



Renewal of pseudonym certificates and upload of Logs (T-Logs and U-Logs)

Deliverable 2.4.1_appendix_1

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

Information about the document

Document: Renewal of pseudonym certificates and upload of Logs (T-Logs and U-Logs)

Date of publication: 12/05/2017

Responsible, Entity: Houda LABIOD, Telecom ParisTech

Status: Version 2.00 – Approved

Input documents: Livrable 2.4.1, Documents provided by C. Lauer, G. Segarra, A. Serval, B. Lonc, C. Tissot.

Publication history

Date	Version	Contributor(s)	Updates & changes	Diffusion
16/10/2015	1.00	H. Labiod		Release 1
12/05/2017	2.00	H. Labiod	Changes: - decision to not use NACS protocol - definition to specify one field for IP Address instead of two - requirement for occurrence of changes related to MAC address and pseudonym certificates Updates: - modifications of CAM-I's fields - correction to ensure global coherence with others deliverables	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

1.	Objective.....	5
2.	ETSI ITS-S reference communication architecture.....	5
3.	Security architecture	7
4.	IEEE WAVE communication architecture	9
5.	Advertisement of services.....	10
5.1	WSA	10
5.2	SAM-WSA	10
5.3	Solution proposed by SCOOP@F	11
6.	NACS: Access control protocol for advertised services	22
7.	Services.....	27
7.1	Renewal of pseudonym certificates	27
7.2	Upload of Logs	28
8.	Conclusion	28
9.	References	29

Table of figures

Figure 1: ITS station architecture (from EN 302 665 [3])	5
Figure 2: ETSI ITS reference architecture	6
Figure 3: Used Protocols (ETSI TS 102 636-6)	6
Figure 4: ITS communications architecture.....	7
Figure 5: Architectural ITS security layers	7
Figure 6: ITS-S security architecture: entities and interfaces	8
Figure 7: The placement of security services within the ITS station architecture	8
Figure 8: IEEE WAVE reference architecture	9
Figure 9: Security Services and entities in WAVE protocol stack	10
Figure 10: General structure of a CAM	12
Figure 11: CAM-I structure	12
Figure 12: Service Advertisement Container	15
Figure 13: Service advertisement with CAM-I message	15
Figure 14: Coding structure of CAM-I message	16
Figure 15: Service Access Request (SAReq) Message structure	22
Figure 16: Service Access Response (SARep) Message structure.....	25
Figure 17: Renewal pseudonym certificates service.....	27

Table of tables

Table 1: Structure of the proposed CAM-I	13
--	----

1. Objective

The present document aims at describing the set of used protocols involved to carry PKI requests as specified in deliverable 2.4.4-6 [1] as well as Logs specified in deliverable 2.3.1.1 [2]. The documents starts by summarizing the technical details related to the security architecture for Intelligent Transport System (ITS) communications as defined in EN 302 665 [3]. The present document also identifies the security layers and interfaces between communication plane and security and management plane in ETSI ITS reference architecture and IEEE WAVE reference architecture (presented in sections 2, 3 and 4). In Section 5, we specify in detail the solution proposed by the Project SCOOP@F to advertise two services considered in the first part of the project: pseudonym certificates renewal and Logs update. The last part of the document describes the protocols needed to transport the PKI requests and logs from vehicles through SCOOP's infrastructure. In Section 6, we give the details on the service access control protocol. Section 7 is composed of two sections. Section 7.1 presents the process of the execution of the first considered service (renewal of pseudonym certificates) composed of three phases: advertisement of the service, access control to the network and the service execution. Similarly, in Section 7.2, the service of logs upload is described through the same three phases. Implementation details related to these services are given in Deliverable 2.4.4-8.

2. ETSI ITS-S reference communication architecture

EN 302 665 [3] describes an ITS-S station architecture based upon 4 processing layers identified as follows:

- Access Layer;
- Networking & Transport Layer;
- Facilities Layer; and
- Applications Layer.

Additionally to these horizontal layers, a vertical Management layer and a Security layer (Figure 1) are defined used for management and security purposes.

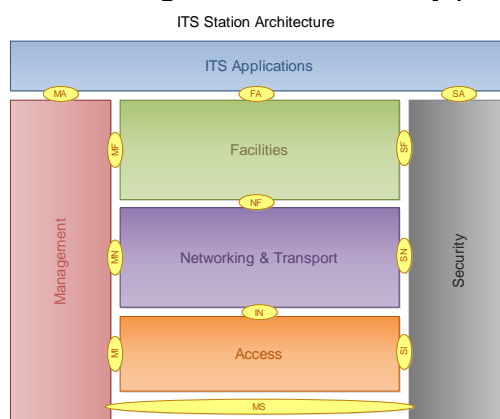


Figure 1: ITS station architecture (from EN 302 665 [3])

Figure 2 gives a more detailed view of this architecture.

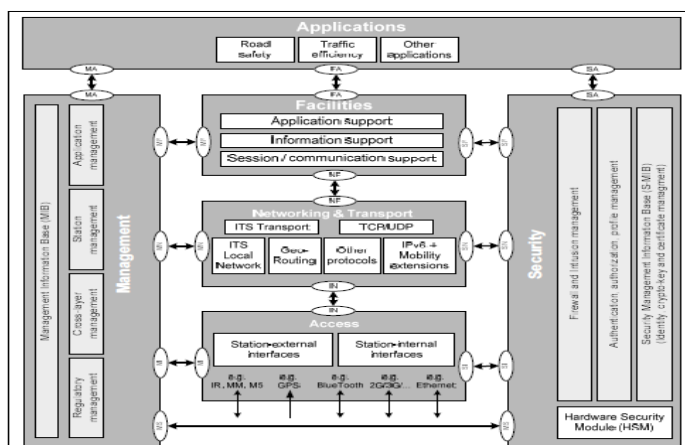


Figure 2: ETSI ITS reference architecture

The various functionalities provided by each layer are as follows:

- The *Access layer* manages the wired and wireless access technologies that are available in ITS stations (e.g. 802.11p, 802.3, 802.11a/b/g/n, etc.). It can be mapped to the *Physical* and *Data Link* layers of the OSI model.
- The *Networking and Transport layers* provide data transport between source and destination ITS stations. ETSI considers the three following combination of networking and transport protocols to achieve this task: BTP over GeoNet, TCP/UDP over IP, TCP/UDP over IP over GeoNet. This layer can be mapped to the *Network layer* and the *Transport layer* of the OSI model.
- The *Facilities layer* provides support to ITS application by sharing generic functions and data such as: generic HMI support, data presentation (e.g. ASN.1), environment information (time, location, etc.), addressing mode and channel selection at lower layers, etc. It can be mapped to the *Session, Presentation and Application layers* of the OSI model.
- The *application layer* runs the ITS applications.
- The *Management cross layer* manages the communication stack depending on ITS applications requirements.
- The *Security cross layer* is in charge of securing communications between ITS stations. To this end, it provides security services to the communication stack and their management.

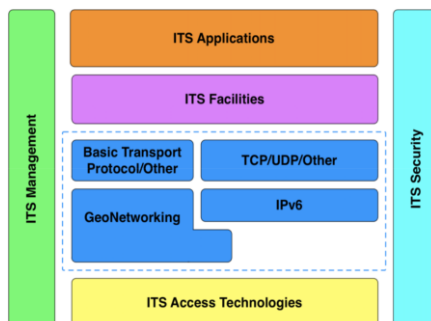


Figure 3: Used Protocols (ETSI TS 102 636-6)

ETSI has standardized a protocol adaptation sub-layer referred to as the GN6ASL (GeoNetworking to IPv6 Adaptation Sub-Layer) which allows for the transport of IPv6

packets by ETSI GeoNetworking protocol, enabling sub-IP multi-hop delivery of IPv6 packets. The ETSI GN geo-broadcasting capability is used by the GN6AASL in order to shape link-local multicast messages to geographical areas. Figure 4 shows the peer-to-peer communications between layers of two ITS-S stations.

