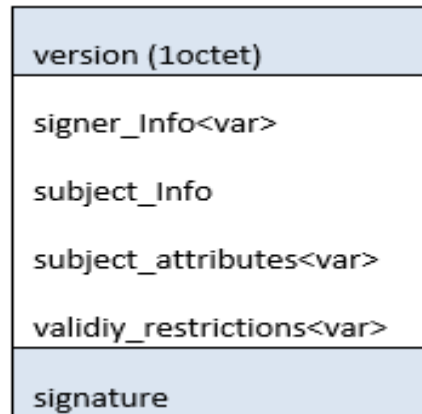


Figure 13: ETSI Certificate format

- **version:** specifies the certificate's version. According to ETSI TS 103 097 v1.1.15 standard, the version shall be set to 2.
- **signer_info:** contains information about the certificate's signer. There are multiple types of signer_info, which are:
 - self: implies that the data is self-signed; no additional data shall be given.
 - certificate_digest_with_sha256: implies that 8 octet digest of the relevant certificate contained in a HashedId8 structure shall be given.
 - certificate: implies that the relevant certificate of the signer CA shall be given.
 - certificate_chain: implies that the complete certificate chain up to the Root CA or a subordinate CA shall be given.
 - certificate_digest_with_other_algorithm: implies that 8 octet digest contained in a HashedId8 structure and the corresponding public key algorithm contained in a PublicKeyAlgorithm structure shall be given.
 - reserved: represent all other cases.
- **subject_info:** specifies information on this certificate's subject. It contains the subject_name which is a variable-length vector, and the type of information represented in the subject_type field, which can be:
 - enrollment_credential: used by the ITS station when communicating with Enrollment CAs
 - authorization_ticket: used by ITS station, when communicating with other ITS stations.
 - authorization_authority: used by Authorization CAs, which sign authorization tickets (pseudonyms).
 - enrollment_authority: used by Enrollment CAs, which sign enrollment credentials (long term certificates).
 - root_ca: used by Root CAs, which sign certificates of other CAs.
 - crl_signer: used by certificate revocation list signers.

- **subject_attributes:** contains additional information about certificate's subject. These attributes specify the technical details of a certificate's subject. There are various types of subject_attributes, and depending on the value of type, additional data shall be given:
 - verification_key: a public key shall be given.
 - encryption_key: a public key shall be given.
 - reconstruction_value: an ECC (Elliptic Curve Cryptography) point contained on an EccPoint structure shall be given. It represents a public key based on elliptic curve cryptography.
 - assurance_level: the assurance level for the subject contained in a SubjectAssurance structure shall be given. The assurance level is a way to represent the ITS-S security of both the platform and storage of secret keys, as well as the confidence in this assessment.
 - its_aid_list: ITS-AIDs (ITS Application ID) contained in a variable-length vector of type IntX shall be given.
 - its_aid_ssp_list: ITS-AIDs with associated SSPs (Service Specific Permissions) contained in a variable-length vector of type ItsAidSsp shall be given.
- **validity_restrictions:** specifies restrictions regarding this certificate's validity. It's a variable length vector that may contain one of the different validity_restriction types below:
 - time_end: represents the expiration date for the associated certificate
 - time_start_and_end: represents the beginning of the validity and expiration data
 - time_start_and_duration: represent the beginning of the validity and the duration of the validity.
 - region: represent the region where the certificate is valid.
- **signature:** contains signature of the certificate signed by the responsible CA. The signature shall be calculated over the encoding of all preceding fields, including all encoded lengths (In case where subject_attributes field contains a field of type reconstruction_value, the signature field shall be omitted).

