|  |
| --- |
| **CHANGE REQUEST** |
|  | ETSI TS 102 941 | **Version** | 2.1.1 | **CR** | 3 | **rev** | 1- |  |
|  |
| **CR Title** | Adding X.509 Enrolment Credentials |
|  |  |
| **Original Source** | ITS WG5 |
|  |  |
| **Work Item Ref** |  | **Submission date** |  |
| **Approving TB**  | ITS | **Approval date** |  |
| **Category:** | **B** | **Release** |  |  |
|  | Use **one** of the following categories:**F** (correction)**A** (correction in an earlier release)**B** (addition of feature) **C** (functional modification of feature)**D** (editorial modification) |  |
|  |  |
| **Reason for change** | For harmonization with IEEE 1609.2.1, the enrolment credential shall also allow x.509 certificate types. |
|  |  |
| **Consequence if not approved** | Manufacturers must implement additional protocols instead of relying on existing infrastructure. |
|  |  |
| **Summary of change** | Adding X.509 Enrolment Credentials |
|  |  |
| **Clauses affected** | 6.1.3, 6.2.3.5.2, A.2.2, A.2.3, C.0 |
|  |  |
| **Linked Change Requests** |  |  |
|  |  |  |
|  |  |
| **Other comments** |  |
|  |  |

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non‑specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <https://docbox.etsi.org/Reference>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 102 731: "Intelligent Transport Systems (ITS); Security; Security Services and Architecture".

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

[3] ETSI TS 103 097: "Intelligent Transport Systems (ITS); Security; Security header and certificate formats".

[4] ETSI TS 102 942: "Intelligent Transport Systems (ITS); Security; Access control".

[5] ETSI TS 102 940: "Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management".

[6] ISO/IEC 8824-1:2015: "Information technology -- Abstract Syntax Notation One (ASN.1): Specification of basic notation".

[7] Recommendation ITU-T X.696 (08/2014): "Information Technology-Specification of Octet Encoding Rules (OER)".

[8] Void.

[9] ETSI TS 102 943: "Intelligent Transport Systems (ITS); Security; Confidentiality services".

[10] ETSI EN 302 637-2: "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service".

[11] ETSI EN 302 637-3: "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service".

[12] ETSI TS 103 301: "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Facilities layer protocols and communication requirements for infrastructure services".

[13] NIST FIPS PUB 198-1: "The Keyed-Hash Message Authentication Code (HMAC)".

[14] Void.

[15] IETF RFC 4862: "IPv6 Stateless Address Autoconfiguration".

[16] ETSI EN 302 636-6-1: "Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 6: Internet Integration; Sub-part 1: Transmission of IPv6 Packets over GeoNetworking Protocols".

[17] Void.

[18] ETSI EN 302 636-4-1: "Intelligent Transport Systems (ITS); Vehicular communications; GeoNetworking; Part 4: Geographical addressing and forwarding for point-to-point and point-to-multipoint communications; Sub-part 1: Media-Independent Functionality".

[19] ETSI TS 102 965: "Intelligent Transport Systems (ITS); Application Object Identifier (ITS-AID); Registration".

[20] IEEE 802.11™: "IEEE Standard for Information technology -- Telecommunications and information exchange between systems -- Local and metropolitan area networks-Specific requirements -- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications".

[21] IETF RFC 5280. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Available from <https://tools.ietf.org/html/rfc5280>.

[22] IETF RFC 5480. Elliptic Curve Cryptography Subject Public Key Information. Available from <https://tools.ietf.org/html/rfc5480>.

[23] IEEE Std 1609.2™-2016: "IEEE Standard for Wireless Access in Vehicular Environments --Security Services for Applications and Management Messages", as amended by IEEE Std 1609.2a™-2017: "Standard for Wireless Access In Vehicular Environments – Security Services for Applications and Management Messages Amendment 1".

[24] IEEE Std. 1609.2.1: “Standard for Wireless Access in Vehicular Environments (WAVE) – Certificate Management Interfaces for End Entities”

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non‑specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ISO/IEC 15408-2: "Information technology - Security techniques - Evaluation criteria for IT security; Part 2: Security functional components".