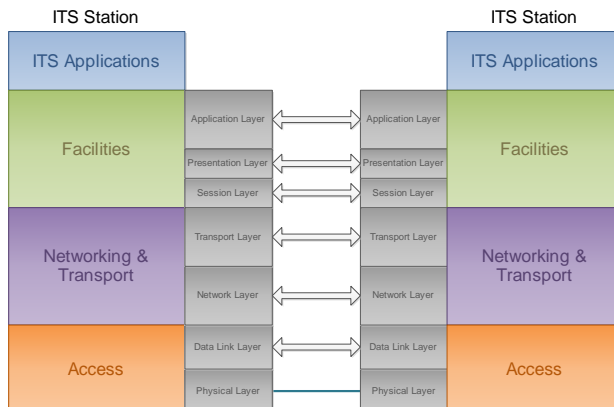


Figure 4: ITS communications architecture

3. Security architecture

EN 302 665 [3] shows Security as a vertical layer adjacent to each of the ITS layers but, in fact, security services are provided on a layer-by-layer basis so that the security layer can be considered to be subdivided into the four basic ITS processing layers as shown in Figure 5. 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.

Figure shows the security services and Security Management functional entities. Furthermore, the Security Entity of the ETSI ITS architecture as described in Figure 2 provides a third sublayer: the security defence layer of the communicating ITS-S, which prevents direct attacks against critical system assets and data and increases the likelihood of the attacker being detected (e.g. Firewall and Intrusion detection or prevention). This latest sublayer is beyond the scope of the present document.

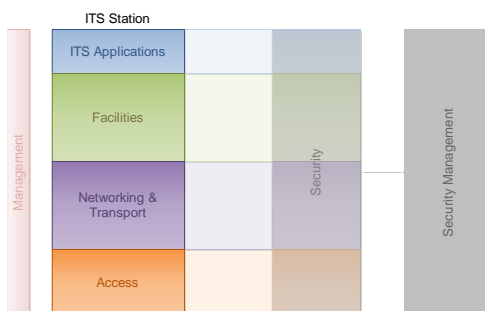


Figure 5: Architectural ITS security layers

Figure 6 shows the functional entities of the ITS-S communications security architecture and the relationship that exist between themselves and the ITS-S communication layers

(specified in SF-SAP, SN-SAP and SI-SAP interfaces).

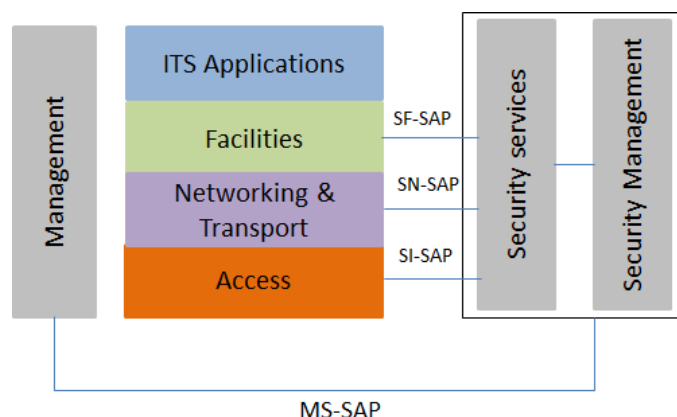
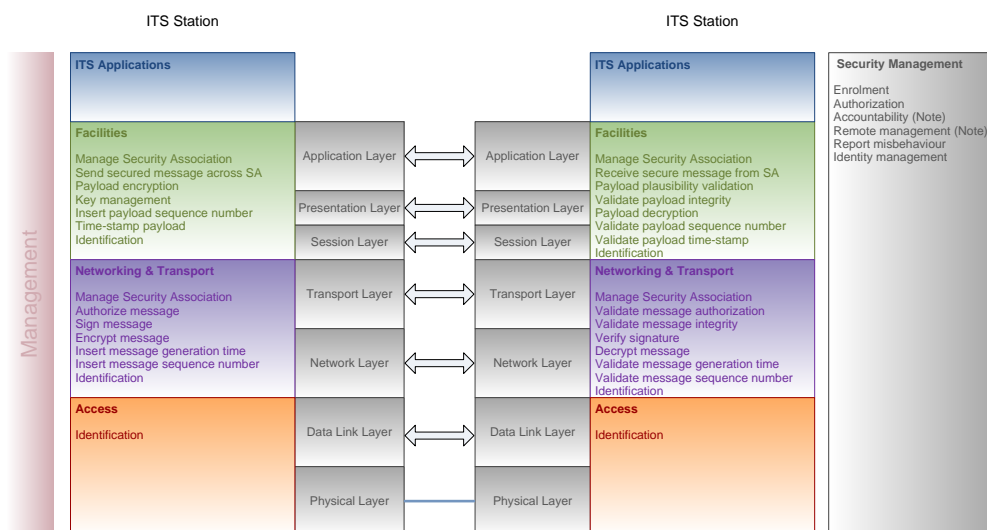


Figure 6: ITS-S security architecture: entities and interfaces

Note: the ITS-S security defense layer is not shown here.

Although the security layer is presented as a transversal layer in the ITS communication architecture, ITS security services are in fact provided on a layer-by-layer basis. That is, security services operate on different layers of the communication stack, as depicted in Figure 7. Furthermore, the security layer in ETSI ITS architecture is composed of two sub-layers: security management services and security defense layer of the communicating ITS-S (ITS-S system protection against wireless network attacks from bad actors, e.g. Firewall and Intrusion detection or prevention).



NOTE: This figure is based on ETSI TS 102 731 security services. The Accountability and Remote management security management services are not specified.

Figure 7: The placement of security services within the ITS station architecture

SF-SAP, SN-SAP and SI-SAP are defined in TS 102 723-7/8/9.

Security management

Each secure ITS station shall be able to store and protect its security related material such as certificates and encryption keys. ETSI proposes to achieve this goal by including a Hardware Security Module (HSM) into the ITS station. The HSM would be in charge of providing security related services (like certificate/key storage and protection or cryptographic operations computation) to all other ITS applications and security services.

4. IEEE WAVE communication architecture

WAVE reference architecture (or protocol stack) is defined in [4] and depicted in Figure 8. It is divided in two planes: a data plane and a management plane. The **data plane** corresponds to the communication stack. It is composed of the following layers:

- The *physical (PHY) and medium access control (MAC)* layers which are based on (and only support) the IEEE 802.11 standard. That is, only wireless communications are considered in WAVE.
- The *logical link control (LLC)* layer which is based on the IEEE 802.2 standard.
- The *IPv6 and UDP/TCP* layers which are the common Network and Transport protocols used over the Internet. Note that WAVE only supports IPv6.
- The *WAVE Short Message Protocol (WSMP)* layer which is especially designed for message exchanges in a vehicular environment.

The **management plane** provides security and management support through:

- A *security layer* which provides security services and their management
- A *management layer* which manages and coordinates layers of the data plane

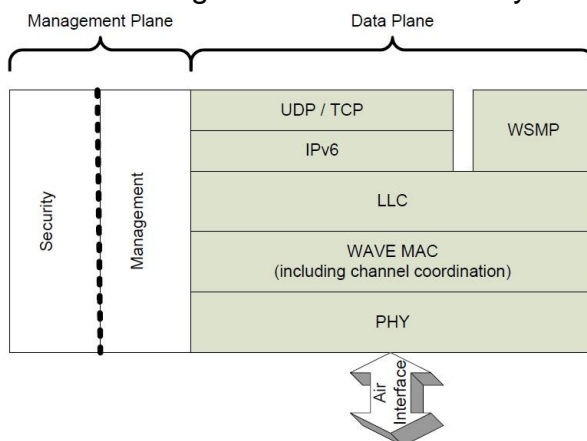


Figure 8: IEEE WAVE reference architecture

Security architecture

WAVE security architecture is shown in Figure 9. The placement of the security services described above in the communication architecture is detailed as well as the corresponding Service Access Points (SAP).