5. Security profiles according to ETSI 103 097 standard [12]

5.1 Security profile for CAM

The fields that shall be included in the SecuredMessage structure for Cooperative Awareness Messages (CAMs) are represented in the table below:

SecuredMessage		
Fields	Value / type that shall be included / field	Description
protocol_version	unit8	Protocol version shall be 2 for the current version of TS 103 097 standard [12]
Header Field signer_info (Shall be included in all CAMs)	certificate_digest_with_sha256	1) Shall be included in all normal cases

SecuredMessage		
Fields	Value / type that shall be included / field	Description
Header Field	certificate_chain or certificate	1) Shall be included one second after the last inclusion of a field of type certificate 2) If the ITS-S receives a CAM from a previously unknown other certificate, it shall include a field of type certificate immediately in the next CAM + restart the timer of the next inclusion of a field of type certificate 3) If an ITS-S receives a CAM whose security header includes a Header Field of type request_unrecognized_certificate, then the ITS-S shall evaluate the list of HashedId3 digests included in that field. <ul style="list-style-type: none"> If the ITS-S find a HashedId3 of its own, currently used authorization ticket and not of the authorization authority in that list, it shall include a signer_info field of the type <u>certificate</u> immediately in the next CAM, instead of including a signer_info field of type certificate_digest_with_sha256. If the ITS-S finds a HashedId3 of its own, currently used authorization authority in that list, it shall include a signer_info field of type <u>certificate_chain</u> containing the currently used authorization ticket and authorization authority certificate immediately in its next CAM, instead of including a signer_info field of type certificate_digest_with_sha256.
	generation_time	Shall be included in all CAMs This field shall contain the current absolute time. The generation_time is valid, if it is in the validity period of the certificate referenced by the signer_info.
	its_aid	This field shall encode the decimal value for CAMs according to ETSI TR 102 965 and ISO TS 17419 ITS-AID registration list standard

SecuredMessage		
Fields	Value / type that shall be included / field	Description
request_unrecognized_certificate	digests<var>	<p>1) Shall be included if an ITS-S received CAMs from other ITS-Ss, which the ITS-S has never encountered before and which included only a signer_info field of type certificate_digest_with_sha256 instead of a signer_info HeaderField of type certificate. In this case the signature of the received CAMs cannot be verified because the verification key is missing.</p> <p>2) The field digests<var> in the structure of request_unrecognized_certificate shall be filled with a list of HashedId3 elements of the missing ITS-S certificates.</p> <p><u>Note:</u> HashedId3 elements can be formed by using the least significant three bytes of the corresponding HashedId8</p>
<p>Note:</p> <ul style="list-style-type: none"> • None of the possible HeaderField cases shall be included more than once. • All other HeaderField (defined in clause 5 in the ETSI TS 103 097 standard) types shall not be used. • Future HeaderField types may be included. • Any other HeaderField types included shall not be used to determine the validity of the message. 		
Payload		<p>1) Shall be included for all CAMs.</p> <p>2) This element shall be of type signed and contain the CAM payload.</p>
TrailerField	signature	<p>Shall be included in all CAMs</p> <p>The signature is calculated over these fields of Secured Message data structure:</p> <ul style="list-style-type: none"> • - protocol_version • - The variable-length vector header_fields including its length • - The complete payload_field field • - The length of the variable-length vector trailer_fields, and the type of the signature trailer field • - If the payload is marked as external, its contents shall be included in the hash as well, at the position where a non-external payload would be. • - The length of the variable-length vector trailer-fields and all data preceding the signature, including the length of the signature fields.

Element	Description
• SecuredMessage	Covered by the signature
• uint8 protocol_version	
• HeaderField header_fields<var>	
• ...	
• Payload payload_fields<var>	
• ...	
• TrailerField trailer_fields<var>	
• TrailerFieldType type	Not covered by the signature
• PublicKeyAlgorithm algorithm	
• EcdsaSignature ecdsa_signature	
• EccPoint R	
• EccPointType type	
• opaque x[32]	ECDSA signature (r,s)
• opaque s[32]	

Table 3: Example of the ECDSA signature generation for a SecuredMessage

The following structure shown in table 4 is an example of security header for a CAM message. The header transports the generation time, identifies the payload as signed, and includes the hash of a certificate, that is, no full certificate is included in this case. Finally, an ECDSA NIST P-256 based signature is attached.

