ETSI TS 103 096-2 V1.1.1 (2013-07)

Technical Specification

Intelligent Transport Systems (ITS); Testing; Conformance test specification for TS 102 867 and TS 102 941; Part 2: Test Suite Structure and Test Purposes (TSS&TP)

Reference

DTS/ITS-0050019

Keywords

ITS, testing, TSS&TP

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Intelligent Transport Systems (ITS).

The present document is part 2 of a multi-part deliverable covering Conformance test specification for ITS Security as identified below:

TS 103 096-1: "Protocol Implementation Conformance Statement (PICS)";

TS 103 096-2: "Test Suite Structure and Test Purposes (TSS&TP)";

TS 103 096-3: "Abstract Test Suite (ATS) and Protocol Implementation eXtra Information for Testing (PIXIT)";

TR 103 096-4: "Validation report".

1 Scope

The present document provides the Test Suite Structure and Test Purposes (TSS&TP) for Security as defined in IEEE P 1609.2 [1], TS 102 941 [2] and TS 102 867 [3] in compliance with the relevant requirements and in accordance with the relevant guidance given in ISO/IEC 9646-7 [9].

The ISO standard for the methodology of conformance testing (ISO/IEC 9646-1 [6] and ISO/IEC 9646-2 [7]) as well as the ETSI rules for conformance testing (ETS 300 406 [10]) are used as a basis for the test methodology.

2 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 reference document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://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.

2.1 Normative references

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

[1] IEEE P1609.2/D12 (January 2012): "IEEE Draft Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages.

[2] ETSI TS 102 941: "Intelligent Transport Systems (ITS); Security; Trust and Privacy Management".

[3] ETSI TS 102 867: "Intelligent Transport Systems (ITS); Security; Stage 3 mapping for IEEE 1609.2".

[4] ETSI TS 103 096-1 (V1.1.1): "Intelligent Transport Systems (ITS); Testing; Conformance test specification for TS 102 867 and TS 102 941; Part 1: Protocol Implementation Conformance Statement (PICS)".

[5] ETSI TS 103 096-3 (V1.1.1): "Intelligent Transport Systems (ITS); Testing; Conformance test specification for TS 102 867 and TS 102 941; Part 3: Abstract Test Suite (ATS) and Protocol Implementation eXtra Information for Testing (PIXIT)".

[6] ISO/IEC 9646-1 (1994): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 1: General concepts".

[7] ISO/IEC 9646-2 (1994): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 2: Abstract Test Suite specification".

[8] ISO/IEC 9646-6 (1994): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 6: Protocol profile test specification".

[9] ISO/IEC 9646-7 (1995): "Information technology -- Open Systems Interconnection -- Conformance testing methodology and framework -- Part 7: Implementation Conformance Statements".

[10] ETSI ETS 300 406 (1995): "Methods for testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".

2.2 Informative references

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]	ETSI EG 202 798: "Intelligent Transport Systems (ITS); Testing; Framework for conformance and interoperability testing".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

- terms given in IEEE 1609.2 [1], TS 102 941 [2] and in TS 102 867 [3];
- terms given in ISO/IEC 9646-6 [8] and in ISO/IEC 9646-7 [9].

3.2 Abbreviations

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

AA	Authorization Authority
BV	Normal behaviour
CA	Certification Authority
CAM	Cooperative Awareness Message
CRL	Certificate Revocation List
CSR	Certificate Signing Request
DENM	Decentralized Environmental Notification Message
EA	Enrolment Authority
EB	Exceptional Behavior
ITS	Intelligent Transport System
ITS-AID	ITS Application ID
ITS-S	ITS Station
IUT	Implementation Under Test
MSEC	Multicast Security
PKI	Public Key Infrastructure
PSID	Provider Service Identifier
SA	Security Association
SSP	Service Specific Permissions
TLS	Transport Layer Security
TP	Test Purposes
TSS	Test Suite Structure

4 Prerequisites and Test Configurations

4.1 Test Configurations

The test configuration 1 as shown in figure 1 is applied for the test group of CA and EA tests.

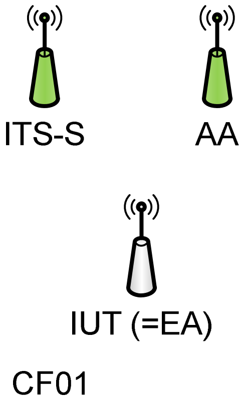

Figure 1: Test Configuration 1

The test configuration 2 as shown in figure 2 is applied for the test group of CA and AA tests.

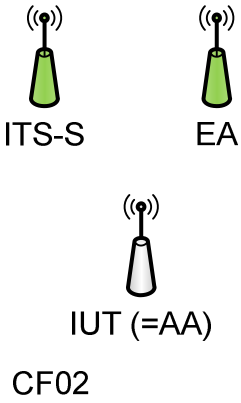

Figure 2: Test Configuration 2

The test configuration 3 as shown in figure 3 is applied for the test group of ITS-S Enrolment and Authorization tests.

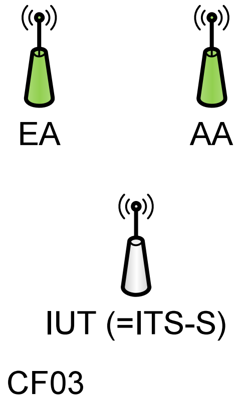

Figure 3: Test Configuration 3

The test configuration 4 as shown in figure 4 is applied for the test group of ITS-S Send and Receive Data tests.

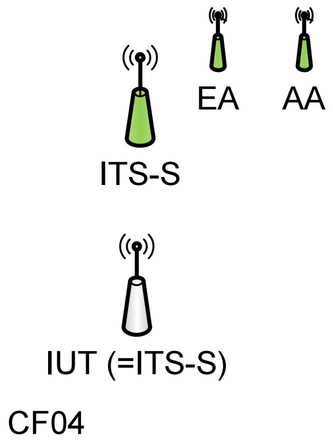

Figure 4: Test Configuration 4

4.2 PKI Hierarchy

The PKI Hierarchy is depicted below. Four different types of certificates are defined. They are listed hereafter.

- CERT_ROOT
- CERT_EA_x
- CERT_AA_x
- CERT_ENR_x
- CERT_AUTH_x

These names are used in the TP definitions, where _x is a placeholder for numbering different certificates.

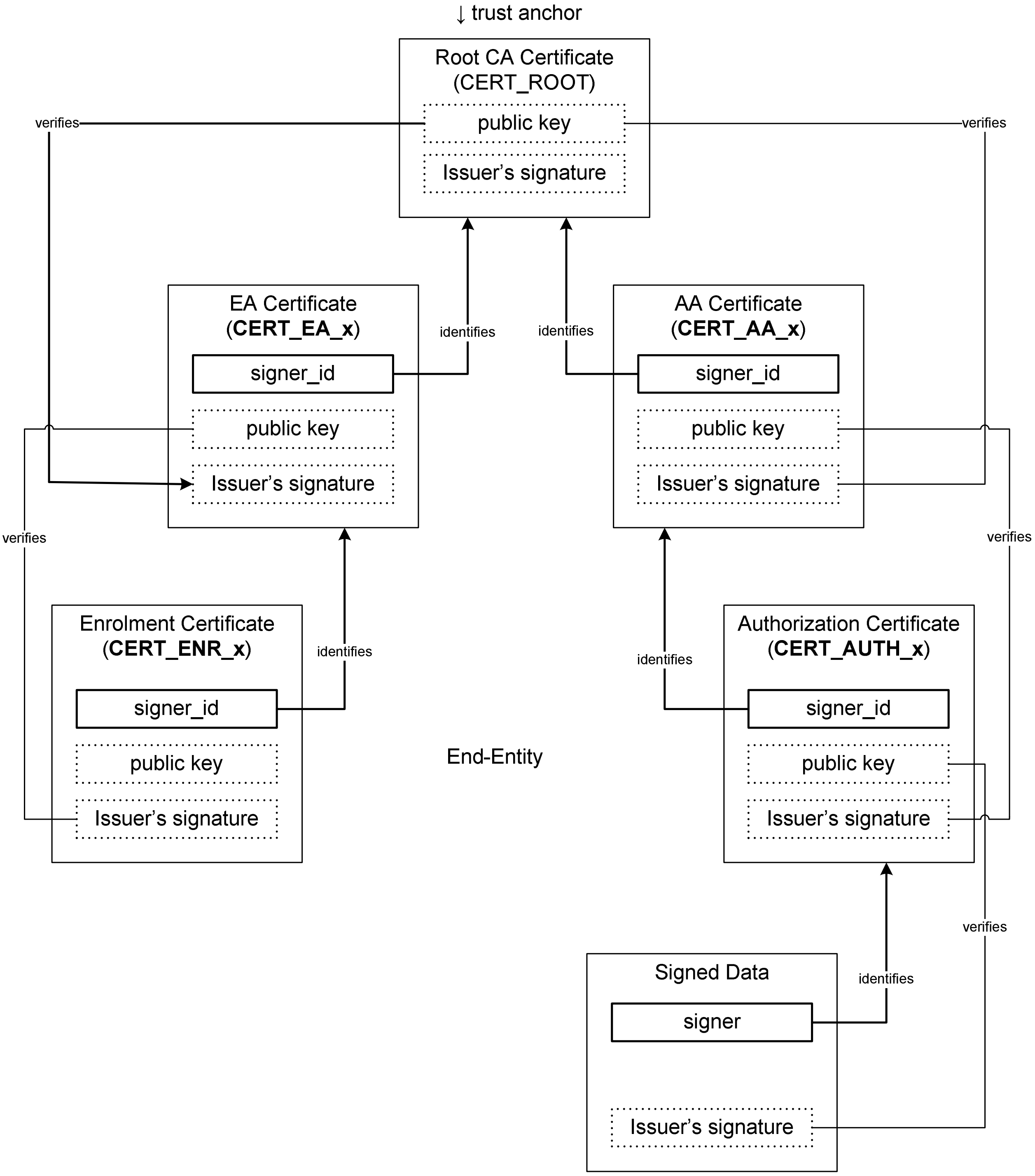

Figure 5: PKI Hierarchy

4.3 Feature Restriction and Pre-Enrolment

4.3.1 Feature Restriction

In this clause all feature restrictions are listed:

- Certificate chains where subordinate certificates make use of inherited permissions are not supported
- Only circular regions

- Only explicit certificates
- Revocation is not tested, i.e. certificate responses contain only empty revocation list
- Update Enrolment Credentials is not tested
- Remove Enrolment Credentials is not tested
- Update Authorization Tickets is not tested
- The name which identifies the CA shall be no longer that 32 bytes

4.3.2 Pre-Enrolment

Enrolment is the process by which an ITS-S obtains an enrolment certificate, which can later be used to authenticate requests for authorization certificates. An ITS-S undergoes initial enrolment by executing the Enrolment Request information flow from TS 102 941 [2].