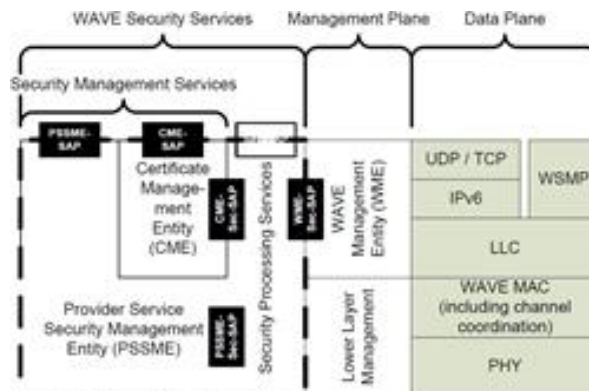


Figure 9: Security Services and entities in WAVE protocol stack

5. Advertisement of services

Advertised services refer to services where a provider network sends out a message of particular type advertising that the service is being offered and an ITS-S with the corresponding user application connects to the service. In this section, we briefly remind the main advertisement messages defined by IEEE and ETSI. Then, we present the message and the associated mechanism proposed by the SCOOP@F project.

5.1 WSA

WAVE Service Announcement (WSA) is described in IEEE 1609.3 but it does not preclude any alternative method of providing Service Announcements including ETSI Facilities service announcement TS 102 890-2. Advertisements are not application messages themselves, though they may contain information allowing the user application to decide whether to connect. For example, a service advertisement for entertainment services might contain an identifier for the media provider. They are broadcasted as unencrypted messages and usually sent multiple times a second. We can cite some examples: Public transport information, Traffic information and recommended itinerary, Point of interest notification, Automatic access control and parking management, Media downloading.

5.2 SAM-WSA

ETSI standard « SAM » TS 102 890-2 describes the SAM (Service Announcement Message) used for service advertisement. It is being developed, a harmonization work with the IEEE "WSA" 1609.3 Rev 3 (March 2014) based on the American standard WAVE is ongoing. SAM-WSA is defined in the Facilities layer. It uses the same communication profile as the CAM message. The structure of SAM-WSA message is given in the standard IEEE 1609.3 Rev 3.

5.3 Solution proposed by SCOOP@F

5.3.1 Context

Work is underway to harmonize standards ETSI / IEEE SAM / WSA within ETSI TC ITS WG2 group to adapt the American standard to ITS G5 context (SAM messages using network/transport layer and to take into account the specific needs expressed by ETSI TC ITS WG5 (certificates renewal).

Taking into account SCOOP@F project's deployment constraints and the delay due to standardization activities, SCOOP@F partners decided to adopt an alternative solution (CAM-I) based on CAM message described in section 5.3.2. The idea of using the CAM-I message is interesting because it is compliant with the CAM existing standard and responds to SCOOP@F project requirements to support pseudonym certificates renewal and logs upload.

5.3.2 Solution for service advertisement based on the use of a CAM-Infrastructure (CAM-I) message

The available services are advertised via the periodic broadcasting of a specific CAM message from a RSU (ITSS-R). It is called a CAM-Infrastructure (CAM-I) message which has the same structure as a CAM message sent by a vehicle (ITSS-V).

In the phase 1 of the project SCOOP@F, we consider the following services:

- Pseudonym certificates renewal requests that are transported to the PKI.
- Upload of log files (T-logs and U-logs).
- Data Exchange with some specific applications through the SCOOP@F platform.

These CAM-I messages are also used for advertisement of DSRC road tolling and for the advertisement service to ITSS-V such as the uploading of general purpose information supporting the evaluation during the pre-deployment phase.

A CAM is composed of one common ITS PDU header and multiple containers (see Figure 10). The ITS PDU header is a common header that includes the information of the protocol version, the message type and the ITS-S ID of the originating ITS-S. For vehicle ITS-Ss a CAM shall comprise one basic container and one high frequency container, and may also include one low frequency container and one or more other special containers:

- The basic container includes basic information related to the originating ITS-S.
- The high frequency container contains highly dynamic information of the originating ITS-S.
- The low frequency container contains static and not highly dynamic information of the originating ITS-S.
- The special vehicle container contains information specific to the vehicle role of the originating vehicle ITS-S.

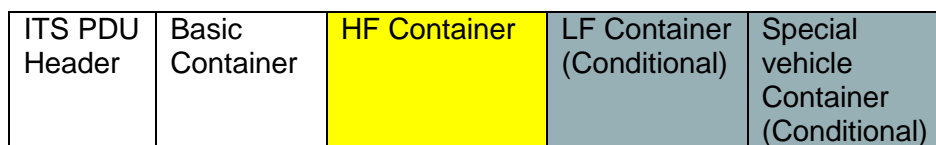


Figure 10: General structure of a CAM

As indicated in the CAM standard, all CAMs generated by a RSU ITS-S shall include a basic container and optionally more containers.

For instance, the proposed CAM-I is composed of an ITS PDU Header, a basic container and a HF container composed of 4 containers.

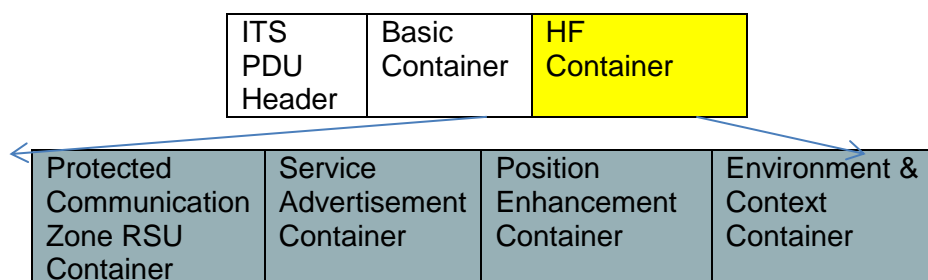


Figure 11: CAM-I structure

The general structure of a CAM-I is illustrated in table 1.

Container	Data Elements	Type	Number of Bytes	M/O	COMMENTS
ITS PDU Header Header	Protocol version	See EN 302637-2 Version 010302		M	Version 1.
	Message ID	See EN 302637-2 Version 010302		M	CAM
	Station ID	See EN 302637-2 Version 010302		M	RSU Identifier
Generation Delta Time	Generation Delta Time	See EN 302637-2 Version 010302		M	
Basic Container	Station Type	See EN 302637-2 Version 010302		M	RSU
	Reference Position	See EN 302637-2 Version 010302		M	Accurate position of the RSU established at installation time.
Protected Communication Zone RSU Container	Data elements for Toll collect protection	See EN 302637-2 Version 010302 and TS 102894-2 v1.2.1		O	Made mandatory in SCOOP at level of toll collect.
Service Advertisement Container	Advertised Service ID	See Figure 12 below 1 Byte		M	In SCOOP, we define: Advertised Service ID= 0 if PKI service supported Advertised Service ID= 1 if upload log service supported
	Service Access Capabilities	See Figure 12 below 1 Byte		M	

	Channel used by the advertised service	Integer	1 Byte	M	SCH1 shall be used in SCOP by the advertised service. Numbering : 0 : CCH, 1 : SCH1 2 : SCH2etc.
	Communication profile used for the service.	Integer	1 Byte	M	See L 2.4.1 for the identification of communication profiles used in SCOP. For this service, it shall be IPv6 over LLC and G5 SCH1 which is coded in L 2.4.1 as CP8 (if TCP) or CP9 (if UDP)
	RSU MAC Address	Hexa decimal	6 Bytes	M	
	RSU IP Address	Hexa Decimal	[4, 16, 20] Bytes	M	In case when IPv4 and/or IPv6 are supported by RSU
Position Enhancement Container	GPS Position Delta Latitude	Similar to EN 302637-2 Version 010302		O	Optional, enable to correct positioning error by comparing the RSU accurate position with the given RSU GPS position.
	GPS Position Delta Longitude	Similar to EN 302637-2 Version 010302		O	Optional, enable to correct positioning error by comparing the RSU accurate position with the given RSU GPS position.
	GPS Position Delta altitude	Similar to EN 302637-2 Version 010302		O	Optional, enable to correct positioning error by comparing the RSU accurate position with the given RSU GPS position.
	Satellite constellation locally available	Integer	1 Byte	O	Enable the use of correction if the vehicle detects the same satellite constellation.
	Traces leading to the RSU	Similar to EN 302 637-3 v1.2.2/TS 102 894-2 v1.2.1		O	Enable the improvement of the vehicle position and the computation of the time window for RSU exchange.
Environment & Context Container	Local meteorological data	Binary	1 Byte	O	Provide local meteorological data for environmental characterization.
	Road environment	Integer	1 Byte	O	Provide the road environment type in which the RSU is positioned
	Traffic condition	Integer	1 Byte	O	Provide the current traffic condition.

