|  |
| --- |
| **CHANGE REQUEST** |
|  | ETSI TS 102 941 | **Version** | 2.1.2 | **CR** | 5 | **rev** | 1- |  |
|  |
| **CR Title** | Adding butterfly key option for AT provisioning |
|  |  |
| **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 AT provisioning shall also allow the butterfly key process as another option. This makes the process more efficient for ITS-Ss that require large number of ATs while preserving their privacy. |
|  |  |
| **Consequence if not approved** | Manufacturers must implement different schemes when operating in multiple markets. AT provisioning for large number of keys requires high computational effort, stable connection and large bandwidth by ITS-S. |
|  |  |
| **Summary of change** | Butterfly key process for AT download |
|  |  |
| **Clauses affected** | 6.2.3.1, 6.2.3.5, 6.3.4, A.2 |
|  |  |
| **Linked Change Requests** |  |  |
|  |  |  |
|  |  |
| **Other comments** |  |
|  |  |

6.2.3.1 Introduction

The overall ITS-S initialization sequence to achieve ITS secured message transfer on ITS G5 is given in Figure 12. The messages are specified in details in clauses 6.2.3.2, 6.2.3.3 and 6.2.3.4.

****

**Figure 12: Sequence to achieve signed message transfer between ITS-Ss**

Instead of the depicted workflow for the Authorization Ticket provisioning, a second variant, called Butterfly key provisioning, exists which can be used as an alternative. An implementation must support one of the two options. Since this only applies to the AT provisioning process, the interoperability between vehicles is ensured.

6.2.3.5 Authorization Management with Butterfly Keys

The authorization management with butterfly keys is another option for AT provisioning which can be used as an alternative to the scheme described in Section 6.2.3.3 and Section 6.2.3.4.

The butterfly key scheme is based on the solution described in IEEE 1609.2.1 [XX].

6.2.3.5.0 Overview

When provisioning large number of Authorization Tickets for vehicles that require pseudonymity, the butterfly key mechanism provides advantages to make the process more efficient for both the ITS-Station (ITS-S) and the PKI. The ITS-S only has to issue a single request and the PKI will expand this request and issue multiple ATs that can be downloaded asynchronously.

The ButterflyKeyRequestMessage shall be sent by an ITS-S to the Enrolment Authority (EA) across the interface at reference point S3 (see clause 6.2.2) to request a batch of authorization ticket to be used in subsequent ITS communications. The EA shall respond with a ButterflyAuthorizationResponseMessage that contains information on when the batch of Authorization Tickets can be downloaded.

The EA then performs the butterfly key expansion and sends multiple ButterflyCertRequestMessages to the Authorization Authority (AA) across the interface at reference point S4. The AA issues the Authorization Tickets (ATs) and responds with a ButterflyCertResponseMessage.

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 ITS-S sends a ButterflyAtDownloadRequestMessage to the EA across the interface at reference point S3 to request the download of its AT batch. The EA responds with a collection of AcaEeCertResponsePrivateSpdu containing the ATs that it received from the AA in the previous step.

Figure 23 shows an example of a message sequence for a successful butterfly authorization request.

****

**Figure 23: Message sequence for butterfly authorization request and response**

All messages supporting the authorization process will satisfy the following security and privacy requirements:

* integrity, data origin authenticity, and confidentiality shall be ensured;
* authorization and access control: only registered and authenticated ITS stations shall get Authorization Tickets that enable them to access to cooperative ITS services.

For ITS-S that need privacy protection such as ITS-S vehicles or personal devices, the following requirements apply:

* pseudonymity of the ITS-S requester towards external attackers and towards the Authorization Authority should be ensured;
* unlinkability of the ITS-S requesting Authorization Tickets: the issued ATs from the butterfly authorization request should not be linked to the same ITS-S requester or linked between them.

6.2.3.5.1 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 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 se 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.

6.2.3.5.2 Butterfly authorization response

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

* 

NOTE: The signature is computed using the verification private key associated with the EA certificate.