[i.2] ETSI TR 102 638: "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions".

[i.3] IETF RFC 4046: "Multicast Security (MSEC) Group Key Management Architecture".

[i.4] IETF RFC 4301: "Security Architecture for the Internet Protocol".

[i.5] IETF RFC 4302: "IP Authentication Header".

[i.6] IETF RFC 4303: "IP Encapsulating Security Payload (ESP)".

[i.7] IETF RFC 5246: "The Transport Layer Security (TLS) Protocol Version 1.2".

[i.8] IETF RFC 3547: "The Group Domain of Interpretation".

[i.9] IETF RFC 3830: "MIKEY: Multimedia Internet KEYing".

[i.10] IETF RFC 4535: "GSAKMP: Group Secure Association Key Management Protocol".

[i.11] IETF RFC 4306: "Internet Key Exchange (IKEv2) Protocol", December 2005.

[i.12] IETF RFC 4877: "Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture".

[i.13] ETSI TS 102 723-8: "Intelligent Transport Systems (ITS); OSI cross-layer topics; Part 8: Interface between security entity and network and transport layer".

[i.14] CVRIA: "Connected Vehicle Reference Implementation Architecture".

NOTE: Available at <http://www.iteris.com/cvria/>.

[i.15] ISO 21210-2010: "Intelligent Transport Systems (ITS) - Communications access for land mobiles (CALM) - Ipv6 networking".

6.1.3 Enrolment

There are two types of enrolment certificate that can be used to authenticate requests for authorization tickets:

* ETSI TS 102 941 enrolment credentials following the specification from this standard. The provisioning of this type is described in this standard.
* X.509 enrolment credentials following the X.509 format which is specified in RFC 5280 [21] and RFC 5480 [22]. The provisioning of this type is out-of-scope of this standard since mechanisms already exist. The standard will explicitly refer to X.509 enrolment credentials when this type is meant.

The following process applies to the provisioning of ETSI TS 102 941 enrolment credentials.

The ITS-S requests its enrolment certificate from the EA (see clause 6.2.3.2).

When an end entity applies for an enrolment certificate, it may indicate that it is entitled to the certificate *directly* or *indirectly*. Compare with Section 4.1.4.2 of IEEE 1609.2.1 [24].

In the *indirect* initial enrolment case, the Enrolment Authority (EA) is given assurance that the end entity is entitled to the enrolment certificate by means outside the scope of the enrolment certificate request. For example, the enrolment request could happen in an environment that is trusted by the EA and the request could be transmitted over a secure connection.

In the *direct* initial enrolment case, before enrolment happens, the end entity generates a keypair known as the canonical keypair and is assigned (or generates) a globally unique identifier known as the canonical identity (cf. Section 6.1.2).

The state transitions for enrolment are shown in Figure 2.

****

**Figure 2: Simplified state machine for the enrolment process**

After a successful enrolment process, the ITS-S shall possess an enrolment credential that shall be used in subsequent authorization requests.

For renewing the Enrolment Certificate at the EA, the ITS-S shall send an EnrolmentRequest signed by the previous valid enrolment credential issued by this EA.

6.1.3.1 Enrolment Credential Profile

The profile for the ETSI enrolment credential is defined in ETSI TS 103 097 along with the ETSI ITS security header and certificate format [3].

Since the format of the X.509 enrolment credential is not defined in ETSI TS 103 097 but used for trust and privacy management in this standard, the profile for X.509 enrolment credentials is specified in this document.

The X.509 enrolment credential shall apply the following conditions. Additional conditions may be required by the individual X.509 PKI in alignment with the Enrolment Authority.

* + - * 1. Generic Requirements

The certificate shall follow the IETF PKIX profile as specified in RFC 5280 [21]. When ECDSA keys are used, the encoding shall follow the specification in RFC 5480 [22]. The cryptographic algorithms shall be aligned with ETSI TS 103 097 [3].