Table 1: Structure of the proposed CAM-I

CAM-I structure is similar to standard CAM. Its header (ITS PDU Header) and Basic Container are compliant with the standard ETSI EN 302 637-2 V010302.

The ITS PDU header includes the protocol version, the message type, the ITSS-R ID of

the originating ITSS-R (RSU). A generation delta time of the message is included. The Basic Container includes the following fields:

- The type of the emitting station RSU (ITSS-R): 15 (Road Side Unit).
- The geographic position of the station RSU (ITSS-R) (i.e. the precise position of the RSU).

More specifically, we define a High Frequency Container which is composed of the following containers:

- Service Advertisement Container: It contains information related to service access capabilities and is coded in ASN.1 (UPER coding).
- Position Enhancement Container
- Environment & Context Container
- Protected Communication Zone RSU Container.

In the following, we describe in detail the Service advertisement Container. It includes the following fields:

- Advertisement Service ID: this byte provides the ID of the advertised service.
- Service Access Capabilities (SAC): The second byte details access capabilities to a private network (ex: network of the road operator that offers an access reserved to its staff and also for SCOOP@F vehicles) or to a global network (ex: Internet network that can also be reserved for SCOOP@F vehicles). The choice of access policy for SCOOP@F vehicles depends on the global access policy selected by each road operator.
 - The first bit indicates if a private access or a global access is available. If bit n°1=0 the RSU (ITSS-R) provides access to a private network and if bit n°1=1 the RSU (ITSS-R) provides access to a global.
 - The bit n°2, when having the value " 1 " indicates the capacity of the RSU (ITSS-R) to establish a continuous exchange with a remote server via the private/global global network. If this one has the value " 0 ", the RSU (ITSS-R) does not possess this capacity.
 - The bit n°3, when having the value " 1 " indicates the capacity of the RSU (ITSS-R) to store locally a message and to transmit it as soon as possible towards a remote server (store and forward) via the private/global network, as soon as this one is available. If this bit has the value " 0 ", the RSU (ITSS-R) does not possess this capacity.
 - The bits n°4-5-6-7 and 8 are reserved for a future usage.
- Channel: This field contains the channel used as indicated in the deliverable 2.4.1 v3.2.
- Communication Profile (CP): The fourth byte indicates the communication profile as indicated in the deliverable 2.4.1 v3.2.
- RSU MAC Address (RM): This field indicates the MAC address of the RSU (ITSS-R).
- RSU IP Address (RIP): This field indicates the IP address of the RSU (ITSS-R). If IPv4 is supported the field contains the IPv4 address of the RSU. If IPv6 is supported the field contains the IPv6 address of the RSU. If both IPv4 and Ipv6 are supported, the field contains both addresses. Since the IPv6 address of the RSU (ITSS-R) is sent to ITSS-Vs through RA (router advertisement) messages we can choose to not send it again through CAM-I message. In this case, there is no need to set this address statically in ITSS-Vs.

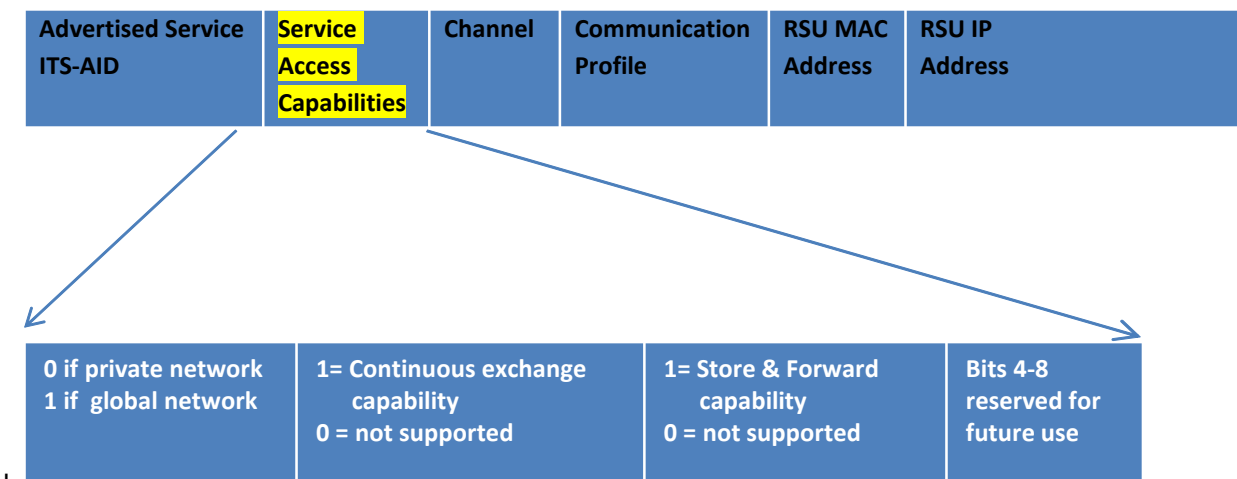


Figure 12: Service Advertisement Container

Periodically, an ITSS-R station (RSU) broadcasts CAM-I messages in order to inform vehicles present in its neighborhood that is capable of forwarding vehicle requests related to the advertised service to access networks (that can be public or private). The frequency of sending CAM-I messages with specific information related to the selected service is given in Deliverable 2.4.4-8.

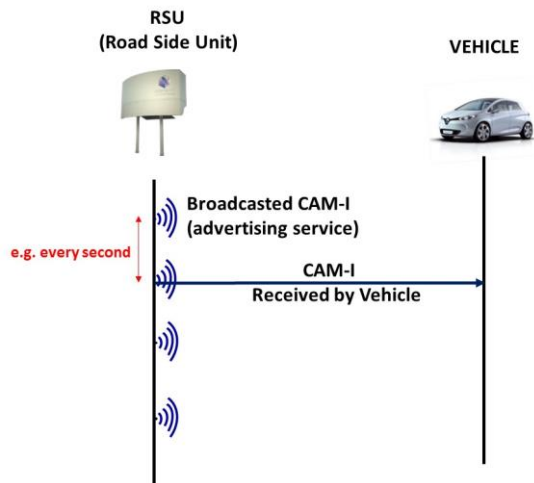


Figure 13: Service advertisement with CAM-I message

CAM-I messages are also used to advertise DSRC road tolling service and uploading of general purpose information supporting the evaluation during the pre-deployment phase. CAM-I messages are signed following the same way as for CAM emitted by vehicles. The fields RSU MAC Address and RSU IP Address should be encrypted but CAM-I encryption will not be supported in SCOOP@F project part 1. To encrypt CAM-I, we can propose to use AES-128-CCM algorithm.

In SCOOP@F project, we use a testing ISO ITS-AID for the CAM-I message set to the value=16490.

General CAM-I ASN.1 structure

The ASN.1 specification of CAM-I is compliant with the ASN.1 specification of CAM detailed in annex A (see EN 302637-2 Version 010302). The coding structure of CAM-I is illustrated Figure 14.