When devices enrol with an Enrolment Authority, they should be authenticated as devices that are entitled to receive enrolment credentials of the type requested. There are two three different authentication approaches:

- Public key: Enrolment requests are authenticated by using a private key of the ITS-S. The corresponding public key is previously registered with a unique ITS-S module ID at the EA in a secure process. Every ITS-S has to be registered separately.
- Certificate: Enrolment requests are authenticated by a certificate or certificate chain.
- Self-signed: Enrolment requests are signed by the public key contained in the enrolment request. In this case the signature provides proof of possession of the corresponding private key, but does not authenticate that the private key holder is in fact authorized to receive an enrolment credential of the type requested. This authorization is provided by other mechanisms.

None of the three authentication approaches start at the device lifecycle: in all cases, there is the question of how the device is originally shown to be authenticated. The test system supports both the certificate and the self-signed forms of enrolment request.

For enrolment request:

- The test system enrolment authority shall accept the following forms of authorization, certificate and self-signed.
- The test system enrolment authority shall check that the signature on the enrolment request is cryptographically valid.
- In the case of an enrolment request signed by a certificate:
    - The test system enrolment authority shall check that the request is consistent with the permissions in the certificate.
    - The test system enrolment authority shall not carry out any other validation on the signing certificate. For example, it shall not check the signature on the signing certificate, check that the certificate chains back to a known CA, or check whether the signing certificate is revoked.

The test system enrolment authority shall issue the enrolment certificate if these validity tests pass.

From the perspective of the IUT, this has the following consequences:

- Certificate: The IUT shall be provisioned with a certificate to authenticate enrolment before testing begins (a pre-enrolment certificate).
    - The supplier shall provide instructions as to how to reset the IUT to a state where it has the pre-enrolment certificate but not the enrolment certificate, to allow the enrolment flow to be run multiple times.

  - The supplier shall chose between two options:
    - The test system generates private key and public certificate for the device.
    - The supplier generates a private key and sends a certificate signing request to the test system.

- Self-signed: The IUT supplier shall provide instructions as to how to set the IUT into a state where it will request enrolment with a self-signed request.

4.4 States in Initial Conditions

Each TP contains an initial condition. The initial condition defines in which initial state the IUT has to be to apply the actual TP. In the corresponding Test Case, when the execution of the initial condition does not succeed, it leads to the assignment of an Inconclusive verdict. This clause defines the different initial states of the IUT.

4.4.1 ITS-S send side states

- Not enrolled state: ITS-S has all info necessary to send an EnrolmentRequest but does not have any Enrolment credentials yet
- Awaiting EnrolmentResponse state: ITS-S has sent an EnrolmentRequest and is waiting for an EnrolmentResponse
- Enrolled, but not authorized state: ITS-S has received EnrolmentResponse and is able to send AuthorizationRequest
- Awaiting AuthorizationResponse state: ITS-S has sent an AuthorizationRequest and is waiting for an AuthorizationResponse
- Authorized state: ITS-S has received a successful AuthorizationResponse

4.4.2 ITS-S receive side states

- Operational state: ITS-S has the root certificate and is ready to receive messages

4.4.3 EA states

- Operational state: EA has obtained its certificate and is ready to receive and send Enrolment messages

4.4.4 AA states

- Operational state: AA has obtained its certificate and is ready to receive and send Authorization messages

4.5 Validity of Signed Communication

The check of the validity of signed communication according to clause 5.5 of IEEE P1609.2/D12 [1] (e.g. consistency check of the certificate chain, consistency check between certificate and message etc) forms an integral part of the test suite and is described in TS 103 096-3 [5], clause 6.

4.6 Introduction of Snippets of Data Structures

The data structures in IEEE P1609.2/D12 [1] can become quite complex. In order to allow to write a TP in a concise form, the usage of snippets has been introduced. A snippet is a partial extract of a data structure which is assigned with values. A snippet can be used within a TP. Please refer to clause 6.1.8 for a complete list of all defined snippets.

Within a TP, any element of the snippet can be overwritten or extended. In the example below the TP extends the snippet **MSG_ENRRSP_TS** 'signature.ecdsa_signature' to ' signature.ecdsa_signature.R.type = = uncompressed'.

```
...
when {
        the IUT receives a valid CertificateResponse (EnrolmentResponse) set to MSG_ENRRSP_TS
            containing certificate_chain[last].signature.ecdsa_signature.R.type
                set to uncompressed
         ...
```

4.7 Variants, Variables and Snippet Naming Convention

The TPs use the concept of variants, variables and snippets. Their definition, how they are used and their naming conventions are defined in this clause.

Variants: In case where for a single field multiple values can be tested (e.g. different public key types), then a table is appended after the TP. This table lists all the different value which need to be tested. The TP identifier is appended with –X (e.g. **TP/SEC/ITS-S/ENR/NB-02-X**). If there are two fields for which multiple values can be tested then X and Y are appended. The field itself is written as X_FIELD_NAME (e.g. **X_PKT_SIGNATURE**).

Variables: Variables are used in TPs in order to highlight the fact that a particular part of request message needs to re-appear in a response message. For example for a TP where the IUT has sent an EnrolementRequest with a permission list, and the test system needs to sent the same permission list back, then the denotation of **V_PERM_LIST (see TP/SEC/ITS-S/ENR/NB-11)**

Snippets: For the definition of snippets refer to the previous clause. The naming convention for snippets is defined to upper case and to have no specific prefix (e.g. **MSG_ENRREQ_IUT**). All snippets in TPs contain hyperlinks which allows to navigate from the TP directly to the snippet definition.

5 Test Suite Structure (TSS)

5.1 Structure for Security tests

Table 1 shows the Test Suite Structure (TSS) including its subgroups defined for conformance testing.

Table 1: TSS for SECURITY

Root	Group	Group	category
SEC	CA	ENR/AUTH	Normal behaviour
			Exceptional behaviour
	EA	ENR	Normal behaviour
			Exceptional behaviour
	AA	AUTH	Normal behaviour
			Exceptional behaviour
	ITS-S	ENR	Normal behaviour
			Exceptional behaviour
		AUTH	Normal behaviour
			Exceptional behaviour
		S-DATA	Normal behaviour
			Exceptional behaviour
		R-DATA	Normal behaviour
			Exceptional behaviour

The test suite is structured as a tree with the root defined as SEC. The tree is of rank 3 with the first rank a Group, the second rank a sub group, and the last rank a category.

5.2 Test groups

The test suite has a total of four levels. The first level is the root. The second level defines different IUTs. The third level defines various functional areas. The fourth level differentiates normal and exceptional behaviour.

5.2.1 Root

The root identifies ITS G5A as defined in IEEE 1609.2 [1], TS 102 941 [2] and TS 102 867 [3].

5.2.2 Groups

There are four functional areas identified as groups:

- Certificate Authority
- Enrolment Authority
- Authorization Authority
- ITS Station

5.2.3 Sub groups

There are four functional areas identified as sub-groups:

- Enrolment
- Authorization
- Send Data
- Receive Data

5.2.4 Categories

Test categories are limited to the normal and exceptional behaviour.

6 Test Purposes (TP)

6.1 Introduction

6.1.1 TP definition conventions

The TP definition is constructed according to EG 202 798 [i.1].

6.1.2 TP Identifier naming conventions

The identifier of the TP is constructed according to table 2.

Table 2: TP naming convention

Identifier:	**TP_<root>_<gr>_<sgr>_<x>_<nn>**		
	<root> = root	SEC	
	<gr> = group	CA	Certificate Authorithy
		EA	Enrolment Authorithy
		AA	Authorization Authority
		ITS-S	ITS Station
	<sgr> =sub-group	ENR	Enrolment
		AUTH	Authorization
		S-DATA	Send Data
		R-DATA	Receive Data
	<x> = type of testing	NB	Normal Behaviour
		EB	Exceptional Behaviour
	<nn> = sequential number		01 to 99
	<X> = Variant for 1st permutation table		A to Z
	<Y> = Variant for 2nd permutation table		A to Z

6.1.3 Rules for the behaviour description

The description of the TP is constructed according to EG 202 798 [i.1].

In the TP the following wordings are used:

- "The IUT is requested to send": An upper layer requests the security layer to apply processing to a packet.
- "The IUT receives": for packets coming from the network and given by the lower layer.
- "The IUT is configured to": the Security Layer on the IUT is requested to include a certain data element, e.g. this can be manually configured or triggered by use of a application that requires this data element.
- "The IUT accepts": the Security Layer on the IUT interprets a received message as passing all the relevant validity tests, including cryptographic validity, and passes it to a higher layer for interpretation.
- "The IUT discards": the Security Layer on the IUT interprets a received message as failing at least one validity test and does not pass it to a higher layer (drops a received message).

6.1.4 Sources of TP definitions

All TPs specified in the present document are derived from the behaviour defined in IEEE 1609.2 [1], TS 102 941 [2] and TS 102 867 [3].

6.1.5 Mnemonics for PICS reference

The following table lists mnemonic names and maps them to the PICS item number.