6.1.3.1.2 Version

The version shall be V3 (defined by the integer value 2).

6.1.3.1.3 Issuer

The identity of the issuer shall contain at least the following attributes:

* countryName;
* organizationName; and
* commonName.

Each attribute shall be limited to a single instance of the attribute. Additional attributes may be present.

The countryName attribute shall specify the country in which the issuer of the certificate is established.

The organizationName attribute shall contain the full registered name of the certificate issuing organization.

The commonName attribute value shall contain a name commonly used by the subject to represent itself. This name

need not be an exact match of the fully registered organization name.

6.1.3.1.4 Subject

The subject field shall include at least the following attributes:

* countryName;
* organizationName; and
* commonName.

Only one instance of each of these attributes shall be present. Additional attributes may be present.

The countryName attribute shall specify the country in which the ITS-S is established.

The organizationName attribute shall contain the full registered name of the ITS-S.

The commonName attribute value shall contain a name commonly used by the subject to represent itself and may correspond to the canonical ID.

6.1.3.1.5 Authority key identifier extension

The authority key identifier extension shall be present, containing a key identifier for the issuing CA's public key.

6.1.3.1.6 Key usage extension

The key usage extension shall be present and shall contain the key usage settings specifying at least non-repudation (bit 1) and digital signature (bit 0).

6.1.3.1.7 Certificate policies extension

This extension should not be marked critical. The certificate policies extension shall be present and shall contain the identifier of at least one certificate policy which reflects the practices and procedures undertaken by the CA

6.1.3.1.8 CRL distribution points extension

If CRL is supported by the issuing CA, the CRL distribution point extension shall be present in certificates.

If the certificate does not include any access location of an OCSP responder, then the certificate shall include a CRL distribution point extension.

When present, the CRL distribution point extension shall include at least one reference to a publicly available CRL.

At least one of the present references shall use either http (http://) or ldap (ldap://) scheme.

The extension shall not be marked critical.

6.1.3.1.9 Authority Information Access extension

The Authority Information Access extension shall be present.

When OCSP is supported by the issuing CA, the Authority Information Access extension shall include an accessMethod OID, id-ad-ocsp, with an accessLocation value specifying at least one access location of an

OCSP responder authoritative to provide certificate status information for the present certificate. In such case, at least one access location shall specify either the http (http://)] or https (https://) scheme. Such access location shall reference a publicly available OCSP responder, which accepts unsigned and unauthenticated status requests.

A reference to at least one OCSP responder shall be present if the certificate does not include any CRL distribution point extension.

6.2.3.5.2 Butterfly Authorization request

The following functional requirements are defined on the state machine of Figure 3 (sender ITS-S for the authorization process):

The ButterflyAuthorizationRequest message and X509SignedButterflyAuthorizationRequest message shall be encrypted using an ETSI TS 103 097 [3] approved encryption algorithm and the public key from the certificate of the Enrolment Authority. The EA certificate may be injected during manufacture or can be obtained from the CTL.

For each butterfly authorization request, the ITS-S shall generate a new caterpillar key pair, used for the expansion to individual verification key paris, corresponding to an approved signature algorithm as specified in ETSI TS 103 097 [3].

The contents of the ButterflyAuthorizationRequest message shall be as described in Figure 24.

The complete nested data structure of the ButterflyAuthorizationRequest message is specified in Figure 24. The specification of the content of this message using ASN.1 [6], [7] shall be as specified in clause A.2.

**.**

NOTE: Encryption is done with ECIES using the public encryption key of the EA. The signature is computed using currently valid private key corresponding to the EC's verification public key.

**Figure 24: ButterflyAuthorizationRequest message**

To create a butterfly authorization request, the ITS-S shall follow this process:

An ECC private key is randomly generated, the corresponding public key (verifyKeyIndicator) is to be used as caterpillar key for the butterfly key expansion.