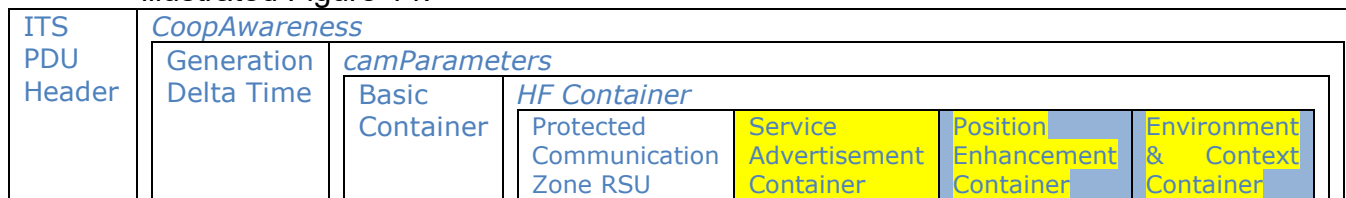


Figure 14: Coding structure of CAM-I message

CAMI-PDU-Descriptions

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

--ITSPDUHEADER's elements from

--ITS-Container{itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wg1 (1) ts (102894) cdd (2) version (1) } :

IMPORTS

SpeedLimit, TrafficRule, EmergencyPriority, CauseCode, RoadworksSubCauseCode, ClosedLanes, DangerousGoodsBasic, SpecialTransportType, LightBarSirenInUse, EmbarkationStatus, PtActivation, VehicleRole, ExteriorLights, PerformanceClass, CenDsrcTollingZone, LateralAcceleration, VerticalAcceleration, LanePosition, SteeringWheelAngle, YawRate, AccelerationControl, CurvatureCalculationMode, Curvature, LongitudinalAcceleration, VehicleLength, VehicleWidth, Speed, DriveDirection, Heading, Altitude, PosConfidenceEllipse, ReferencePosition, ItsPduHeader, Latitude, Longitude

FROM ITS-Container { itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wg1 (1) ts (102894) cdd (2) version (1) }

BasicVehicleContainerHighFrequency, BasicVehicleContainerLowFrequency, SpecialTransportContainer, PublicTransportContainer, RoadWorksContainerBasic, DangerousGoodsContainer, EmergencyContainer, RescueContainer, SafetyCarContainer

FROM CAM-PDU-Descriptions { itu-t (0) identified-organization (4) etsi (0) itsDomain (5) wg1 (1) en (302637) cam (2) version (1) };

--General Structure of CAM-I:

```
CAMI ::= SEQUENCE {
    header ItsPduHeader,
    cam CoopAwareness
}
```

--ITSPDUHEADER's elements :

--ItsPduHeader ::= SEQUENCE {

--protocolVersion INTEGER{currentVersion(1)} (0..255),

--messageID INTEGER{denm(1), cam(2), poi(3), spat(4), map(5), ivi(6), ev-rsr(7)} (0..255),

--stationID StationID,

--}

--StationID ::= INTEGER(0..4294967295)

--Coop Awareness's elements:

```
CoopAwareness ::= SEQUENCE {
```



```

        generationDeltaTime GenerationDeltaTime,
        camParameters CamParameters
    }

    GenerationDeltaTime ::= INTEGER { oneMilliSec(1) } (0..65535)

--Cam Parameters's elements:

    CamParameters ::= SEQUENCE {
        basicContainer BasicContainer,
        highFrequencyContainer HighFrequencyContainer,
        lowFrequencyContainer LowFrequencyContainer OPTIONAL,
        specialVehicleContainer SpecialVehicleContainer OPTIONAL,
        ...
    }

--Basic Container's elements:

    BasicContainer ::= SEQUENCE {
        stationType StationType,
        referencePosition ReferencePosition,
        ...
    }

    StationType ::= INTEGER {unknown(0), pedestrian(1), cyclist(2), moped(3), motorcycle(4),
passengerCar(5), bus(6),
        lightTruck(7),      heavyTruck(8),      trailer(9),      specialVehicles(10),      tram(11),
roadSideUnit(15)} (0..255)

--ReferencePosition ::= SEQUENCE {
--latitude Latitude,
--longitude Longitude,
--positionConfidenceEllipse PosConfidenceEllipse ,
--altitude Altitude
--}
--Longitude ::= INTEGER {oneMicrodegreeEast (10), oneMicrodegreeWest (-10), --unavailable(1800000001)}
--(-1800000000..1800000001)
--Latitude ::= INTEGER {oneMicrodegreeNorth (10), oneMicrodegreeSouth (-10), --unavailable(900000001)}
--(-900000000..900000001)
--Altitude ::= SEQUENCE {
--altitudeValue AltitudeValue,
--altitudeConfidence AltitudeConfidence
--}
--AltitudeValue ::= INTEGER {referenceEllipsoidSurface(0), oneCentimeter(1), --unavailable(800001)} (-
--100000..800001)
--AltitudeConfidence ::= ENUMERATED {
--alt-000-01 (0),
--alt-000-02 (1),
--alt-000-05 (2),
--alt-000-10 (3),
--alt-000-20 (4),
--alt-000-50 (5),
--alt-001-00 (6),
--alt-002-00 (7),
--alt-005-00 (8),
--alt-010-00 (9),
--alt-020-00 (10),

```

```
--alt-050-00 (11),
--alt-100-00 (12),
--alt-200-00 (13),
--outOfRange (14),
--unavailable (15)
--}
--PosConfidenceEllipse ::= SEQUENCE {
--semiMajorConfidence SemiAxisLength,
--semiMinorConfidence SemiAxisLength,
--semiMajorOrientation HeadingValue
--}
--SemiAxisLength ::= INTEGER {oneCentimeter(1), outOfRange(4094), unavailable(4095)}      --(0..4095)
--HeadingValue ::= INTEGER {wgs84North(0), wgs84East(900), wgs84South(1800), --wgs84West(2700),
--unavailable(3601)} (0..3601)
```

--High Frequency Container's elements:

```
HighFrequencyContainer ::= CHOICE {
    basicVehicleContainerHighFrequency BasicVehicleContainerHighFrequency,
    rsuContainerHighFrequency RSUContainerHighFrequency,
    ...
}
```

--Low Frequency Container's elements:

```
LowFrequencyContainer ::= CHOICE {
    basicVehicleContainerLowFrequency BasicVehicleContainerLowFrequency,
    ...
}
```

--Special Vehicle Container's elements:

```
SpecialVehicleContainer ::= CHOICE {
    publicTransportContainer PublicTransportContainer,
    specialTransportContainer SpecialTransportContainer,
    dangerousGoodsContainer DangerousGoodsContainer,
    roadWorksContainerBasic RoadWorksContainerBasic,
    rescueContainer RescueContainer,
    emergencyContainer EmergencyContainer,
    safetyCarContainer SafetyCarContainer,
    ...
}
```

--RSU Container High Frequency's elements:

```
RSUContainerHighFrequency ::= SEQUENCE {
    protectedCommunicationZonesRSU ProtectedCommunicationZonesRSU OPTIONAL,
    ...,
    serviceAdvertisementContainer ServiceAdvertisementContainer,
    positionEnhancementContainer PositionEnhancementContainer,
    environmentAndContextContainer EnvironmentAndContextContainer
    --environment&ContextContainer Environment&ContextContainer
}
```

--Protected Communication Zones RSU's elements:

ProtectedCommunicationZonesRSU ::= SEQUENCE (SIZE(1..16)) OF ProtectedCommunicationZone

```
ProtectedCommunicationZone ::= SEQUENCE {
    protectedZoneType ProtectedZoneType,
    expiryTime TimestampIts OPTIONAL,
    protectedZoneLatitude Latitude,
    protectedZoneLongitude Longitude,
    protectedZoneRadius ProtectedZoneRadius OPTIONAL,
    protectedZoneID ProtectedZoneID OPTIONAL
}
```

```
ProtectedZoneType ::= ENUMERATED {
    cenDsrcTolling (0),
    ...
}
```

TimestampIts ::= INTEGER {utcStartOf2004(0), oneMillisecAfterUTCStartOf2004(1)}
(0..4398046511103)

ProtectedZoneRadius ::= INTEGER {oneMeter(1)} (1..255,...)