Element	Value	Description	Length in octets
• SecuredMessage			
• uint8 protocol_version	0x02		1
• HeaderField header_fields<var>	0x15	length: 21 octets	1
• HeaderFieldType type	0x80	signer_info	1
• SignerInfoType signer_info	0x01	certificate_digest_with_sha256	1
• HashedId8 digest	[...]		8
• HeaderFieldType type	0x00	generation_time	1
• Time64 generation_time	[...]		8
• HeaderFieldType type	0x05	its_aid	1
• IntX its_aid	0x24	ITS-AID for CAM	1
• Payload payload_field		payload	
• PayloadType payload_type	0x01	signed	1
• opaque data<var>	0x00	length: 0 octets	1
• [raw payload data]			0
• TrailerField trailer_fields<var>	0x43	length: 67 octets	1
• TrailerFieldType type	0x01	signature	1
• PublicKeyAlgorithm algorithm	0x00	ecdsa_nistp256_with_sha256	1
• EcdsaSignature ecdsa_signature			
• EccPoint R			
• EccPointType type	0x00	x_coordinate_only	1
• opaque x[32]	[...]		32
• opaque s[32]	[...]		32
The total size of the security header structure is 93 octets.			

Table 4: An example signed header for CAM

5.2 Security profile for DENM

DENMs shall not be encrypted, but some cryptographic applications can be applied to the header like the signature of the message. The fields that shall always be included in the SecuredMessage structure for Decentralized Environmental Notification Messages (DENMs) are represented in the table below:

SecuredMessage		
Fields	Value / type that shall be included	Description
protocol_version	unit 8	Protocol version shall be 2 for the current version of TS 103 097 standard [12]
HeaderFi signer_info	certificate	<ul style="list-style-type: none"> - Shall be included in all DENMs - This field shall contain an element of type certificate

SecuredMessage		
Fields	Value/type that shall be included	Description
HeaderField	generation_time	<ul style="list-style-type: none"> - Shall be included in all DENMs - This field shall contain the current absolute time - The generation_time is valid, if it is in the validity period of the certificate referenced by the signer_info
	generation_location	<ul style="list-style-type: none"> - Shall be included in all DENMs - This field shall contain the current location of the ITS-S at the point in time the contents of the security headers are fixed prior to the signing process. - The generation_location is valid, either if there is no geographic validity restriction in the certificate referenced by the signer_info, or if it is inside the geographic validity restriction of this certificate.
	its_aid	<ul style="list-style-type: none"> - This field shall encode the decimal value for DENMs according to ETSI TR 102 965. - It is equal to 0x25 in the one octet field, according to ISO TS 17419 ITS-AID registration list standard.
	Note : <ul style="list-style-type: none"> • None of the possible HeaderField cases shall be included more than once. • All other HeaderField types shall not be used. • Future HeaderField types may be included. • Any other HeaderField types included shall not be used to determine the validity of the message. 	
Payload	signed	<ul style="list-style-type: none"> - At least one Payload element shall be included in all DENMs - This element shall be of type signed and contain the DENM payload.
TrailerFields	signature	<ul style="list-style-type: none"> - signature is a TrailerField element that shall be included in all DENMs - The signature is calculated over these fields of SecuredMessage data structure: <ul style="list-style-type: none"> - protocol_version - The variable-length vector header-fields including its length - The complete payload_field field - The length of the variable-length vector trailer-fields and the type of the signature trailer field

Table 5: Fields that shall be included in SecuredMessage structure for DENMs

Conclusion

The state of the art presented in this document shows that almost all organizations and research groups which are working on the C-ITS project are interested by the system security, and the privacy of users, and they are looking for the appropriate solution for that. For these reasons researches on security and privacy have been developed in several European, and US projects.