Table 3: Mnemonics for PICS reference

Mnemonic	PICS item
PIC_Generate_SignPayload	[4] Table A.5/1
PIC_Generate_SignExternalPayload	[4] Table A.5/2
PIC_Generate_SignPartialPayload	[4] Table A.5/3
PIC_Generate_Identified	[4] Table A.5/7
PIC_Generate_GenerationTime	[4] Table A.5/9
PIC_Generate_GenerationLocation	[4] Table A.5/10
PIC_Generate_ExpirationTime	[4] Table A.5/11
PIC_Generate_Certificate	[4] Table A.5/13
PIC_Generate_Ecdsa224	[4] Table A.5/15
PIC_Generate_Ecdsa256	[4] Table A.5/16
PIC_Generate_ExplicitCertificates	[4] Table A.5/17
PIC_Generate_Uncompressed	[4] Table A.5/19
PIC_Generate_Compressed	[4] Table A.5/20
PIC_Generate_CompressedFastVerification	[4] Table A.5/21
PIC_Generate_UncompressedKey	PIC_Generate_Uncompressed
PIC_Generate_CompressedKey	PIC_Generate_Compressed AND PIC_Generate_CompressedFastVerification
PIC_Generate_XCoordinateOnlyKey	PIC_Generate_Compressed AND NOT PIC_Generate_CompressedFastVerification
PIC_Generate_SelfSigned	[4] Table A.34/2
PIC_Generate_StartValidity	[4] Table A.34/16
PIC_Generate_LifetimeIsDuration	[4] Table A.34/17
PIC_ Generate_StartValidityIsATimestamp	NOT PIC_Generate_LifetimeIsDuration
PIC_Generate_VerificationKey224	[4] Table A.34/19
PIC_Generate_VerificationKey256	[4] Table A.34/20
PIC_Generate_EncryptionKey	[4] Table A.34/21
PIC_Generate_PsidArrayWithMoreThan8Entries	[4] Table A.37/2
PIC_Verify_Uncompressed	[4] Table A.14/17
PIC_Verify_Compressed	[4] Table A.14/18
PIC_Verify_CompressedFastVerification	[4] Table A.14/19
PIC_Verify_UncompressedKey	PIC_Verify_Uncompressed
PIC_Verify_CompressedKey	PIC_Verify_Compressed AND PIC_Verify_CompressedFastVerification
PIC_Verify_XCoordinateOnlyKey	PIC_Verify_Compressed AND NOT PIC_Verify_CompressedFastVerification
PIC_Verify_SelfSigned	[4] Table A.35/1
PIC_Verify_StartValidity	[4] Table A.41/9
PIC_Verify_LifetimeIsDuration	[4] Table A.41/10
PIC_Verify_StartValidityIsATimestamp	NOT PIC_Verify_LifetimeIsDuration
PIC_Verify_VerificationKey224	[4] Table A.41/11
PIC_Verify_VerificationKey256	[4] Table A.41/12
PIC_Verify_EncryptionKey	[4] Table A.41/13
PIC_Verify_PsidArrayWithMoreThan8Entries	[4] Table A.45/2

6.1.6 Message encapsulation

CertificateRequest message encapsulation

```
Structure 1609Dot2Data {
    containing type
        indicating encrypted
    containing encrypted_data
        containing symm_algorithm set to unknown
        containing recipients
            containing cert_id
            containing enc_key
        containing ciphertext
/----------------- After deciphering process ------------------/
/           containing type                                    /
/               set to certificate_request                     /
/           containing request                                 /
/               containing the CerticateRequest data           /
/--------------------------------------------------------------/
}
```

NOTE: When a TP refers to a CertificateRequest, then it is assumed that the CertificateRequest is received in a 1609Dot2Data as described above.

CertificateResponse message encapsulation

```
Structure 1609Dot2Data {
    containing type
        indicating encrypted
    containing encrypted_data
        containing symm_algorithm set to unknown
        containing recipients
            containing cert_id
            containing enc_key
        containing ciphertext
/--------------------------- After deciphering process -----------------------------/
/           containing type                                                         /
/               set to certificate_response                                         /
/           containing request                                                      /
/               containing the CerticateResponse data                               /
/-----------------------------------------------------------------------------------/
}
```

NOTE: When a TP refers to a CertificateResponse, then it is assumed that the CertificateResponse is received in a 1609Dot2Data as described above.

CertificateRequestError message encapsulation

```
Structure 1609Dot2Data {
    containing type
        indicating encrypted
    containing encrypted_data
        containing symm_algorithm set to unknown
        containing recipients
            containing cert_id
            containing enc_key
        containing ciphertext
/------------------------ After deciphering process ---------------------------/
/           containing type                                                   /
/               set to certificate_request_error                              /
/           containing request                                                /
/               containing the CertificateRequestError data                   /
/-----------------------------------------------------------------------------/
}
```

NOTE: When a TP refers to a CertificateRequestError, then it is assumed that the CertificateRequestError is received in a 1609Dot2Data as described above.

6.1.7 Used constants

NAME	Value
CLT	Current Local Time
ANY_VALUE_OR_NONE	*
ANY_VALUE	?
ANY_SCOPE	anonymous_scope or id_scope or sec_data_exch_ca_scope
ETSI_LAT	
ETSI_LON	
NICE_LAT	
NICE_LON	
PARIS_LAT	
PARIS_LON	
PSID_A	These PSIDs shall be defined before test execution
PSID_B	
PSID_C	
PSID_D	
PSID_E	
PSID_F	
PSID_G	
PSID_H	
PSID_I	These PSIDs shall be defined only when IUT supports more than 8 PSID
PSID_J	
PSID_K	
PSID_L	

6.1.8 Snippets definitions

6.1.8.1 Regions

Table 4: Regions definitions

```
REGION_LARGE :=
GeographicRegion {
    containing region_type set to 'circle'
    containing circular_region
        containing center
            containing latitude set to ETSI_LAT
            containing longitude set to ETSI_LON
        containing radius set to 65KM
}

REGION_MEDIUM :=
GeographicRegion {
    containing region_type set to 'circle'
    containing circular_region
        containing center
            containing latitude set to ETSI_LAT
            containing longitude set to ETSI_LON
        containing radius set to 32KM
}

REGION_SMALL :=
GeographicRegion {
    containing region_type set to 'circle'
    containing circular_region
        containing center
            containing latitude set to ETSI_LAT
            containing longitude set to ETSI_LON
        containing radius set to 1KM
}

REGION_OUTSIDE :=
GeographicRegion {
    containing region_type set to 'circle'
    containing circular_region
        containing center
            containing latitude set to PARIS_LAT
            containing longitude set to PARIS_LON
        containing radius set to 65KM
}

REGION_INTERSECTING :=
GeographicRegion {
    containing region_type set to 'circle'
    containing circular_region
        containing center
            containing latitude set to NICE_LAT
            containing longitude set to NICE_LON
        containing radius set to 65KM
}
```

6.1.8.2 Certificates

6.1.8.2.1 Authorities certificates

Table 5: Root certificate definition

```
CERT_ROOT :=
Certificate {
    containing version_and_type
        set to 'explicit_certificates'(2)
    containing unsigned_certificate
        containing subject_type
            set to 'root_ca'
        containing cf
            set to 'use_start_validity' and 'lifetime_is_duration'
        not containing signer_id
        containing scope
            containing name
                set to 'ETSI Root CA'
            containing permitted_subject_types
                set to array[1] {
                    'sec_data_exch_ca'
                }
            containing permissions
                containing type
                    set to 'specified'
                containing permissions_list
                    set to array[0]
            containing region
                containing region_type
                    set to 'none'
        containing expiration
            set to '2020-12-31'
        containing lifetime
            set to '10Y'
        containing crl_series
            set to 0
        containing verification_key
            containing algorithm
                set to 'ecdsa_nistp256_with_sha256'
            containing public_key
                containing type
                    set to 'uncompresed'
                containing x/y
                    set to a valid key for ECDSA-256
        not containing encryption_key
    containing signature
        containing ecdsa_signature
            verifiable with unsigned_certificate.verification_key
            containing R
                containing type
                    set to 'x_coordinate_only'
                containing x
}
```

Table 6: Enrolment authority certificate definition

```
CERT_EA :=
Certificate {
    containing version_and_type
        set to 'explicit_certificates'(2)
    containing unsigned_certificate
        containing subject_type
            set to 'sec_data_exch_ca'
        containing cf
            set to 'use_start_validity' and 'lifetime_is_duration'
        containing signer_id
            set to the 8-byte hash of CERT_ROOT
        containing signature_alg
            set to 'ecdsa_nistp256_with_sha256'
        containing scope
            containing name
                set to 'ETSI EA'
            containing permitted_subject_types
                set to array[1] {
                    'sec_data_exch_ca'
                }
            containing permissions
                containing type
                    set to 'specified'
                containing permissions_list
                    set to array[0]
            containing region
                set to REGION_LARGE
        containing expiration
            set to '2020-12-31'
        containing lifetime
            set to '10Y'
        containing crl_series
            set to 0
        containing verification_key
```

```
            containing algorithm
                set to 'ecdsa_nistp256_with_sha256'
            containing public_key
                containing type
                    set to 'uncompresed'
                containing x/y
                    set to a valid key for ECDSA-256
        containing encryption_key
            containing algorithm
                set to 'ecies_nistp256'
            containing supported_symm_alg
                set to 'aes_128_ccm'
            containing public_key
                containing type
                    set to 'uncompresed'
                containing x/y
                    set to a valid key for ECIES-256
    containing signature
        containing ecdsa_signature
            verifiable with CERT_ROOT.verification_key
            containing R
                containing type
                    set to 'x_coordinate_only'
                containing x
}
```

Table 7: Authorization authority certificate definition

```
CERT_AA :=
Certificate {
    containing version_and_type
        set to 'explicit_certificates'(2)
    containing unsigned_certificate
        containing subject_type
            set to 'sec_data_exch_ca'
        containing cf
            set to 'use_start_validity' and 'lifetime_is_duration'
        containing signer_id
            set to the 8-byte hash of CERT_ROOT
        containing signature_alg
            set to 'ecdsa_nistp256_with_sha256'
        containing scope
            containing name
                set to 'ETSI AA'
            containing permitted_subject_types
                set to array[1] {
                    'sec_data_exch_ca'
                }
            containing permissions
                containing type
                    set to 'specified'
                containing permissions_list
                    set to array[0]
            containing region
                set to REGION_LARGE
        containing expiration
            set to '2020-12-31'
        containing lifetime
            set to '10Y'
        containing crl_series
            set to 0
        containing verification_key
            containing algorithm
                set to 'ecdsa_nistp256_with_sha256'
            containing public_key
                containing type
                    set to 'uncompresed'
                containing x/y
                    set to a valid key for ECDSA-256
        containing encryption_key
            containing algorithm
                set to 'ecies_nistp256'
            containing supported_symm_alg
                set to 'aes_128_ccm'
            containing public_key
                containing type
```

```
                    set to 'uncompresed'
                containing x/y
                    set to a valid key for ECIES-256
    containing signature
        containing ecdsa_signature
            verifiable with CERT_ROOT.verification_key
            containing R
                containing type
                    set to 'x_coordinate_only'
                containing x
}
```

6.1.8.2.2 End-Entities certificates

6.1.8.2.2.1 Certificates issued by test system

Table 8: Enrolment certificate issued by test system

```
CERT_ENR_TS :=
Certificate {
    containing version_and_type
        set to 'explicit_certificates'(2)
    containing unsigned_certificate
        containing subject_type
            set to 'sec_data_exch_csr'
        containing cf
            indicating 'use_start_validity' and 'lifetime_is_duration'
        containing signer_id
            set to 8-byte hash of the CERT_EA
        containing signature_alg
            set to 'ecdsa_nistp256_with_sha256'
        containing scope
            containing name
                set to 'EC_SCOPE_DEFAULT'
            containing permitted_subject_types
                set to MSG_ENRREQ_IUT.unsigned_csr
                        .type_specific_data.sec_data_exch_ca_scope.permitted_subject_types
            containing permissions
                set to MSG_ENRREQ_IUT.unsigned_csr
                        .type_specific_data.sec_data_exch_ca_scope.permissions
            containing region
                set to MSG_ENRREQ_IUT.unsigned_csr.type_specific_data.sec_data_exch_ca_scope.region
        containing expiration
        containing lifetime
        containing crl_series
            set to 0
        containing verification_key
            set to MSG_ENRREQ_IUT.unsigned_csr.verification_key
    containing signature
        containing ecdsa_signature
            verifiable with CERT_EA.verification_key
            containing R
                containing type
                    set to 'compressed_y_0' or 'compressed_y_1'
                containing x/y
                    set to a valid key for ECDSA-256
}
```

NOTE: This certificate is a response to the EnrolmentRequest message **MSG_ENRREQ_IUT**.