**Figure 25: ButterflyAuthorizationResponse message**

To read a butterfly authorization response, the ITS-S shall receive an EtsiTs103097Data-Signed structure, containing an EtsiTs102941Data structure, containing a butterflyAuthorizationResponse structure:

The outermost structure is an EtsiTs103097Data-Signed structure that shall contain 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 an EtsiTs102941Data structure;
		- in the headerInfo:
* the psid shall be se 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 as a digest referencing the hashedId8 of EA certificate;
	+ the signature over tbsData computed using the EA private key corresponding to its public verification key found in the EA certificate.

The butterflyAuthorizationResponse shall contain:

* The version shall be set to 2;
* The generationTime shall contain the generation time of this data structure;
* The currentI shall contain the i-value that is associated with the current batch of Authorization Tickets (ATs);
* The requestHash is the left-most 16 octets of the SHA256 digest of the COER representation of the topmost EtsiTs103097Data-Encrypted structure of the received request (see Figure 17 and Figure 18);
* The nextDlTime shall contain the time after which the ITS-S shold try to download the AT batch;
* The acpcTreeId shall be absent.

6.2.3.5.3 Butterfly certificate request

The following functional requirements are defined on the communication flow of Figure 23.

The butterfly expansion, for both the verification and, if applicable, the certificate encryption key, shall be done according to IEEE 1609.2.1 The EA may provide additional privacy to end entities by “shuffling” together individual certificate requests from many different end entities, creating confusion at the AA as to which certificates belong to which end entities.

The ButterflyCertRequest message shall be encrypted using an ETSI TS 103 097 [3] approved algorithm and the encryption shall be done with the public key provided by the Authorization Authority (AA).

The complete nested data structure of the ButterflyCertRequest message is specified in Figure 26. The specification of the ButterflyCertRequest 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 AA. The signature is computed using the verification private key associated with the EA certificate.

**Figure 26: ButterflyCertRequest message**

To create a ButterflyCertRequest, the Enrolment Authority (EA) shall follow this process:

* A RaAcaCertRequest structure is built, with:
	+ The version shall be set to 2.
	+ The generationTime shall correspond to the generation time of this structure;
	+ The flags shall be unset for all options (no butterflyExplicit, no useCubk);
	+ The linkageInfo shall be absent;
	+ The certEncKey corresponding to the cocoon key that shall be derived by the EA
		- For the original option: using the caterpillar key provided as encryptionKey and expansion parameter;
		- for the unified option: using the caterpillar key provided as verifyKeyIndicator and expansion parameter;
	+ The tbsCert structure corresponding to the information provided to the EA by the ITS-S with the ButterflyAuthorizationRequest except for the verifyKeyIndicator being expanded to the cocoon key by the EA;
* An EtsiTs102941Data structure is built, with:
	+ the version set to v1 (integer value set to 1);
	+ the content set to the previous data structure (RaAcaCertRequest).
* 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 RaAcaCertRequest structure;
		- in the headerInfo:
			* the psid shall be se 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 EA certificate;
	+ the signature over tbsData computed using the private key corresponding to the EA’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 AA certificate in recipientId; and
	+ the encrypted data encryption key in encKey, the public key to use for encryption is the encryptionKey found in the AA certificate referenced in recipientId;
	+ the component ciphertext containing the encrypted representation of the previous EtsiTs103097Data-Signed structure.

6.2.3.5.4 Butterfly certificate response

The following functional requirements are defined on the communication flow of Figure 23.

The ButterflyCertResponse message shall be encrypted using an ETSI TS 103 097 [3] approved algorithm and the encryption shall be done with the same AES key as the one used by the EA requestor for the encryption of the ButterflyCertRequest message.

The complete nested data structure of the ButterflyCertResponse message is specified in Figure 27. The specification of the ITS-S ButterflyCertResponse message using ASN.1 [6], [7] shall be as specified in clause A.2.