The first chapter of this state of the art presented three different cryptographic mechanisms, which are encryption algorithms including symmetric and asymmetric encryption, digital signature, and asymmetric public key infrastructure (PKI). We are interested by the PKI since it is a worldwide commonly followed approach, and it represents the most effective solution for C-ITS systems. In fact, it provides trusted message exchange for a very large set of users, secure communications especially for safety-critical applications which trigger their actions based on data received from other network entities, ensures integrity and non-repudiation, and protects driver privacy appropriately by not requiring participants to disclose their identities.

The common primary objectives of a PKI suggested by the major of standards are issuing and provisioning of valid certificates to respective ITS stations, limiting digital credentials misuse by controlling their validity, and excluding compromised ITS stations or PKI entities from the network activities by revoking their credentials.

Industry organizations such as Car2Car Communication Consortium and the CAMP/VIIC in the USA have developed a security framework which is coordinated between these organizations. In Europe, projects concerned by C-ITS takes the System Security very seriously. Discussions with national IT-Security authorities and responsible bodies are organized to involve technical aspects like certificates, suitable encryption algorithms and hardware requirements, as well as organizational aspects. In USA, Safety Pilot Model Deployment project has identified some security issues related to privacy, authentication, false messages, and denial of services, and they deduced that implementing a PKI solution and a digital signature is a must. Both European and USA security approaches understand the need for an SCMS-like PKI system; the SCMS would operate a PKI structure in order to maintain secure communication within the system. USA and Europe wants to employ Long-Term CA and Pseudonym CA, however some differences still exist; US structure contains three authorities not currently found in the European structure, which are: Linkage Authority, Misbehavior Authority and Registration Authority.

Finally, the common primary objectives of a PKI suggested by major working groups and described in different standards are still quite similar: issuing and provisioning of valid certificates to ITS stations, limiting digital credentials misuse by controlling their validity, and excluding compromised ITS stations or PKI entities from the network activities by revoking their credentials [16].

In SCOOP@F project, we focus on the common model defined by ETSI/IEEE/C2C and ETSI's certificates formats.

References

- [1] http://en.wikipedia.org/wiki/Symmetric-key_algorithm.
- [2] <http://msdn.microsoft.com/en-us/library/ff650720.aspx>.
- [3] NIST Special Publication SP 800_38C, Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality.
- [4] <http://msdn.microsoft.com/en-us/library/ff647097.aspx>.
- [5] http://en.wikipedia.org/wiki/Public_key_certificate.
- [6] Public Key Infrastructure (PKI) - Infrastructure de Gestion de Clé (IGC) - Ahmed Serhrouchni Tutorial.
- [7] IEEE Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages.
- [8] Rim Moalla: Securing Future Cooperative ITS Applications, thèse de doctorat présentée le 29/09/2014.
- [9] ETSI TS 102 940 V1.1.1 (2012-06) Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management.
- [10] ETSI TS 102 941 V1.1.1 (2012-06) Intelligent Transport Systems (ITS); Security; Trust and Privacy Management.
- [11] ETSI TS 102 731 V1.1.1 (2010-09) Intelligent Transport Systems (ITS); Security; Security Services and Architecture.
- [12] ETSI TS 103 097 V1.1.15 (2014-11) Intelligent Transport Systems (ITS); Security; Security header and certificate formats.
- [13] C2C-CC PKI Memo – version 1.20 – January 2011.
- [14] Vehicle-to-Vehicle Security Credential Management System; Request for Information – Daniel C. Smith, Senior Associate Administrator for Vehicle Safety.
- [15] D2.4 Non-Technical Deployment Elements – v1.0 – COMeSafety2.
- [16] D3.5 Advances in harmonizing the Deployment Approach for C-ITS in Europe – v1.0 – COMeSafety2.