Table 9: Authorization certificate issued by test system

```
CERT_AUTH_TS :=
Certificate {
    containing version_and_type
        set to 'explicit_certificates'(2)
    containing unsigned_certificate
        containing subject_type
            set to 'sec_data_exch_csr'
        containing cf
            indicating 'use_start_validity' and 'lifetime_is_duration'
        containing signer_id
            set to 8-byte hash of the CERT_AA
        containing signature_alg
            set to 'ecdsa_nistp256_with_sha256'
        containing scope
            containing name
                set to 'AC_SCOPE_DEFAUL'
            containing permitted_subject_types
                set to MSG_AUTHREQ_IUT.unsigned_csr
                        .type_specific_data.sec_data_exch_ca_scope.permitted_subject_types
            containing permissions
                set to MSG_AUTHREQ_IUT.unsigned_csr
                        .type_specific_data.sec_data_exch_ca_scope.permissions
            containing region
                set to MSG_AUTHREQ_IUT.unsigned_csr
                        .type_specific_data.sec_data_exch_ca_scope.region
        containing expiration
        containing lifetime
        containing crl_series
            set to 0
        containing verification_key
            set to MSG_AUTHREQ_IUT.unsigned_csr.verification_key
    containing signature
        containing ecdsa_signature
            verifiable with CERT_EA.verification_key
            containing R
                containing type
                    set to 'compressed_y_0' or 'compressed_y_1'
                containing x/y
                    set to a valid key for ECDSA-256
}
```

NOTE: This certificate is a response to the AuthorizationRequest message **MSG_AUTHREQ_IUT**.

6.1.8.2.2.2 Certificates issued by implementation under test

Table 10: Enrolment certificate issued by IUT

```
CERT_ENR_IUT :=
Certificate {
    containing version_and_type
        set to explicit_certificates(2)
    containing unsigned_certificate
        containing subject_type
            set to MSG_ENRREQ_TS.unsigned_csr.subject_type
        containing cf
            set to MSG_ENRREQ_TS.unsigned_csr.cf
        containing signer_id
            set to 8-byte hash of the CERT_EA
        containing signature_alg
            set to 'ecdsa_nistp256_with_sha256'
        containing scope
            containing name
            containing permitted_subject_types
                set to MSG_ENRREQ_TS.unsigned_csr
                            .type_specific_data.sec_data_exch_ca_scope.permitted_subject_types
            containing permissions
                containing type set to 'specified'
                containing permissions_list
                    set to the intersection between
                        MSG_ENRREQ_TS.unsigned_csr
                            .type_specific_data.sec_data_exch_ca_scope.permissions
                        and CERT_EA.scope.permissions.permissions_list
            containing region
                containing region_type set to 'circle'
                containing circular_region
                    set to the intersection between
                        MSG_ENRREQ_TS.unsigned_csr.type_specific_data.sec_data_exch_ca_scope.region
                        and CERT_EA.scope.region.circular_region
        containing expiration
            set to any timestamp > CLT
        containing lifetime if cf has use_start_validity and lifetime_is_duration flags set
            set to  any value > expiration - CLT
        containing start_validity if cf indicating use_start_validity but not lifetime_is_duration
            set to  any timestamp < CLT
        containing crl_series
        containing verification_key
            set to MSG_ENRREQ_TS.unsigned_csr.verification_key
    containing signature
        containing ecdsa_signature
            verifiable with CERT_EA.verification_key
}
```

NOTE: This certificate is a response to the EnrolmentRequest message **MSG_ENRREQ_TS**.

Table 11: Authorization certificate issued by IUT

```
CERT_AUTH_IUT :=
Certificate {
    containing version_and_type
        set to 'explicit_certificates'(2)
    containing unsigned_certificate
        containing subject_type
            set to MSG_AUTHREQ_TS.unsigned_csr.subject_type
        containing cf
            set to MSG_AUTHREQ_TS.unsigned_csr.cf
        containing signer_id
            set to 8-byte hash of the CERT_AA
        containing signature_alg
            set to 'ecdsa_nistp256_with_sha256'
        containing type_specific_data
            containing anonymous_scope if subject_type set to 'sec_data_exch_anonymous'
                containing permissions
                    containing type set to 'specified'
                    containing permissions_list
                        set to the intersection between MSG_AUTHREQ_TS.unsigned_csr
                                .type_specific_data.sec_data_exch_ca_scope.permissions
                        and CERT_AA.scope.permissions.permissions_list
                containing region
                    containing region_type set to 'circle'
                    containing circular_region
                        set to the intersection between MSG_AUTHREQ_TS.unsigned_csr
                                .type_specific_data.sec_data_exch_ca_scope.region
                        and CERT_AA.scope.region.circular_region
            or containing id_scope if subject_type set to 'sec_data_exch_anonymous'
                containing name[0..32]
                containing permitted_subject_types
                    set to MSG_AUTHREQ_TS.unsigned_csr
                                .type_specific_data.sec_data_exch_ca_scope.permitted_subject_types
                containing permissions
                    containing type set to 'specified'
                    containing permissions_list
                        set to the intersection between MSG_AUTHREQ_TS.unsigned_csr
                                .type_specific_data.sec_data_exch_ca_scope.permissions
                        and CERT_AA.scope.permissions.permissions_list
                containing region
                    containing region_type set to 'circle'
                    containing circular_region
                        set to the intersection between MSG_AUTHREQ_TS.unsigned_csr
                                .type_specific_data.sec_data_exch_ca_scope.region
                        and CERT_AA.scope.region.circular_region
        containing expiration
            set to any timestamp > CLT
        containing lifetime if cf has use_start_validity and lifetime_is_duration flags set
            set to  any value > expiration - CLT
        containing start_validity if cf indicating use_start_validity but not lifetime_is_duration
            set to  any timestamp < CLT
        containing crl_series
        containing verification_key
            set to MSG_AUTHREQ_TS.unsigned_csr.verification_key
    containing signature
        containing ecdsa_signature
            verifiable using CERT_AA.verification_key
}
```

NOTE: This certificate is a response to the AuthorizationRequest message **MSG_AUTHREQ_TS**.

6.1.8.3 Messages

6.1.8.3.1 ITS station testing

6.1.8.3.1.1 Enrolment

Table 12: EnrolmentRequest message received by the test system from the ITS-S

```
MSG_ENRREQ_IUT :=
CertificateRequest{
    containing signer
        containing type
            set to  'certificate' or
                    'certificate_chain' or
                    'self'
        containing certificate if signer.type set to 'certificate' or
        containing certificates if signer.type set to 'certificate_chain'
    containing unsigned_csr
        containing version_and_type
            set to 'explicit_certificates'(2)
        containing request_time
            set to any timestamp <= CLT
        containing subject_type
            set to  'sec_data_exch_csr'
        containing cf
            not indicating 'encryption_key' flag
        containing type_specific_data
            containing sec_data_exch_ca_scope
                containing name [0..32]
                containing permitted_subject_types
                    set to  array[1] := {
                        'sec_data_exch_anonymous' or 'sec_data_exch_identified_localized'
                    }
                containing permission
                    containing type
                        set to 'specified'
                    containing permissions_list
                containing region
                    containing region_type
                        set to 'circle'
                    containing circular_region
        containing expiration
            set to any timestamp > CLT
        containing lifetime if cf indicating 'use_start_validity' and 'lifetime_is_duration'
        containing start_validity if cf indicating 'use_start_validity'
                                and not indicating 'lifetime_is_duration'
            set to any timestamp < expiration
        containing verification_key
            containing algorithm set to 'ecdsa_nistp256_with_sha256'
            containing public_key
        containing response_encryption_key
            containing algorithm set to 'ecies_nistp256'
            containing supported_symm_alg set to 'aes_128_ccm'
            containing public_key
    containing signature
        containing ecdsa_signature
            verifiable using {
                signer.certificate.unsigned_certificate.verification_key
                        if signer.type is 'certificate'
                or signer.certificates[last].unsigned_certificate.verification_key
                        if signer.type is 'certificate_chain'
                or unsigned_csr.verification_key
                        if signer.type is 'self'
            }
}
```

Table 13: EnrolmentResponse message sent by the test system to the ITS-S

```
MSG_ENRRSP_TS :=
CertificateResponse {
    containing f
        set to 'NotRequested' (0)
    containing certificate_chain
        set to array[] = {
            CERT_ROOT,
            CERT_EA,
            CERT_ENR_TS
        }
    containing crl_path
        set to length 0
}
```

Table 14: EnrolmentRequestError message sent by the test system to the ITS-S

```
MSG_ENRERR_TS :=
CertificateRequestError {
    containing signer.type
        set to 'certificate'
    containing signer.certificate
        set to CERT_EA
    containing request_hash
        set to HASH(MSG_ENRREQ_IUT)
    containing reason
    containing signature
        containing ecdsa_signature
            verifiable using CERT_EA.unsigned_certificate.verification_key
}
```

6.1.8.3.1.2 Authorization

Table 15: AuthorizationRequest message received by the test system from the ITS-S

```
MSG_AUTHREQ_IUT :=
CertificateRequest{
    containing signer
        containing type
            set to  'certificate' or
                    'certificate_chain'
        containing certificate if signer.type set to 'certificate' or
        containing certificates if signer.type set to 'certificate_chain'
    containing unsigned_csr
        containing version_and_type
            set to 'explicit_certificates'(2)
        containing request_time
            set to any timestamp <= CLT
        containing subject_type
            set to 'sec_data_exch_anonymous' or 'sec_data_exch_identified_localized'
        containing cf
            not indicating 'encryption_key' flag
        containing type_specific_data
            containing anonymous_scope if subject_type set to 'sec_data_exch_anonymous'
                containing permissions
                    containing type
                        set to 'specified'
                    containing permissions_list
                containing region
                    containing region_type
                        set to 'circle'
                    containing circular_region
            or containing id_scope if subject_type set to 'sec_data_exch_identified_localized'
                containing name [0..32]
                containing permissions
                    containing type
                        set to 'specified'
                    containing permissions_list
                containing region
                    containing region_type
                        set to 'circle'
                    containing circular_region
        containing expiration
            set to any timestamp > CLT
        containing lifetime if cf indicating 'use_start_validity' and 'lifetime_is_duration'
        containing start_validity if cf indicating 'use_start_validity'
                                and not indicating 'lifetime_is_duration'
            set to any timestamp < expiration
        containing verification_key
            containing algorithm set to 'ecdsa_nistp256_with_sha256'
            containing public_key
        containing response_encryption_key
            containing algorithm set to 'ecies_nistp256'
            containing supported_symm_alg set to 'aes_128_ccm'
            containing public_key
    containing signature
        containing ecdsa_signature
            verifiable using CERT_ENR_TS.unsigned_certificate.verification_key
}
```