ProtectedZoneID ::= INTEGER (0.. 134217727)

--Service Advertisement Container's elements:

```
ServiceAdvertisementContainer ::= SEQUENCE {
    advertisedServiceId AdvertisedServiceId,
    serviceAccessCapabilities ServiceAccessCapabilities,
    channelUsedByTheAdvertisedService ChannelUsedByTheAdvertisedService,
    communicationProfileUsedForTheService CommunicationProfileUsedForTheService,
    rsuMacAddress RsuMacAddress,
    rsuIpAddress RsuIpAddress
}
```

AdvertisedServiceId ::= INTEGER(0.. 255)

```
ServiceAccessCapabilities ::= BIT STRING {
    globalNetwork (0),
    continuousExchangeCapability (1),
    storeForwardCapability (2)
} (SIZE(8))
```

ChannelUsedByTheAdvertisedService ::= ENUMERATED {cch(0),sch1(1), sch2(2), sch3(3),
sch4(4), sch5(5), sch6(6)}

CommunicationProfileUsedForTheService ::= ENUMERATED {btpgeonet(0),tcpipV4(1),
tcpipV6(2)}

--ChannelUsedByTheAdvertisedService ::= INTEGER {CCH(0),SCH1(1), SCH2(2), SCH3(3), --SCH4(4), SCH5(5), SCH6(6)}
(0..255)

--CommunicationProfileUsedForTheService ::= INTEGER {BTPGEONET(0),TCPIPv4(1), --TCPIPv6(2)} (0..255)

RsuMacAddress ::= OCTET STRING (SIZE(6))

```

RsuIpAddress ::= CHOICE {
    rsuipv4Andv6Address Ipv4Andv6,
    rsuiPv4Address IPv4Address,
    rsuiPv6Address IPv6Address
}
--RsuIpAddress ::= SEQUENCE {
--RsuiPv4Address IPv4Address OPTIONAL,
--RsuiPv6Address IPv6Address OPTIONAL }

Ipv4Andv6 ::= SEQUENCE {
    rsuiPv4Address IPv4Address,
    rsuiPv6Address IPv6Address
}

IPv4Address ::= OCTET STRING (SIZE(4))

IPv6Address ::= OCTET STRING (SIZE(16))

--Position Enhancement Container's elements:

PositionEnhancementContainer ::= SEQUENCE {
    gpsPositionDeltaLatitude DeltaLatitude OPTIONAL,
    gpsPositionDeltaLongitude DeltaLongitude OPTIONAL,
    gpsPositionDeltaAltitude DeltaAltitude OPTIONAL,
    satelliteConstellationLocallyAvailable SatelliteConstellationLocallyAvailable
OPTIONAL,
    tracesLeadingToTheRsu TracesLeadingToTheRsu OPTIONAL
    --tracesLeadingToTheRsu Traces OPTIONAL
}

SatelliteConstellationLocallyAvailable ::= INTEGER(0..255)

TracesLeadingToTheRsu ::= SEQUENCE SIZE(1..7) OF PathHistory

--Traces ::= SEQUENCE SIZE(1..7) OF PathHistory

PathHistory ::= SEQUENCE (SIZE(0..40)) OF PathPoint

PathPoint ::= SEQUENCE {
    pathPosition DeltaReferencePosition,
    pathDeltaTime PathDeltaTime OPTIONAL
}

DeltaReferencePosition ::= SEQUENCE {
    deltaLatitude DeltaLatitude,
    deltaLongitude DeltaLongitude,
    deltaAltitude DeltaAltitude
}

DeltaLongitude ::= INTEGER {oneMicrodegreeEast (10), oneMicrodegreeWest (-10),
unavailable(131072)}(-131071..131072)

DeltaLatitude ::= INTEGER {oneMicrodegreeNorth (10), oneMicrodegreeSouth (-10),
unavailable(131072)} (-131071..131072)

```

```
DeltaAltitude ::= INTEGER {oneCentimeterUp (1), oneCentimeterDown (-1),
unavailable(12800)} (-12700..12800)
```

```
PathDeltaTime ::= INTEGER {tenMilliSecondsInPast(1)} (1..65535, ...)
```

--Environment & Context Container's elements:

```
EnvironmentAndContextContainer ::= SEQUENCE {
    localMeteorologicalData LocalMeteorologicalData OPTIONAL,
    roadEnvironment RoadEnvironment OPTIONAL,
    trafficCondition TrafficCondition OPTIONAL
}
```

```
--Environment&ContextContainer ::= SEQUENCE {
--localMeteorologicalData LocalMeteorologicalData OPTIONAL,
--roadEnvironment RoadEnvironment OPTIONAL,
--trafficCondition TrafficCondition OPTIONAL
--}
```

```
LocalMeteorologicalData ::= BIT STRING (SIZE(8))
```

```
RoadEnvironment ::= INTEGER(0..255)
```

```
TrafficCondition ::= INTEGER(0..255)
```

END

6. NACS: Access control protocol for advertised services

Once the advertisement message CAM-I is intercepted by vehicles in the neighborhood of the ITSS-R (RSU), the vehicle must go through an access control/authorization phase to the network (global or private) before executing the selected service.

This access control protocol is named NACS (Network Access Control for C-ITS Services). The notion of the context of the application of this protocol is very important. In the project SCOOP@F, we recommend to execute this protocol only if we are in a favorable context. Several criteria can be taken into account in the context definition as for example the quality of the link between the vehicle and the ITSS-R (RSU), the signal to noise ratio, the load of used SCHx channels, the trace, the speed, etc. At first, we can simplify the structure of the context to be considered. The vehicle can also send an access request when it is stationary or travelling at low speed in front of an ITSS-R (RSU) station. Therefore, a communication over a single hop will be preferred.

The protocol NACS includes two messages:

- Service Access Request (SAReq)
- Service Access Response (SARep)

Messages exchanged by means of this mechanism can use several communication profiles. The chosen profile is established in agreement between vehicles and ITSS-Rs (RSUs) according to the selected policy by the road operator's network. In the case of SCOOP@F part 1, the preferred communication profile is UDP/TCP, IPv4/IPv6, G5 (SCH1) because we choose a one-hop communication to execute the two considered services (renewal of pseudonym certificates and upload of logs).

- SAReq

Figure 15 gives a description of the structure of the SAReq message. It includes the following fields:

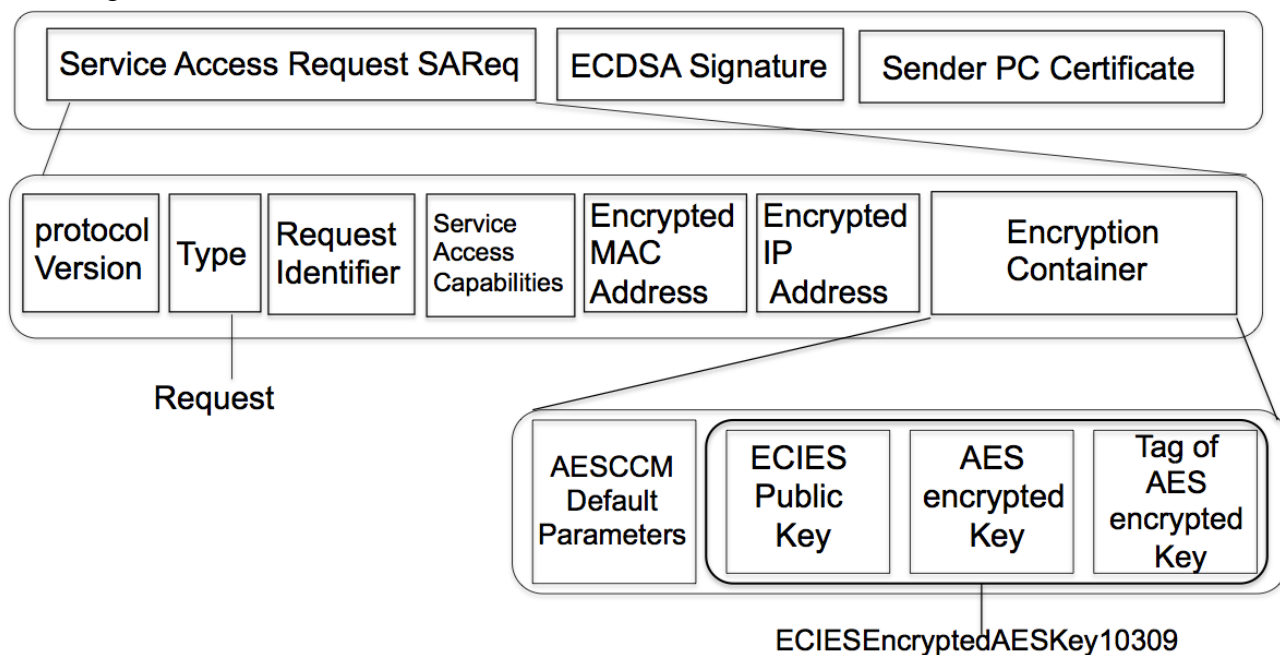


Figure 15: Service Access Request (SAReq) Message structure

- Service Access Request, which contains:
 - PV (1 byte): the first byte indicates the version of the protocol.
 - Type (1 byte): this field indicates the type of the message (0: for a request, 1: for a positive response (access accepted), 2: negative response (access rejected)).
 - ID (1 byte): this field indicates the identifier (sequence number) of the message. The Identifier field is one octet. The Identifier field must be the same if a SReq message is retransmitted while waiting for a Response. Any new (non-retransmission) SReq must modify the Identifier field.
 - SAC (1 byte): this field indicates the wished access capabilities. This field has the same structure as the one included in the CAM-I message.
 - Encrypted Vehicle MAC Address (VM) (22 bytes): this field contains the MAC address of the requestor vehicle.
 - Encrypted Vehicle IP Address (VIP): this field contains the IP address of the vehicle. If IPv4 is supported the field contains the IPv4 address of the vehicle. If IPv6 is supported the field contains the IPv6 address of the vehicle. According to the chosen IP address the field's size could be of 20, 32 or 52 bytes.
 - Encryption Container: this field contains data considered in encryption process :
 - AESCCMDefaultParameters: this field contains the AES encryption nonce (12 bytes).
 - ECIESEncryptedAESKey: this field contains (1) the ECIES public key (33 bytes) used for the encryption of the AES key, (2) the encrypted AES Key (16 bytes) and (3) the Tag of the AES encrypted Key (16 bytes).

SReq message is signed. Consequently, the SReq message contains signature and sender certificate fields. Vehicle MAC Address and Vehicle IP Address are encrypted using AES-128-CCM algorithm. An encrypted AES Key will be used by the algorithm AES-128-CCM. This key will be encrypted using ECIES algorithm. An ECIES public key EPK is used.

The ASN1 structure of SReq is described as follows (DER encoding):

```
SReq DEFINITIONS AUTOMATIC TAGS ::= BEGIN
```

```
MessageSReq ::= SEQUENCE {
    sareq SReq,
    signature Signature,
    certificate OCTET STRING
}
```

```
SReq ::= SEQUENCE {
    protocolVersion PV,
    type Type,
    id ID,
    serviceAccessCapabilities ServiceAccessCapabilities,
    vehicleMacAddress VehicleMacAddress,
    vehicleIpAddress VehicleIpAddress,
    encryption Encryption
}
```

```

Signature ::= SEQUENCE {
    signatureAlgorithm OBJECT IDENTIFIER DEFAULT { ecdsa-with-SHA256 },
    signatureValue SignatureValue
}

ecdsa-with-SHA256 OBJECT IDENTIFIER ::= {iso(1) member-body(2) us(840) ansi-X9-
62(10045) signatures(4) ecdsa-with-SHA2(3) 2 }

SignatureValue ::= OCTET STRING

PV ::= INTEGER {currentVersion(1)} (0..255)

Type ::= ENUMERATED {
    request(0)
    --accessAccepted(1),
    --accessRejected(2)
}

ID ::= INTEGER (0..255)

ServiceAccessCapabilities ::= BIT STRING {
    globalNetwork (0),
    continuousExchangeCapability (1),
    storeForwardCapability (2)
} (SIZE(8))

--AESCCM encryption: EncryptedMessage has n bytes (Message has n bytes) and a TAG of
16 bytes

VehicleMacAddress ::= OCTET STRING (SIZE(22)) --TAG + 16 bytes

VehicleIpAddress ::= CHOICE {
    vehicleIpv4Andv6Address Ipv4Andv6,
    vehicleIPv4Address IPv4Address,
    vehicleIPv6Address IPv6Address
}

Ipv4Andv6 ::= SEQUENCE {
    vehicleIPv4Address IPv4Address,
    vehicleIPv6Address IPv6Address
}

IPv4Address ::= OCTET STRING (SIZE (20)) --TAG + 16 bytes

IPv6Address ::= OCTET STRING (SIZE(32)) --TAG + 16 bytes

Encryption ::= SEQUENCE {
    aes-nonce AESCCMDefaultParameters,
    encryptedKeyMaterial ECIESEncryptedAESKey103097
}

AESCCMDefaultParameters ::= OCTET STRING (SIZE(12))

```



```
ECIESEncryptedAESKey103097 ::= SEQUENCE {
    v PK,
    c OCTET STRING (SIZE(16)),
    t OCTET STRING (SIZE(16))
}
```

```
PK ::= SEQUENCE {
    type ECCPublicKeyType,
    x INTEGER
}
```

```
ECCPublicKeyType ::= ENUMERATED {
    compressed-lsb-y-0 (2),
    compressed-lsb-y-1 (3)
}
```

END

• SARep

An answer to SAReq request is sent to the vehicle through the SARep message. It uses the same communication profile as the demand access request. Its structure is presented in Figure 16; it is composed of the following fields:

- Service Access Response, which contains:
 - PV: the first byte indicates the version of the protocol.
 - Type: the Type field is one byte. It indicates the type of the message (1: for a positive response (access accepted), 2: negative response (access rejected), 3: access rejected with reason 1, 4: access rejected with reason 2, etc).
 - ID: The Identifier field is one octet and aids in matching requests to responses. The Identifier field must match the Identifier field of SARep that it is sent in response to SAReq.

SARep message is signed. Consequently, the SARep message contains signature and sender certificate fields.

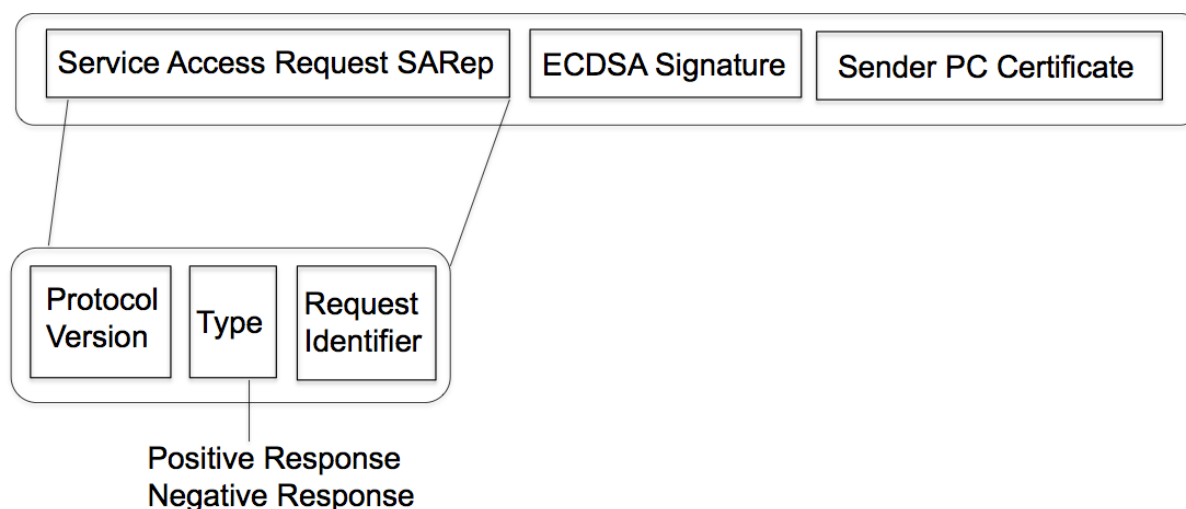


Figure 16: Service Access Response (SARep) Message structure

SARep ASN1 structure is described as follows (DER encoding) :

SARep DEFINITIONS AUTOMATIC TAGS ::= BEGIN

```
MessageSARep ::= SEQUENCE {
    sarep SARep,
    signature Signature,
    certificate OCTET STRING
}
```

```
SARep ::= SEQUENCE {
    protocolVersion PV,
    type Type,
    id ID
}
```

```
Signature ::= SEQUENCE {
    signatureAlgorithm OBJECT IDENTIFIER DEFAULT { ecdsa-with-SHA256 },
    signaturevalue SignatureValue
}
```

```
ecdsa-with-SHA256 OBJECT IDENTIFIER ::= {iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) 2 }
```

```
SignatureValue ::= OCTET STRING
```

```
PV ::= INTEGER {currentVersion(1)} (0..255)
```

```
Type ::= ENUMERATED {
    --request(0)
    accessAccepted(1),
    accessRejected(2)
}
```

```
ID ::= INTEGER (0..255)
```

END

SAReq and SARep messages are defined at Application Layer and signed following the same principle used to sign CAM and DENM messages.