A ToBeSignedCertificate structure is built, with:

The id being set to none;

the cracaId being set to '000000'H;

the crlSeries being set to 0;

the validityPeriod shall specify the validity of the first AT batch;

the geographicRegion can be optionally included.

the appPermissions containing the requested permissions for the Authorization Tickets,

the verifyKeyIndicator shall contain the generated caterpillar key

NOTE: The butterfly authorization request does not support the provisioning of ATs with encryption keys.

An EeRaCertRequest structure is built, with:

The version being set to 2,

The generationTime being set to the generation time of this structure,

The certificateType being set to explicit,

The tbsCert shall contain the ToBeSignedCertificate structure that was generated previously,

The additionalParams structure shall contain either of the following

the original option with the ButterflyParamsOriginal containing a signingExpansion containing a freshly generated 16 Byte string to be used as a key for the expansion function for signing , an encryptionKey containing the caterpillar public key for encryption, and an encryptionExpansion containing a freshly generated 16 Byte string to be used as a key for the expansion function for encryption.

the unified option with the ButterflyExpansion containing a freshly generated 16 Byte string that shall be used as key for the butterfly expansion.

An EtsiTs102941Data structure is built, with:

the version set to v1 (integer value set to 1);

the content set to the previous data structure (EeRaCertRequest).

An EtsiTs103097Data-Signed structure is built containing: hashId, tbsData, signer and signature:

the hashId shall indicate the hash algorithm to be used as specified in ETSI TS 103 097 [3];

in the tbsData:

the payload shall contain the EeRaCertRequest structure;

in the headerInfo:

the psid shall be set to "secured certificate request" as assigned in ETSI TS 102 965 [19];

the generationTime shall be present;

all other components of the component tbsdata.headerInfo not used and absent;

the signer declared as a digest referencing the hashedId8 of the EC certificate;

the signature over tbsData computed using the private key corresponding to the EC's verification public key.

An EtsiTs103097Data-Encrypted structure is built, with:

the component recipients containing one instance of RecipientInfo of choice certRecipInfo, containing:

the hashedId8 of the EA certificate in recipientId; and

the encrypted data encryption key in encKey, the public key to use for encryption is the encryptionKey found in the EA certificate referenced in recipientId;

the component ciphertext containing the encrypted representation of the previous EtsiTs103097Data-Signed structure.

NOTE: The EA may provide out-of-band means for the ITS-S to specify its preferences for which AA or AAs should be used as well as additional parameters, e.g., the version used for the certificate format specification.

The contents of the X509SignedButterflyAuthorizationRequest message is illustrated in Figure 25.

The complete nested data structure of the X509SignedButterflyAuthorizationRequest message is specified in Figure 25. The specification of the content of this message using ASN.1 [6], [7] shall be as specified in clause A.2.



NOTE: Encryption is done with ECIES using the public encryption key of the EA. The signature is computed using currently valid private key corresponding to the X.509 EC's verification public key.

**Figure 25: X509SignedButterflyAuthorizationRequestMessage**

All contents of the EeRaCertRequest shall be built as described above for the ButterflyAuthorizationRequestMessage.

A ScmsPdu-Scoped structure is built, with:

the version set to v2 (integer value set to 2);

the content set to a EeRaInterfacePdu which contains the previous data structure (EeRaCertRequest).

A SignedX509CertificateRequest structure is built containing: hashAlgorithmId, tbsRequest, signer, and signature:

the hashAlgorithmId shall indicate the hash algorithm to be used as specified in ETSI TS 103 097 [3];

the tbsRequest shall contain the previous structure (ScmsPdu-Scoped);

the signer declared as a SignerSingleX509Cert containing the X.509 EC; X.509 certificates are encoded with the ASN.1 Distinguished Encoding Rules and cannot be “directly” imported into these structures.

the signature over tbsData computed using the private key corresponding to the X.509 EC's verification public key. The signature is generated on the hash of this structure, obtained per the rules specified for hashing data objects in clause 5.3.1 of IEEE Std 1609.2a-2017 [23], with the parameter Data Input equal to the C-OER encoding of tbsRequest, and the parameter Signer Identifier Input equal to the signer’s certificate, i.e., the X.509 certificate contained in the OCTET STRING indicated by the first X509Certificate in signer.