Table 16: EnrolmentResponse message received by the test system from the EA

```
MSG_AUTHRSP_TS :=
CertificateResponse {
    containing f
        set to 'NotRequested' (0)
    containing certificate_chain
        set to array[] = {
            CERT_ROOT,
            CERT_AA,
            CERT_AUTH_TS
        }
    containing crl_path
        set to length 0
}
```

Table 17: EnrolmentRequestError message sent by the test system to the ITS-S

```
MSG_AUTHERR_TS :=
CertificateRequestError {
    containing signer.type
        set to 'certificate'
    containing signer.certificate
        set to CERT_AA
    containing request_hash
        set to HASH(MSG_AUTHREQ_IUT)
    containing reason
    containing signature
        containing ecdsa_signature
            verifiable using CERT_AA.unsigned_certificate.verification_key
}
```

6.1.8.3.1.3 Send and Recive Data

Table 18: 1609Dot2Data message to be sent by the test system to the ITS-S under test

```
MSG_SIGNED_TS :=
Structure 1609Dot2Data {
    containing protocol_version
        set to 2
    containing type
        set to 'signed'
    containing signed_data
        containing signer
        containing unsigned_data
            containing psid
            containing data
        containing signature
}
```

Table 19: 1609Dot2Data message received by the test system from the ITS-S under test

```
MSG_SIGNED_IUT :=
Structure 1609Dot2Data {
    containing protocol_version
        set to 2
    containing type
        set to 'signed'
        or set to 'signed_partial_payload'
        or set to 'signed_external_payload'
    containing signed_data
        containing signer
        containing unsigned_data
            containing psid
        containing signature
            verifiable using signer
}
```

6.1.8.3.2 Enrolment Authority testing

Table 20: EnrolmentRequest message sent by the test system to the EA

```
MSG_ENRREQ_TS :=
CertificateRequest {
    containing signer
        containing type
            set to 'certificate'
        containing certificate
            set to CERT_ROOT
    containing unsigned_csr
        containing version_and_type
            set to 'explicit_certificates'(2)
        containing request_time
            set to CLT
        containing subject_type
            set to 'sec_data_exch_csr'
        containing cf
            indicating 'use_start_validity' and 'lifetime_is_duration'
        containing type_specific_data
            containing sec_data_exch_ca_scope
                containing name
                    set to 'EC_SCOPE_DEFAULT'
                containing permitted_subject_types
                    set to array[1]
                        containing 'sec_data_exch_identified_localized'
                containing permission
                    containing type
                        set to 'specified'
                    containing permissions_list
                        set to array[1]
                            containing PSID_A
                containing region
                    set to REGION_SMALL
        containing expiration
            set to 31. Dec 2020
        containing lifetime
            set to 10Y
        containing verification_key
            containing algorithm
                set to 'ecdsa_nistp256_with_sha256'
            containing public_key
                containing type
                    set to 'x_coordinate_only'
                containing x
                    set to a valid key for ECDSA-256
        containing response_encryption_key
            containing algorithm
                set to 'ecies_nistp256'
            containing supported_symm_alg
                set to 'aes_128_ccm'
            contains public_key
                contains type
                    set to 'x_coordinate_only'
                containing x
                    set to a valid key for ECIES-256
    containing signature
        containing ecdsa_signature
            verifiable by signer.certificate.unsigned_certificate.verification_key
}
```

Table 21: EnrolmentResponse message received by the test system from the EA

```
MSG_ENRRSP_IUT :=
CertificateResponse {
    containing f
    containing certificate_chain
        set to array[3]
            containing CERT_ROOT
            containing CERT_EA
            containing CERT_ENR_IUT
}
```

Table 22: EnrolmentRequestError message received by the test system from the EA

```
MSG_ENRERR_IUT :=
CertificateRequestError {
    containing signer.type
        set to 'certificate'
    containing signer.certificate
        set to CERT_EA
    containing request_hash
        set to HASH(MSG_ENRREQ_TS)
    containing reason
    containing signature
        containing ecdsa_signature
            verifiable using CERT_EA.unsigned_certificate.verification_key
}
```

6.1.8.3.3 Authorization Authority testing

Table 23: AuthorizationRequest message to be sent by the test system to the AA

```
MSG_AUTHREQ_TS :=
CertificateRequest{
    containing signer
        containing type
            set to 'certificate_chain'
        containing certificates
            set to array[3]
                containing CERT_ROOT
                containing CERT_EA
                containing CERT_ENR_IUT
    containing unsigned_csr
        containing version_and_type
            set to 'explicit_certificates'(2)
        containing request_time
            set to CLT
        containing subject_type
            set to 'sec_data_exch_identified_localized'
        containing cf
            indicating 'use_start_validity' and 'lifetime_is_duration'
        containing type_specific_data
            containing id_scope
                containing name
                    set to 'AC_SCOPE_DEFAULT'
                containing permissions
                    containing type
                        set to 'specified'
                    containing permissions_list
                        set to array[1]
                            containing PSID_A
                containing region
                    containing region_type
                        set to 'circle'
                    containing circular_region
                        set to REGION_SMALL
        containing expiration
            set to '31 Dec 2020'
        containing lifetime
            set to '10Y'
        containing verification_key
            containing algorithm
                set to 'ecdsa_nistp256_with_sha256'
            containing public_key
                containing type
                    set to 'x_coordinate_only'
                containing x
                    set to a valid key for ECDSA-256
        containing response_encryption_key
            containing algorithm
                set to 'ecies_nistp256'
            containing supported_symm_alg
                set to 'aes_128_ccm'
            contains public_key
                contains type
                    set to 'x_coordinate_only'
                containing x
                    set to a valid key for ECIES-256
    containing signature
        containing ecdsa_signature
            verifiable by signer.certificate.unsigned_certificate.verification_key
}
```

Table 24: AuthorizationResponse message received by the test system from the AA

```
MSG_AUTHRSP_IUT :=
CertificateResponse {
    containing f
    containing certificate_chain
        set to array[3]
            containing CERT_ROOT
            containing CERT_AA
            containing CERT_AUTH_IUT
}
```

Table 25: AuthorizationRequestError message received by the test system from the AA

```
MSG_AUTHERR_IUT :=
CertificateRequestError {
    containing signer.type
        set to 'certificate'
    containing signer.certificate
        set to CERT_AA
    containing request_hash
        set to HASH(MSG_AUTHREQ_IUT)
    containing reason
    containing signature
        containing ecdsa_signature
            verifiable using CERT_AA.unsigned_certificate.verification_key
}
```

6.2 Test purposes for SECURITY

6.2.1 ITS Station

6.2.1.1 Enrolment

6.2.1.1.1 Normal Behaviour

6.2.1.1.1.1 Enrolment Request verification

<table>
<tr><td><b>TP Id</b></td><td><b>TP/SEC/ITS-S/ENR/NB-01</b></td></tr>
<tr><td><b>Summary</b></td><td>Check that ITS-S generates correctly a generic EnrolmentRequest message</td></tr>
<tr><td><b>Reference</b></td><td>IEEE P1609.2/D12 [1], 6.3.33
ETSI TS 102 941 [2] Table 1 : Contents of ITS-S EnrolmentRequest message</td></tr>
<tr><td><b>Config Id</b></td><td>CF03</td></tr>
<tr><td><b>PICS Selection</b></td><td></td></tr>
<tr><td colspan="2"><b>Initial conditions</b></td></tr>
<tr><td colspan="2">with {
    the IUT in 'NotEnrolled' state
}</td></tr>
<tr><td colspan="2"><b>Expected behaviour</b></td></tr>
<tr><td colspan="2">ensure that {
    when {
        the IUT is requested to send an EnrolmentRequest message
    }
    then {
        the IUT sends a valid CertificateRequest set to <b>MSG_ENRREQ_IUT</b>
    }
}</td></tr>
</table>

<table>
<tr><td colspan="2"><b>TP Id</b></td><td><b>TP/SEC/ITS-S/ENR/NB-02-X</b></td></tr>
<tr><td colspan="2"><b>Summary</b></td><td>Check that ITS-S generates enrolment request with signature of different types</td></tr>
<tr><td colspan="2"><b>Reference</b></td><td>IEEE P1609.2/D12 [1], 6.2.17
ETSI TS 102 941 [2] Table 1 : Contents of ITS-S EnrolmentRequest message</td></tr>
<tr><td colspan="2"><b>Config Id</b></td><td>CF03</td></tr>
<tr><td colspan="2"><b>PICS Selection</b></td><td></td></tr>
<tr><td colspan="3"><b>Initial conditions</b></td></tr>
<tr><td colspan="3">with {
    the IUT in 'NotEnrolled' state
    the IUT is configured to use signature of form <b>X_PKT_SIGNATURE</b>
}</td></tr>
<tr><td colspan="3"><b>Expected behaviour</b></td></tr>
<tr><td colspan="3">ensure that {
    when {
        the IUT is requested to send an EnrolmentRequest message
    }
    then {
        the IUT sends a valid CertificateRequest set to <b>MSG_ENRREQ_IUT</b>
            containing signature.ecdsa_signature
                containing R.type
                    set to <b>X_PKT_SIGNATURE</b>
    }
}</td></tr>
<tr><td colspan="3"><b>Variants</b></td></tr>
<tr><td><b>X</b></td><td><b>PIC</b></td><td><b>X_PKT_SIGNATURE</b></td></tr>
<tr><td>A</td><td>PIC_Generate_XCoordinateOnlyKey</td><td>x_coordinate_only</td></tr>
<tr><td>B</td><td>PIC_Generate_CompressedKey</td><td>compressed_lsb_y_0/1</td></tr>
<tr><td>C</td><td>PIC_Generate_UncompressedKey</td><td>uncompressed</td></tr>
</table>