NOTE: Encryption is done with the AES key used for the encryption of the Butterfly Authorization Request. The signature is computed using the verification private key associated with the AA certificate.

**Figure27: ButterflyAuthorizationResponse message**

To read a butterfly certificate response, the EA shall receive an EtsiTs103097Data-Encrypted structure, containing an EtsiTs103097Data-Signed structure, containing an EtsiTs102941Data structure, containing a butterflyCertResponse structure:

* The outermost structure is an EtsiTs103097Data-Encrypted structure with:
	+ the component recipients containing one instance of RecipientInfo of choice pskRecipInfo, which contains the HashedId8 of theSymmetricEncryptionKey structure containing the symmetric key used by the EA to encrypt the ButterflyCertRequest message to which the response is built;
	+ the component ciphertext, once decrypted, contains an EtsiTs103097Data-Signed structure.

If the EA has been able to decrypt the content, this expected EtsiTs103097Data-Signed structure shall contain 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 an EtsiTs102941Data structure;
		- in the headerInfo:
* the psid shall be se 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 as a digest referencing the hashedId8 of AA certificate;
	+ the signature over tbsData computed using the AA private key corresponding to its public verification key found in the AA certificate.

The butterflyCertificateResponse shall contain:

* The version shall be set to 2;
* The generationTime shall contain the generation time of this data structure;
* The requestHash is the left-most 16 octets of the SHA256 digest of the COER representation of the topmost EtsiTs103097Data-Encrypted structure of the received request (see Figure 17 and Figure 18);
* The acaResponse shall contain encrypted Authorization Ticket (AT) in the private field. The encoding is specified in Section 7.3.4 of IEEE 1609.2.1

6.2.3.5.5 Butterfly AT download request

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

Note: The AT download request does not contain confidential information and thus it may also be issued by other means, e.g., using a REST API.

* If the ButterflyAtDownloadRequest message shall be encrypted, it shall use an ETSI TS 103 097 [3] approved encryption algorithm and the public key provided by the authorization authority.
* The contents of the Butterfly AT Download Request message shall be as described in Figure 28.
* The complete nested data structure of the Butterfly AT Download Request message is specified in Figure 28. The specification of the content of the Butterfly AT Download Request 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 28: ButterflyAtDownloadRequest message**

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

* An EeRaDownloadRequest structure is built, with:
	+ The generationTime being set to the generation time of the this structure,
	+ The filename being set in the format *h-i*.zip
		- *h* is the 16-byte ASCII hex representation of the HashedId8 of the submitted butterfly authorization request message, as returned in the requestHash field of butterflyAuthorizationResponse structure.
		- *i* is the i-value.
* An EtsiTs102941Data structure is built, with:
	+ the version set to v1 (integer value set to 1);
	+ the content set to the previous data structure (EeRaDownloadRequest).
* 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 EeRaDownloadRequest structure;
		- in the headerInfo:
			* the psid shall be se 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.

6.2.3.5.5 Butterfly AT download response

The butterfly AT download response corresponds to a ZIP file, as requested with the ButterflyAtDownloadRequest message. The EA generates this ZIP file by collecting all AcaEeCertResponsePrivateSpdu that is received as acaResponse with the ButterflyCertResponse message from the AA.

The data format within the ZIP file shall correspond to the specification in IEEE 1609.2.1.

6.3.4 Specification of Full CTL and Delta CTL

[…]

EaEntry shall contain an EA certificate, the URL for the connection by the AA, and optionally the URL for the connection by the ITS-Station. If the EA only supports butterfly authorization requests, the URL for connection by the AA shall be set to an empty string.

AaEntry shall contain an AA certificate, and optionally the URL for the connection by the ITS-Station. The URL for connection by the EA to allow butterfly key requests shall be distributed using out-of-band-measures.

[…]

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) }

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)}

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})})}

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]]

 }

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) }

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)}

;

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

-- 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})})}

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

 })

 })

 })

END