An Ieee1609Dot2Data-SignedX509AuthenticatedCertRequest structure is built containing protocolVersion and content:

 the version set to v2 (integer value set to 2);

 the content set to a signedX509CertificateRequest which contains the previous data structure (SignedX509CertificateRequest).

An EtsiTs103097Data-Encrypted-Unicast structure is built, with:

the component recipients containing one instance of RecipientInfo of choice certRecipInfo, containing:

the hashedId8 of the EA certificate in recipientId; and

the encrypted data encryption key in encKey, the public key to use for encryption is the encryptionKey found in the EA certificate referenced in recipientId;

the component ciphertext containing the encrypted representation of the previous Ieee1609Dot2Data-SignedX509AuthenticatedCertRequest structure.

NOTE: The EA may provide out-of-band means for the ITS-S to specify its preferences for which AA or AAs should be used as well as additional parameters, e.g., the version used for the certificate format specification.

A.2.2 Security Management messages for CA

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 This file contains the EtsiTs102941Messages module containing all possible PKI messages.

 It should be used when all PKI messages needs to be implemented (for example, for CA development)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

EtsiTs102941MessagesCa

 { itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg5(5) ts(102941) messagesCa(0) version2(2)}

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS

EtsiTs103097Data-Signed,

--EtsiTs103097Data-Encrypted,

EtsiTs103097Data-SignedExternalPayload

--EtsiTs103097Data-SignedAndEncrypted

FROM EtsiTs103097Module

{ itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg5(5) ts(103097) securedMessageV1(0)}

Version,

EtsiTs103097Data-Encrypted-Unicast,

EtsiTs103097Data-SignedAndEncrypted-Unicast

FROM EtsiTs102941BaseTypes

{ itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg5(5) ts(102941) baseTypes(3) version2(2) }

InnerEcRequestSignedForPop, InnerEcResponse

FROM EtsiTs102941TypesEnrolment

{ itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg5(5) ts(102941) enrolment(4) version2(2) }

InnerAtRequest, InnerAtResponse

FROM EtsiTs102941TypesAuthorization

{ itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg5(5) ts(102941) authorization(5) version2(2) }

ToBeSignedCrl, ToBeSignedTlmCtl, ToBeSignedRcaCtl

FROM EtsiTs102941TrustLists

{ itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg5(5) ts(102941) trustLists(6) version2(2) }

AuthorizationValidationRequest, AuthorizationValidationResponse

FROM EtsiTs102941TypesAuthorizationValidation

{ itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg5(5) ts(102941) authValidation(7) version2(2) }

EeRaInterfacePdu, EeRaCertRequest, RaEeCertInfo, EeRaDownloadRequest

FROM Ieee1609Dot2Dot1EeRaInterface

{iso(1) identified-organization(3) ieee(111) standards-association-numbered-series-standards(2) wave-stds(1609) dot2(2) extension-standards(255) dot1(1) interfaces(1) ee-ra(11) major-version-2(2)}

RaAcaCertRequest, AcaRaCertResponse

FROM Ieee1609Dot2Dot1AcaRaInterface

{iso(1) identified-organization(3) ieee(111) standards-association-numbered-series-standards(2) wave-stds(1609) dot2(2) extension-standards(255) dot1(1) interfaces(1) aca-ra(4) major-version-2(2)}

Ieee1609Dot2Data-SignedX509AuthenticatedCertRequest, ScmsPdu-Scoped, SignerSingleX509Cert

FROM Ieee1609Dot2Dot1Protocol

{iso(1) identified-organization(3) ieee(111) standards-association-numbered-series-standards(2) wave-stds(1609) dot2(2) extension-standards(255) dot1(1) interfaces(1) protocol(17) major-version-2(2)}

CaCertificateRequest

FROM EtsiTs102941TypesCaManagement

{ itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg5(5) ts(102941) caManagement(8) version2(2) }

;

/\*\*\*\*\*\*\*\*\*\*\*\*

-- Messages

\*\*\*\*\*\*\*\*\*\*\*\*/

EnrolmentRequestMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{enrolmentRequest PRESENT})})}

EnrolmentResponseMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{enrolmentResponse PRESENT})})}

AuthorizationRequestMessage ::= EtsiTs103097Data-Encrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{authorizationRequest PRESENT})})}

AuthorizationRequestMessageWithPop ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{authorizationRequest PRESENT})})}

AuthorizationResponseMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{authorizationResponse PRESENT})})}

CertificateRevocationListMessage ::= EtsiTs103097Data-Signed{EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{certificateRevocationList PRESENT})})}

TlmCertificateTrustListMessage ::= EtsiTs103097Data-Signed{EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{certificateTrustListTlm PRESENT})})}

RcaCertificateTrustListMessage ::= EtsiTs103097Data-Signed{EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{certificateTrustListRca PRESENT})})}

AuthorizationValidationRequestMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{authorizationValidationRequest PRESENT})})}

AuthorizationValidationResponseMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{authorizationValidationResponse PRESENT})})}

CaCertificateRequestMessage ::= EtsiTs103097Data-Signed {EtsiTs102941Data(WITH COMPONENTS{..., content (WITH COMPONENTS{caCertificateRequest PRESENT})})}

CaCertificateRekeyingMessage ::= EtsiTs103097Data-Signed {EtsiTs103097Data-Signed {EtsiTs102941Data(WITH COMPONENTS{..., content (WITH COMPONENTS{caCertificateRequest PRESENT})})}}

ButterflyAuthorizationRequestMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{butterflyAuthorizationRequest PRESENT})})} X509SignedButterflyAuthorizationRequestMessage ::= EtsiTs103097Data-Encrypted-Unicast {EtsiTs102941ButterflyAuthorizationRequest-X509Signed}

ButterflyAuthorizationResponseMessage ::= EtsiTs103097Data-Signed {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{butterflyAuthorizationResponse PRESENT})})}

ButterflyAtDownloadRequestMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{butterflyAtDownloadRequest PRESENT})})}

ButterflyCertRequestMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{butterflyCertificateRequest PRESENT})})}

ButterflyCertResponseMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{butterflyCertificateResponse PRESENT})})}

/\*\*\*\*\*\*\*\*\*\*\*\*

-- EtsiTs102941Data

\*\*\*\*\*\*\*\*\*\*\*\*/

EtsiTs102941Data::= SEQUENCE {

 version Version (v1),

 content EtsiTs102941DataContent

 }

EtsiTs102941DataContent ::= CHOICE {

 enrolmentRequest InnerEcRequestSignedForPop,

 enrolmentResponse InnerEcResponse,

 authorizationRequest InnerAtRequest,

 authorizationResponse InnerAtResponse,

 certificateRevocationList ToBeSignedCrl,

 certificateTrustListTlm ToBeSignedTlmCtl,

 certificateTrustListRca ToBeSignedRcaCtl,

 authorizationValidationRequest AuthorizationValidationRequest,

 authorizationValidationResponse AuthorizationValidationResponse,

 caCertificateRequest CaCertificateRequest,

 ...,

 /\* Extension for link certificates \*/

 linkCertificateTlm ToBeSignedLinkCertificateTlm,

 singleSignedLinkCertificateRca ToBeSignedLinkCertificateRca,

 doubleSignedlinkCertificateRca RcaSingleSignedLinkCertificateMessage,

 /\* Extension for butterfly key provisioning \*/

 [[butterflyAuthorizationRequest EeRaCertRequest,

 butterflyAuthorizationResponse RaEeCertInfo,

 butterflyCertificateRequest RaAcaCertRequest,

 butterflyCertificateResponse AcaRaCertResponse,

 butterflyAtDownloadRequest EeRaDownloadRequest]]

 }

/\*\*\*\*\*\*\*\*\*\*\*\*

-- X509 structures

\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*

 \* @class EtsiTs102941ButterflyAuthorizationRequest-X509Signed

 \*

 \* @brief This structure defines a butterfly authorization request,

 \* signed with an X.509 certificate, by redefining structures from

 \* IEEE 1609.2.1 to fit into the ETSI naming schema.