TP Id	**TP/SEC/ITS-S/ENR/NB-03**
Summary	Check that ITS-S generates enrolment request with signature calculated using compressed representation of all public keys
Reference	IEEE P1609.2/D12 [1], 6.2.17 ETSI TS 102 941 [2], Table 1 : Contents of ITS-S EnrolmentRequest message
Config Id	CF03
PICS Selection	PIC_Generate_UncompressedKey

Initial conditions

```
with {
    the IUT in 'NotEnrolled' state
    the IUT is configured to use uncompressed public keys for verification_key
    the IUT is configured to use uncompressed public keys for response_encryption_key
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send an EnrolmentRequest message
    }
    then {
        the IUT sends a valid CertificateRequest set to MSG_ENRREQ_IUT
            containing unsigned_csr.verification_key.public_key.type (V_PKT_VK)
                set to 'uncompressed'
            containing unsigned_csr.response_encryption_key.public_key.type (V_PKT_REK)
                set to 'uncompressed'
            containing signature.ecdsa_signature
                calculated using compressed representation of V_PKT_VK and V_PKT_REK
    }
}
```

TP Id	**TP/SEC/ITS-S/ENR/NB-04**
Summary	Check that ITS-S generates valid self-signed enrolment request.
Reference	IEEE P1609.2 [1], clause 6.2.17 ETSI TS 102 941 [2], see table 1
Config Id	CF03
PICS Selection	PIC_Generate_SelfSigned

Initial conditions

```
with {
    the IUT in 'NotEnrolled' state
    the IUT is configured to use a self-signed enrolment request
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send an EnrolmentRequest message
    }
    then {
        the IUT sends a valid CertificateRequest set to MSG_ENRREQ_IUT
            containing signer.type
                set to 'self'
            containing signature
                verified using unsigned_csr.verification_key
    }
}
```

TP Id	**TP/SEC/ITS-S/ENR/NB-05**
Summary	Check that ITS-S generates valid enrolment request with a different response_encryption_key for every request.
Reference	IEEE P1609.2/D12 [1], clause 6.3.34 ETSI TS 102 941 [2], see table 1
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in 'NotEnrolled' state
}
```

Expected behaviour

```
ensure that {
    when {
        each time the IUT is requested to send an EnrolmentRequest message
    }
    then {
        the IUT sends a valid CertificateRequest set to MSG_ENRREQ_IUT
            containing unsigned_csr.response_encryption_key
                set to value different from the previous ones
    }
}
```

TP Id	**TP/SEC/ITS-S/ENR/NB-06**
Summary	Check that ITS-S generates valid enrolment request with a certificate containing more than 8 PSID entries
Reference	IEEE P1609.2/D12 [1], clause 6.3.34 ETSI TS 102 941 [2], see table 1
Config Id	CF03
PICS Selection	PIC_Generate_PsidArrayWithMoreThan8Entries

Initial conditions

```
with {
    the IUT in 'NotEnrolled' state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send an EnrolmentRequest message with more than 8 PSID entries
    }
    then {
        the IUT sends a valid CertificateRequest set to MSG_ENRREQ_IUT
            containing unsigned_csr.type_specific_data.permission.permissions_list
                containing more than 8 entries
    }
}
```

6.2.1.1.1.2 Enrolment Response acceptance

TP Id	**TP/SEC/ITS-S/ENR/NB-07**
Summary	Check that ITS-S correctly decrypts enrolment response.
Reference	IEEE P1609.2/D12 [1], clause 5.6.2.1
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse (EnrolmentResponse)
    }
    then {
        the IUT decrypts the response
    }
}
```

TP Id	**TP/SEC/ITS-S/ENR/NB-08**
Summary	Check that the ITS-S accepts a valid enrolment response having correct fields and values.
Reference	IEEE P1609.2/D12 [1], clause 5.6.2.2 ETSI TS 102 941 [2], see table 2
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT having sent an EnrolmentRequest set to MSG_ENRREQ_IUT
        containing unsigned_csr.type_specific_data.sec_data_exch_ca_scope
            containing permissions.permissions_list (V_PERM_LIST)
    the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateResponse (EnrolmentResponse) set to MSG_ENRRSP_TS
            containing response.certificate_chain[last]
                containing unsigned_certificate.scope.permissions.permissions_list
                    set to V_PERM_LIST
    }
    then {
        the IUT accepts the CertificateResponse
    }
}
```

<table>
<tr><th>TP Id</th><th>TP/SEC/ITS-S/ENR/NB-09</th></tr>
<tr><td><b>Summary</b></td><td>Check that the ITS-S accepts a valid enrolment response even if the permissions in the issued certificate are a subset of requested permissions</td></tr>
<tr><td><b>Reference</b></td><td>IEEE P1609.2/D12 [1], clause 5.6.2.2
ETSI TS 102 941 [2], see table 2</td></tr>
<tr><td><b>Config Id</b></td><td>CF03</td></tr>
<tr><td><b>PICS Selection</b></td><td></td></tr>
<tr><td colspan="2"><b>Initial conditions</b></td></tr>
<tr><td colspan="2">with {
    the IUT having sent an EnrolmentRequest set to <b>MSG_ENRREQ_IUT</b>
        containing unsigned_csr.type_specific_data.sec_data_exch_ca_scope
            containing permissions.permissions_list (<b>V_PERM_LIST</b>)
    the IUT awaiting EnrolmentResponse
}</td></tr>
<tr><td colspan="2"><b>Expected behaviour</b></td></tr>
<tr><td colspan="2">ensure that {
    when {
        the IUT receives a valid CertificateResponse (EnrolmentResponse) set to <b>MSG_ENRRSP_TS</b>
            containing response.certificate_chain[last].unsigned_certificate.scope.permissions.permissions_list
                set to a subset of <b>V_PERM_LIST</b>
    }
    then {
        the IUT accepts the CertificateResponse
    }
}</td></tr>
</table>

TP Id	**TP/SEC/ITS-S/ENR/NB-10-X**
Summary	Check that ITS-S accepts enrolment response with different public key types
Reference	IEEE P1609.2/D12 [1], clause 6.2.17 ETSI TS 102 941 [2], see table 2
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateResponse (EnrolmentResponse) set to MSG_ENRRSP_TS
            containing certificate_chain[last]
                containing verification_key.public_key.type
                    set to X_PKT_VK
                containing signature.ecdsa_signature.R.type
                    set to X_PKT_SIGNATURE
    }
    then {
        the IUT accepts the CertificateResponse
    }
}
```

Variants

X	**X_PKT_SIGNATURE**	**X_PKT_VK**	**PIC Selection**
A	compressed_lsb_y_0/1	compressed_lsb_y_0/1	PIC_Verify_CompressedKeyKey
B	compressed_lsb_y_0/1	x_coordinate_only	PIC_Verify_CompressedKeyKey PIC_Verify_XCoordinateOnlyKey
C	compressed_lsb_y_0/1	uncompressed	PIC_Verify_UncompressedKey
D	x_coordinate_only	compressed_lsb_y_0/1	PIC_Verify_CompressedKeyKey PIC_Verify_XCoordinateOnlyKey
E	x_coordinate_only	x_coordinate_only	PIC_Verify_XCoordinateOnlyKey
F	x_coordinate_only	uncompressed	PIC_Verify_UncompressedKey PIC_Verify_XCoordinateOnlyKey
G	uncompressed	compressed_lsb_y_0/1	PIC_Verify_UncompressedKey PIC_Verify_CompressedKeyKey
H	uncompressed	x_coordinate_only	PIC_Verify_UncompressedKey PIC_Verify_XCoordinateOnlyKey
I	uncompressed	uncompressed	PIC_Verify_UncompressedKey

TP Id	**TP/SEC/ITS-S/ENR/NB-11**
Summary	Check that the ITS-S accepts a valid enrolment response with signature calculated using compressed representation of uncompressed public keys.
Reference	IEEE P1609.2/D12 [1], clause 6.2.17 ETSI TS 102 941 [2], see table 2
Config Id	CF03
PICS Selection	PIC_Verify_UncompressedKey
Initial conditions	

with {
    the IUT awaiting EnrolmentResponse
}	

Expected behaviour	

ensure that {
    when {
        the IUT receives a CertificateResponse (EnrolmentResponse) set to **MSG_ENRRSP_TS**
            containing certificate_chain[last]
                containing unsigned_certificate.verification_key.public_key.type (**V_PKT_VK**)
                    set to 'uncompressed'
                containing signature.ecdsa_signature
                    calculated using compressed representation of **V_PKT_VK**
    }
    then {
        the IUT accepts the CertificateResponse
    }
}	

TP Id	**TP/SEC/ITS-S/ENR/NB-12**
Summary	Check that the ITS-S accepts a valid enrolment response with start_validity and lifetime.
Reference	IEEE P1609.2/D12 [1], clause 6.3.2 ETSI TS 102 941 [2], see table 2
Config Id	CF03
PICS Selection	
Initial conditions	

with {
    the IUT awaiting EnrolmentResponse
}	

Expected behaviour	

ensure that {
    when {
        the IUT receives a CertificateResponse (EnrolmentResponse) set to **MSG_ENRRSP_TS**
            containing certificate_chain[last].unsigned_certificate
                containing cf
                    indicating use_start_validity
                    indicating lifetime_is_duration
                containing lifetime
                    set to '10Y'
    }
    then {
        the IUT accepts the CertificateResponse
    }
}	

<table>
<tr><th>TP Id</th><th>TP/SEC/ITS-S/ENR/NB-13</th></tr>
<tr><td><b>Summary</b></td><td>Check that the ITS-S accepts a valid enrolment response with start_validity value.</td></tr>
<tr><td><b>Reference</b></td><td>IEEE P1609.2/D12 [1], clause 6.3.2
ETSI TS 102 941 [2], see table 2</td></tr>
<tr><td><b>Config Id</b></td><td>CF03</td></tr>
<tr><td><b>PICS Selection</b></td><td>NOT PIC_Verify_LifetimeIsDuration</td></tr>
<tr><td colspan="2"><b>Initial conditions</b></td></tr>
<tr><td colspan="2">with {
    the IUT awaiting EnrolmentResponse
}</td></tr>
<tr><td colspan="2"><b>Expected behaviour</b></td></tr>
<tr><td colspan="2">ensure that {
    when {
        the IUT receives a valid CertificateResponse (EnrolmentResponse) set to <b>MSG_ENRRSP_TS</b>
            containing certificate_chain[last].unsigned_certificate
                containing cf
                    indicating 'use_start_validity'
                    and not indicating 'lifetime_is_duration'
                containing expiration
                containing start_validity
                    set to a timestamp < expiration
    }
    then {
        the IUT accepts the CertificateResponse
    }
}</td></tr>
</table>