The communication profile selected for SCOOP@F is given in deliverable 2.4.4-8.

Initially, NACS protocol has been defined to be generic applicable for different services and different contexts in order to be flexible to meet C-ITS services evolution. However, NACS is not applied in SCOOP for optimization since PKI requests and logs upload are encrypted and authenticated.

7. Services

In SCOOP@F part 1, we deal with two services:

- The service of renewal of certificates through ITSS-Rs (RSUs) that forward the associated requests to the PKI servers.
- The service of upload of logs (T-Logs and U-Logs) through RSUs which take care to upload Logs.

MAC address and pseudonym certificates change must not occur during communication sessions in particular PKI requests, logs upload and DENM transmission.

7.1 Renewal of pseudonym certificates

The proposed mechanism as described by Figure 17 is defined at application layer and is based on a client-server protocol. Requests towards PKI servers and their associated responses sent from PKI servers use HTTP protocol (see details in the available PKI specification deliverable 2.4.4-6). The preferred communication profile is TCP, IPv4/IPv6, G5 (SCH1). The figure below shows the pseudo-algorithm of the service execution for the renewal of one pseudonym certificate including exchanges between the vehicle, the station ITSS-R (RSU) and PKI entities PCA and LTCA.

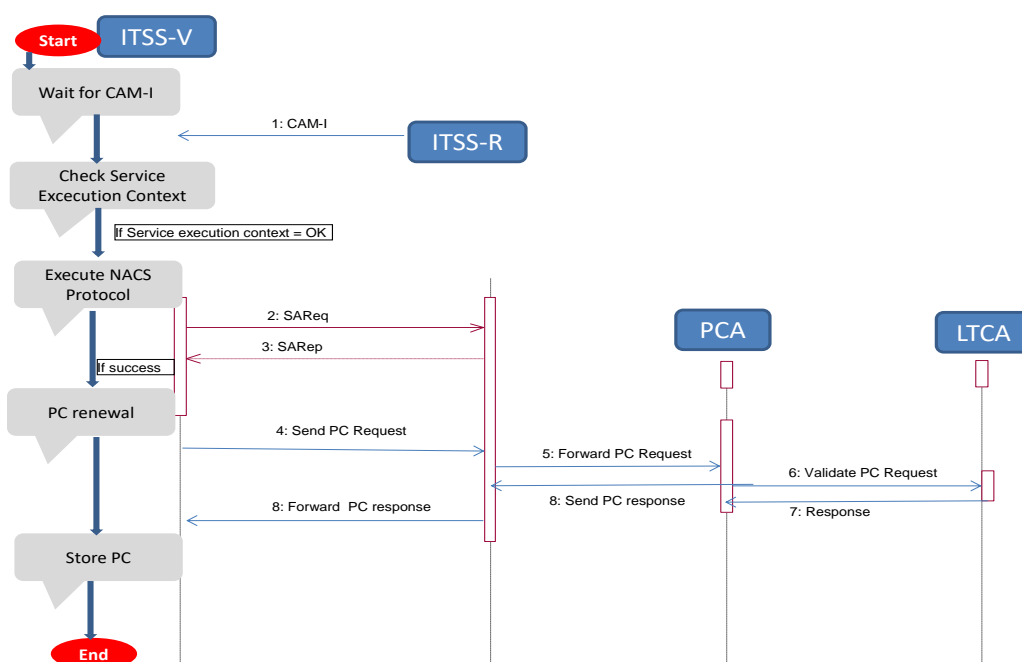


Figure 17: Renewal pseudonym certificates service

The communication profile selected for SCOOP@F and more details on execution service context are given in deliverable 2.4.4-8. Thresholds and some implementation details are given in Deliverable 2.4.4-8. Application of this protocol is clarified in deliverable 2.4.4-8 based on the type of the initiator station type (manufacturer vehicle, operator vehicle or RSU).

NACS protocol is not applied in SCOOP for optimization since PKI requests are encrypted and authenticated.

7.2 Upload of Logs

To transmit T-logs and U-logs to the Logs's server, we use the protocol SFTP (SSH File Transfer Protocol or FTP over SSH) at the application layer. The used communication profile is: TCP, IPv4/IPv6, G5(SCH1). Fragmentation is applied at TCP layer. The format and the size of logs's files are defined in Deliverable 2.4.1.3bis [5]. A pair of (login, password) is used to secure SFTP communications.

SSH File Transfer Protocol (SFTP)

Also called Secure File Transfer Protocol or FTP over SSH, not to confuse with FTP over SSL/TLS (FTPS) described by IETF RFC 4217 or with Simple File Transfer Protocol described by IETF RFC 913. SFTP represents a network protocol that provides file access, file transfer and file management over any reliable, bidirectional octet stream. The last published version is Version 6, described and enhanced in the IETF draft describing the protocol [6]. SFTP is an extension of SSH protocol Version 2.0 described by RFC 4251. Thus, SFTP's authentication and security are provided by its underlying protocol SSH (port 22). Encryption is ensured by a combination of asymmetric algorithms (RSA) and symmetric ones (DES, 3DES, AES...). Authentication is provided by: (1) SSH keys (public and private keys) or (2) by login password authentication.

The communication profile selected for SCOOP@F and some technical details (version of SFTP, login and password) are given in deliverable 2.4.4-8.

NACS protocol is not applied in SCOOP for optimization since logs upload requests are secured.

8. Conclusion

This deliverable defines generic mechanisms to be used to execute renewal pseudonym certificates and logs upload services. These mechanisms include: (1) service advertisement through the broadcasting of the CAM-I message, (2) NACS access control protocol and (3) execution service procedures. We can consider that the deliverable 2.4.1.1 contains the first versions of the specified protocols. These protocols could be improved during SCOOP@F part 2. Implementation details related to these services are given in Deliverable 2.4.4-8.

9. References

[1] Deliverable 2.4.4-6

[2] Deliverable 2.3.1.1

[3] ETSI EN 302 665: "Intelligent Transport Systems (ITS); Communications Architecture".

[4] IEEE 1609.3 2010: "Wireless Access in Vehicular Environments (WAVE) - Networking Services".

And also:

- ETSI TS 102 890-2: "Intelligent Transport Systems (ITS); Facilities layer function; Services announcement specification".
- ETSI TS 102 723-7: "Intelligent Transport Systems; OSI cross-layer topics; Part 7: Interface between security entity and access layer".
- ETSI TS 102 723-8: "Intelligent Transport Systems; OSI cross-layer topics; Part 8: Interface between security entity and network and transport layers".
- ETSI TS 102 723-9: "Intelligent Transport Systems; OSI cross-layer topics; Part 9: Interface between security entity and facilities layer".

[5] Deliverable 2.4.1.3bis

[6] SSH File Transfer Protocol, draft-ietf-secsh-filexfer-13.txt,
<https://tools.ietf.org/html/draft-ietf-secsh-filexfer-13>.