 \*/

EtsiTs102941ButterflyAuthorizationRequest-X509Signed ::= Ieee1609Dot2Data-SignedX509AuthenticatedCertRequest {

 ScmsPdu-Scoped {

 EeRaInterfacePdu (WITH COMPONENTS {

 eeRaCertRequest})

 },

 SignerSingleX509Cert

}

END

A.2.3 Security Management messages for ITS-S\_WithPrivacy

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

 This file contains the EtsiTs102941MessagesItss module providing the ITS-S subset

 of messages defined in the module EtsiTs102941MessagesCA

 It should never be imported together with the module EtsiTs102941MessagesCA.

 Use the EtsiTs102941MessagesCA if all possible PKI message types are needed.

 This module blocks the usage of unencrypted EC signature for AA requests.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

EtsiTs102941MessagesItss

 { itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg5(5) ts(102941) messagesItss(1) version2(2)}

DEFINITIONS AUTOMATIC TAGS ::=

BEGIN

IMPORTS

EtsiTs103097Data-Signed

--EtsiTs103097Data-Encrypted,

--EtsiTs103097Data-SignedAndEncrypted

FROM EtsiTs103097Module

{ itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg5(5) ts(103097) securedMessageV1(0) }

EtsiTs103097Data-Encrypted-Unicast,

EtsiTs103097Data-SignedAndEncrypted-Unicast,

Version

FROM EtsiTs102941BaseTypes

{ itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg5(5) ts(102941) baseTypes(3) version2(2) }

InnerEcRequestSignedForPop, InnerEcResponse

FROM EtsiTs102941TypesEnrolment

{ itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg5(5) ts(102941) enrolment(4) version2(2) }

InnerAtRequest, InnerAtResponse

FROM EtsiTs102941TypesAuthorization

{ itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg5(5) ts(102941) authorization(5) version2(2) }

ToBeSignedCrl, ToBeSignedTlmCtl, ToBeSignedRcaCtl

FROM EtsiTs102941TrustLists

{ itu-t(0) identified-organization(4) etsi(0) itsDomain(5) wg5(5) ts(102941) trustLists(6) version2(2) }

EeRaInterfacePdu, EeRaCertRequest, RaEeCertInfo, EeRaDownloadRequest

FROM Ieee1609Dot2Dot1EeRaInterface

{iso(1) identified-organization(3) ieee(111) standards-association-numbered-series-standards(2) wave-stds(1609) dot2(2) extension-standards(255) dot1(1) interfaces(1) ee-ra(11) major-version-2(2)}

RaAcaCertRequest, AcaRaCertResponse

FROM Ieee1609Dot2Dot1AcaRaInterface

{iso(1) identified-organization(3) ieee(111) standards-association-numbered-series-standards(2) wave-stds(1609) dot2(2) extension-standards(255) dot1(1) interfaces(1) aca-ra(4) major-version-2(2)}

Ieee1609Dot2Data-SignedX509AuthenticatedCertRequest, ScmsPdu-Scoped, SignerSingleX509Cert

FROM Ieee1609Dot2Dot1Protocol

{iso(1) identified-organization(3) ieee(111) standards-association-numbered-series-standards(2) wave-stds(1609) dot2(2) extension-standards(255) dot1(1) interfaces(1) protocol(17) major-version-2(2)}

;

/\*\*\*\*\*\*\*\*\*\*\*\*

-- Messages

\*\*\*\*\*\*\*\*\*\*\*\*/

EnrolmentRequestMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{enrolmentRequest PRESENT})})}

EnrolmentResponseMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{enrolmentResponse PRESENT})})}

AuthorizationRequestMessage ::= EtsiTs103097Data-Encrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{authorizationRequest PRESENT})})}

AuthorizationRequestMessageWithPop ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{authorizationRequest PRESENT})})}

AuthorizationResponseMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{authorizationResponse PRESENT})})}