6.2.1.1.1.3 Enrolment Request Error acceptance

<table>
<tr><th>TP Id</th><th>TP/SEC/ITS-S/ENR/NB-14</th></tr>
<tr><td><b>Summary</b></td><td>Check that ITS-S correctly decrypts enrolment request error.</td></tr>
<tr><td><b>Reference</b></td><td>IEEE P1609.2/D12 [1], clause 5.6.2.1</td></tr>
<tr><td><b>Config Id</b></td><td>CF03</td></tr>
<tr><td><b>PICS Selection</b></td><td></td></tr>
<tr><td colspan="2"><b>Initial conditions</b></td></tr>
<tr><td colspan="2">with {
    the IUT awaiting EnrolmentResponse
}</td></tr>
<tr><td colspan="2"><b>Expected behaviour</b></td></tr>
<tr><td colspan="2">ensure that {
    when {
        the IUT receives a CertificateRequestError (EnrolmentResponse)
    }
    then {
        the IUT decrypts the response
    }
}</td></tr>
</table>

TP Id	**TP/SEC/ITS-S/ENR/NB-15**
Summary	Check that the ITS-S accepts a valid enrolment request error having correct fields and values.
Reference	IEEE P1609.2/D12 [1], clause 5.6.2.2 ETSI TS 102 941 [2], see table 3
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT having sent an EnrolmentRequest (V_REQUEST) set to MSG_ENRREQ_IUT
    the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateRequestError (EnrolmentResponse) set to MSG_ENRERR_TS
            containing request_hash
                set to the hash of the V_REQUEST
                    calculated using compressed representation of all public keys
    }
    then {
        the IUT accepts the CertificateRequestError
    }
}
```

TP Id	**TP/SEC/ITS-S/ENR/NB-16-X**
Summary	Check that ITS-S accepts enrolment request error with various types of signature public keys.
Reference	IEEE P1609.2/D12 [1], clause 6.2.17 ETSI TS 102 941 [2], see table 3
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateRequestError (EnrolmentResponse) set to MSG_ENRERR_TS
            containing signature.ecdsa_signature.R.type
                set to X_PKT_SIGNATURE
    }
    then {
        the IUT accepts the CertificateRequestError
    }
}
```

Variants

X	**X_PKT_SIGNATURE**	**PIC Selection**
A	x_coordinate_only	PIC_Verify_XCoordinateOnlyKey
B	compressed_lsb_y_0/1	PIC_Verify_CompressedKey
C	uncompressed	PIC_Verify_UncompressedKey

6.2.1.1.2 Exceptional Behavior

<table>
<tr><th>TP Id</th><th>TP/SEC/ITS-S/ENR/EB-01</th></tr>
<tr><td><b>Summary</b></td><td>Check that ITS-S discards enrolment response if the subordinate certificate's validity region is large than the issuing certificate's validity region.</td></tr>
<tr><td><b>Reference</b></td><td>IEEE P1609.2/D12 [1], clause 5.5.3.3</td></tr>
<tr><td><b>Config Id</b></td><td>CF03</td></tr>
<tr><td><b>PICS Selection</b></td><td></td></tr>
<tr><td colspan="2"><b>Initial conditions</b></td></tr>
<tr><td colspan="2"><pre>with {
    the IUT awaiting EnrolmentResponse
    and the TS configured to use EA certificate <b>CERT_EA</b>
        containing unsigned_certificate.scope.region
            set to <b>REGION_SMALL</b>
}</pre></td></tr>
<tr><td colspan="2"><b>Expected behaviour</b></td></tr>
<tr><td colspan="2"><pre>ensure that {
    when {
        the IUT receives a CertificateResponse (EnrolmentResponse) set to <b>MSG_ENRRSP_TS</b>
            containing certificate_chain[last]   (<b>CERT_ENR_TS</b>)
                containing unsigned_certificate.scope.region
                    set to <b>REGION_LARGE</b>
    }
    then {
        the IUT discards the CertificateResponse
    }
}</pre></td></tr>
</table>

<table>
<tr><th>TP Id</th><th>TP/SEC/ITS-S/ENR/EB-02</th></tr>
<tr><td><b>Summary</b></td><td>Check that ITS-S discards enrolment response if the subordinate certificate's validity region is outside of the issuing certificate's validity region.</td></tr>
<tr><td><b>Reference</b></td><td>IEEE P1609.2/D12 [1], clause 5.5.3.3</td></tr>
<tr><td><b>Config Id</b></td><td>CF03</td></tr>
<tr><td><b>PICS Selection</b></td><td></td></tr>
<tr><td colspan="2"><b>Initial conditions</b></td></tr>
<tr><td colspan="2"><pre>with {
    the IUT awaiting EnrolmentResponse
    and the TS configured to use EA certificate <b>CERT_EA</b>
        containing unsigned_certificate.scope.region
            set to <b>REGION_SMALL</b>
}</pre></td></tr>
<tr><td colspan="2"><b>Expected behaviour</b></td></tr>
<tr><td colspan="2"><pre>ensure that {
    when {
        the IUT receives a CertificateResponse (EnrolmentResponse) set to <b>MSG_ENRRSP_TS</b>
            containing certificate_chain[last]   (<b>CERT_ENR_TS</b>)
                containing unsigned_certificate.scope.region
                    set to <b>REGION_OUTSIDE</b>
    }
    then {
        the IUT discards the CertificateResponse
    }
}</pre></td></tr>
</table>

TP Id	**TP/SEC/ITS-S/ENR/EB-03**
Summary	Check that ITS-S discards enrolment response if the subordinate certificate's validity period is longer than issuing certificate's validity period.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.3
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT awaiting EnrolmentResponse
    and the TS configured to use EA certificate CERT_EA
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse (EnrolmentResponse) set to MSG_ENRRSP_TS
            containing certificate_chain[last]   (CERT_ENR_TS)
                containing unsigned_certificate.expiration > CERT_EA.unsigned_certificate.expiration
    }
    then {
        the IUT discards the CertificateResponse
    }
}
```

TP Id	**TP/SEC/ITS-S/ENR/EB-04**
Summary	Check that ITS-S discards enrolment response if the subordinate certificate's permissions are not included in issuing certificate.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.3
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT has sent a valid EnrolmentRequest set to MSG_ENRREQ_IUT
        containing unsigned_csr.type_specific_data.sec_data_exch_ca_scope.permissions.permissions_list
            set to array [2]
                containing PSID_A
                containing PSID_B
    and the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse (EnrolmentResponse) set to MSG_ENRRSP_TS
            containing certificate_chain[last-1]    (CERT_EA)
                containing unsigned_certificate.scope.permissions.permissions_list
                    set to array[1]
                        containing PSID_A
            containing certificate_chain[last]    (CERT_ENR_TS)
                containing unsigned_certificate.scope.permissions.permissions_list
                    set to array[1]
                        containing PSID_B
    }
    then {
        the IUT discards the CertificateResponse
    }
}
```

TP Id	**TP/SEC/ITS-S/ENR/EB-05-X**
Summary	Check that ITS-S discards enrolment response if the message content type is different than 'encrypted'.
Reference	IEEE P1609.2/D12 [1], clause 5.6.2.1
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data structure
            containing type
                set to X_INVALID_CONTENT_TYPE
            containing encrypted_data.ciphertext
/---------------- After deciphering process ------------------
                containing type
                    set to 'certificate_response'
                containing response
                    set to MSG_ENRRSP_TS
/--------------------------------------------------------------------
    }
    then {
        the IUT discards the received message
    }
}
```

Variants

X	**X_INVALID_CONTENT_TYPE**
A	unsecured (0),
B	signed(1)
C	certificate_request(3)
D	certificate_response(4)
E	anonymous_certificate_response(5)
F	certificate_request_error(6)
G	crl_request(7)
H	crl(8)
I	signed_partial_payload(9)
J	signed_external_payload(10)
K	signed_wsa(11)
L	certificate_response_acknowledgment (12)
M	ANY_VALUE(128)

TP Id	**TP/SEC/ITS-S/ENR/EB-06-X**
Summary	Check that ITS-S discards enrolment response if the protocol_version is not 2.
Reference	IEEE P1609.2/D12 [1], clause 6.2.1
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data structure
            containing protocol_version
                set to X_INVALID_VERSION_NUMBER
            containing type
                set to 'encrypted'
            containing encrypted_data.ciphertext
/----------------- After deciphering process -------------------
                containing type
                    set to 'certificate_response'
                containing response
                    set to MSG_ENRRSP_TS
/----------------------------------------------------------------------
    }
    then {
        the IUT discards the received message
    }
}
```

Variants

X	**X_INVALID_VERSION_NUMBER**
A	0
B	1
C	3
D	255

TP Id	**TP/SEC/ITS-S/ENR/EB-07**
Summary	Check that ITS-S discards enrolment request error if the signer type is not valid.
Reference	IEEE P1609.2/D12 [1], clause 6.2.4
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data structure
            containing signed_data.signer.type
                set to 'self'
    }
    then {
        the IUT discards the received message
    }
}
```

TP Id	TP/SEC/ITS-S/ENR/EB-08-X
Summary	Check that ITS-S discards enrolment respond if the certificate is not an explicit one.
Reference	IEEE P1609.2/D12 [1], clause 6.3.1
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse (EnrolmentResponse) set to MSG_ENRRSP_TS
            containing certificate_chain[last].version_and_type
                set to X_INVALID_CERT_VERSION_AND_TYPE
    }
    then {
        the IUT discards the received message
    }
}
```

Variants

X	X_INVALID_CERT_VERSION_AND_TYPE
A	0
B	1
C	3
D	255

TP Id	TP/SEC/ITS-S/ENR/EB-09
Summary	Check that ITS-S discards enrolment response if the hash was not calculated using compressed representation of public keys.
Reference	IEEE P1609.2/D12 [1], clause 6.3.1
Config Id	CF03
PICS Selection	PIC_Verify_UncompressedKey

Initial conditions

```
with {
    the IUT awaiting EnrolmentResponse
    and the TS configured to use EA certificate CERT_EA
        containing unsigned_certificate.verification_key.public_key.type (V_PKT_VK_EA)
            set to 'uncompressed'
        containing unsigned_certificate.encryption_key.public_key.type (V_PKT_EK_EA)
            set to 'uncompressed'
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse (EnrolmentResponse) set to MSG_ENRRSP_TS
            containing certificate_chain[last]
                containing unsigned_certificate.signer_id
                    calculated using uncompressed representation of V_PKT_VK_EA and V_PKT_EK_EA
    }
    then {
        the IUT discards the received message
    }
}
```