CertificateRevocationListMessage ::= EtsiTs103097Data-Signed{EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{certificateRevocationList PRESENT})})}

TlmCertificateTrustListMessage ::= EtsiTs103097Data-Signed{EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{certificateTrustListTlm PRESENT})})}

RcaCertificateTrustListMessage ::= EtsiTs103097Data-Signed{EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{certificateTrustListRca PRESENT})})}

ButterflyAuthorizationRequestMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{butterflyAuthorizationRequest PRESENT})})} X509SignedButterflyAuthorizationRequestMessage ::= EtsiTs103097Data-Encrypted-Unicast {EtsiTs102941ButterflyAuthorizationRequest-X509Signed}

ButterflyAuthorizationResponseMessage ::= EtsiTs103097Data-Signed {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{butterflyAuthorizationResponse PRESENT})})}

ButterflyAtDownloadRequestMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{butterflyAtDownloadRequest PRESENT})})}

ButterflyCertRequestMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{butterflyCertificateRequest PRESENT})})}

ButterflyCertResponseMessage ::= EtsiTs103097Data-SignedAndEncrypted-Unicast {EtsiTs102941Data (WITH COMPONENTS{..., content (WITH COMPONENTS{butterflyCertificateResponse PRESENT})})}

/\*\*\*\*\*\*\*\*\*\*\*\*

-- EtsiTs102941Data

\*\*\*\*\*\*\*\*\*\*\*\*/

EtsiTs102941Data::= SEQUENCE {

 version Version (v1),

 content EtsiTs102941DataContent

 }

EtsiTs102941DataContent ::= CHOICE {

 enrolmentRequest InnerEcRequestSignedForPop,

 enrolmentResponse InnerEcResponse,

 authorizationRequest InnerAtRequest,

 authorizationResponse InnerAtResponse,

 certificateRevocationList ToBeSignedCrl,

 certificateTrustListTlm ToBeSignedTlmCtl,

 certificateTrustListRca ToBeSignedRcaCtl,

 ...,

 /\* Extension for link certificates \*/

 linkCertificateTlm ToBeSignedLinkCertificateTlm,

 singleSignedLinkCertificateRca ToBeSignedLinkCertificateRca,

 doubleSignedlinkCertificateRca RcaSingleSignedLinkCertificateMessage,

 /\* Extension for butterfly key provisioning \*/

 [[butterflyAuthorizationRequest EeRaCertRequest,

 butterflyAuthorizationResponse RaEeCertInfo,

 butterflyCertificateRequest RaAcaCertRequest,

 butterflyCertificateResponse AcaRaCertResponse,

 butterflyAtDownloadRequest EeRaDownloadRequest]]

 } (WITH COMPONENTS{...,

 authorizationRequest (WITH COMPONENTS{...,

 ecSignature (WITH COMPONENTS{...,

 encryptedEcSignature PRESENT

 })

 })

 })

/\*\*\*\*\*\*\*\*\*\*\*\*

-- X509 structures

\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*

 \* @class EtsiTs102941ButterflyAuthorizationRequest-X509Signed

 \*

 \* @brief This structure defines a butterfly authorization request,

 \* signed with an X.509 certificate, by redefining structures from

 \* IEEE 1609.2.1 to fit into the ETSI naming schema.

 \*/

EtsiTs102941ButterflyAuthorizationRequest-X509Signed ::= Ieee1609Dot2Data-SignedX509AuthenticatedCertRequest {

 ScmsPdu-Scoped {

 EeRaInterfacePdu (WITH COMPONENTS {

 eeRaCertRequest})

 },

 SignerSingleX509Cert

}

END

C.0 General

All the messages exchanged are sent as HTTP POST requests. HTTP/1.1 is used.

Parameters for the POST requests and responses are described in the tables below containing the corresponding messages descriptions.

Note: Additional authorization information, such as OAuth access tokens, may be added to requests that are signed with a X.509 enrolment credential if the certificate does not contain authorization information. When ETSI certificates are used, authorization is performed based on the certificate request message SSPs in the ETSI certificate defined in B.4.