TP Id	**TP/SEC/ITS-S/ENR/EB-10**
Summary	Check that ITS-S discards enrolment response without specified expiration time.
Reference	IEEE P1609.2/D12 [1], clause 6.3.2
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse (EnrolmentResponse) set to MSG_ENRRSP_TS
            containing certificate_chain[last].unsigned_certificate.expiration
                set to 0
    }
    then {
        the IUT discards the received message
    }
}
```

TP Id	**TP/SEC/ITS-S/ENR/EB-11**
Summary	Check that ITS-S discards enrolment response which includs PSIDs that are not specified in upper certificates.
Reference	IEEE P1609.2/D12 [1], clause 6.3.2
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse (EnrolmentResponse) set to MSG_ENRRSP_TS
            containing certificate_chain
                set to array with length > 1
            containing certificate_chain[last-1].unsigned_certificate.scope.permissions.permissions_list
                set to array[1]
                    containing PSID_A
            containing certificate_chain[last].unsigned_certificate.scope.permissions.permissions_list
                set to array[1]
                    containing PSID_B
    }
    then {
        the IUT discards the CertificateResponse
    }
}
```

TP Id	**TP/SEC/ITS-S/ENR/EB-12**
Summary	Check that ITS-S discards enrolment response if it has duplicated PSID.
Reference	IEEE P1609.2/D12 [1], clause 6.3.9
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT having sent an EnrolmentRequest set to MSG_ENRREQ_IUT
        containing unsigned_csr.type_specific_data.sec_data_exch_ca_scope
            containing permissions.permissions_list (V_PERM_LIST)
    the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse (EnrolmentResponse) set to MSG_ENRRSP_TS
        containing unsigned_certificate.scope.permissions.permissions_list
            set to array[2]
                containing V_PERM_LIST[0]
                containing V_PERM_LIST[0]
    }
    then {
        the IUT discards the received message
    }
}
```

TP Id	**TP/SEC/ITS-S/ENR/EB-13-X**
Summary	Check that ITS-S discards enrolment response if the latitude is less than –900 000 000 or greater than 900 000 000.
Reference	IEEE P1609.2/D12 [1], clause 6.3.18
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse (EnrolmentResponse) set to MSG_ENRRSP_TS
            containing certificate_chain[last].unsigned_certificate
                containing scope.region.circular_region.center.latitude
                    set to X_INVALID_LATITUDE
    }
    then {
        the IUT discards the received message
    }
}
```

Variants

X	**X_INVALID_LATITUDE**
A	900000001
B	-900000001

TP Id	**TP/SEC/ITS-S/ENR/EB-14-X**
Summary	Check that ITS-S discards enrolment response if the longitude is less than -1 800 000 000 or greater than 1 800 000 000.
Reference	IEEE P1609.2/D12 [1], clause 6.3.18
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT awaiting EnrolmentResponse
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse set to MSG_ENRRSP_TS
            containing certificate_chain[last].unsigned_certificate
                containing scope.region.circular_region.center.longitude
                    set to X_INVALID_LONGITUDE
    }
    then {
        the IUT discards the received message
    }
}
```

Variants

X	**X_INVALID_LONGITUDE**
A	1800000001
B	-1800000001

6.2.1.2 Authorization

6.2.1.2.1 Normal Behavior

TP Id	**TP/SEC/ITS-S/AUTH/NB-01**
Summary	Check that ITS-S generates correctly a generic AuthorizationRequest message.
Reference	ETSI TS 102 941 [2], see table 4
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send an AuthorizationRequest message
    }
    then {
        the IUT sends a valid CertificateRequest set to MSG_AUTHREQ_IUT
    }
}
```

TP Id	**TP/SEC/ITS-S/AUTH/NB-02-X**
Summary	Check that ITS-S generates authorization request with various signature types.
Reference	IEEE P1609.2/D12 [1], clause 6.2.17
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT is configured to use signature of type X_PKT_SIGNATURE
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send an AuthorizationRequest message
    }
    then {
        the IUT sends a valid CertificateRequest set to MSG_AUTHREQ_IUT
            containing signature.ecdsa_signature.R.type
                set to X_PKT_SIGNATURE
    }
}
```

Variants

X	**PIC Selection**	**X_PKT_SIGNATURE**
A	PIC_Generate_CompressedKey	compressed_lsb_y_0/1
B	PIC_Generate_XCoordinateOnlyKey	x_coordinate_only
C	PIC_Generate_UncompressedKey	uncompressed

TP Id	**TP/SEC/ITS-S/AUTH/NB-03**
Summary	Check that ITS-S generates valid authorization request with a certificate containing lifetime field when cf flag is set use_start_validity and lifetime_is_duration.
Reference	IEEE P1609.2/D12 [1], clause 6.3.2
Config Id	CF03
PICS Selection	PIC_Generate_StartValidity AND PIC_Generate_LifetimeIsDuration

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT is configured to use use_start_validity and lifetime_is_duration
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send an AuthorizationRequest message
    }
    then {
        the IUT sends a valid CertificateRequest set to MSG_AUTHREQ_IUT
            containing unsigned_csr
                containing cf
                    indicating 'use_start_validity'
                    indicating 'lifetime_is_duration'
                containing lifetime
    }
}
```

TP Id	**TP/SEC/ITS-S/AUTH/NB-04**
Summary	Check that ITS-S generates valid authorization request with a certificate containing start_validity field when cf flag is set use_start_validity.
Reference	IEEE P1609.2/D12 [1], clause 6.3.2
Config Id	CF03
PICS Selection	PIC_Generate_StartValidity AND NOT PIC_Generate_LifetimeIsDuration
Initial conditions	

with {
the IUT in Enrolled state
the IUT is configured to use 'use_start_validity' but not 'lifetime_is_duration'
}	

Expected behaviour	

ensure that {
when {
the IUT is requested to send an AuthorizationRequest message
}
then {
the IUT sends a valid CertificateRequest set to **MSG_AUTHREQ_IUT**
containing unsigned_csr
containing cf
indicating 'use_start_validity'
not indicating 'lifetime_is_duration'
containing start_validity
}
}	

TP Id	**TP/SEC/ITS-S/AUTH/NB-05**
Summary	Check that ITS-S generates valid authorization request with a CSR certificate with name of length > 0 and <= 32.
Reference	IEEE P1609.2/D12 [1], clause 6.3.19
Config Id	CF03
PICS Selection	
Initial conditions	

with {
the IUT in Enrolled state
}	

Expected behaviour	

ensure that {
when {
the IUT is requested to send an AuthorizationRequest message
}
then {
the IUT sends a valid CertificateRequest set to **MSG_AUTHREQ_IUT**
containing unsigned_csr.containing type_specific_data.id_scope.name
set to value of length > 0 and <= 32 or of length zero (see Note)
}
}	

NOTE: Value of length 0 is encoded as '00'.	

TP Id	TP/SEC/ITS-S/AUTH/NB-06
Summary	Check that ITS-S generates valid authorization request with a certificate containing more than 8 entries in the permissions_list field.
Reference	IEEE P1609.2/D12 [1], clause 6.3.9
Config Id	CF03
PICS Selection	PIC_Generate_PsidArrayWithMoreThan8Entries

Initial conditions

```
with {
    the IUT in Enrolled state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send an AuthorizationRequest message with more than 8 PSID entries
    }
    then {
        the IUT sends a valid CertificateRequest set to MSG_AUTHREQ_IUT
            containing unsigned_csr.type_specific_data.id_scope.permissions.permissions_list
                set to array with length > 8
    }
}
```

TP Id	TP/SEC/ITS-S/AUTH/NB-07-X
Summary	Check that ITS-S generates valid authorization request with a certificate containing 1 to 8 entries in the permissions_list field.
Reference	IEEE P1609.2/D12 [1], clause 6.3.23
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send an AuthorizationRequest message with X_N PSID items
    }
    then {
        the IUT sends a valid CertificateRequest set to MSG_AUTHREQ_IUT
            containing unsigned_csr.type_specific_data.id_scope.permissions.permissions_list
                set to array with length X_N
    }
}
```

Variants

X	X_N
A	1
B	4
C	8

TP Id	**TP/SEC/ITS-S/AUTH/NB-08**
Summary	Check that ITS-S generates valid authorization request with a valid hash.
Reference	IEEE P1609.2/D12 [1], clause 6.3.1
Config Id	CF03
PICS Selection	PIC_Generate_UncompressedKey

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has obtained an Enrolment Certificate (CERT_ENR_TS)
        containing unsigned_certificate.verification_key.public_key.type (V_PKT_VK_ENR)
            set to 'uncompressed'
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send an AuthorizationRequest message
    }
    then {
        the IUT sends a valid CertificateRequest set to MSG_AUTHREQ_IUT
            containing signer
                containing certificate or certificates[last]
                    containing unsigned_certificate.signer_id
                        calculated using compressed representation of V_PKT_VK_ENR
    }
}
```

TP Id	**TP/SEC/ITS-S/AUTH/NB-09**
Summary	Check that ITS-S generates valid authorization request with a valid signature.
Reference	IEEE P1609.2/D12 [1], clause 6.3.33
Config Id	CF03
PICS Selection	PIC_Generate_UncompressedKey

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT is configured to send requests with uncompressed verification_key
    the IUT is configured to send requests with uncompressed response_encryption_key
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send an AuthorizationRequest message
    }
    then {
        the IUT sends a valid CertificateRequest set to MSG_AUTHREQ_IUT
            containing unsigned_csr.verification_key.public_key.type (V_PKT_VK)
                set to 'uncompressed'            containing unsigned_csr.response_encryption_key.public_key.type
(V_PKT_REK)
                set to 'uncompressed'
            containing signature.ecdsa_signature
                calculated using compressed representation of V_PKT_VK and V_PKT_REK
    }
}
```

TP Id	**TP/SEC/ITS-S/AUTH/NB-10**
Summary	Check that ITS-S generates valid authorization request with a different response_encryption_key for every request.
Reference	[1], clause 6.3.34
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
}
```

Expected behaviour

```
ensure that {
    when {
        each time the IUT is requested to send an AuthorizationRequest message
    }
    then {
        the IUT sends a valid CertificateRequest set to MSG_AUTHREQ_IUT
            containing unsigned_csr.response_encryption_key
                set to value <> from the previous ones
    }
}
```

TP Id	**TP/SEC/ITS-S/AUTH/NB-11**
Summary	Check that the ITS-S accepts a valid authorization response having correct fields and values.
Reference	ETSI TS 102 867 [3], clause 5.1.2.1, table 14
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateResponse (AuthorizationResponse) set to MSG_AUTHRSP_TS
    }
    then {
        the IUT accepts the CertificateResponse
    }
}
```

TP Id	**TP/SEC/ITS-S/AUTH/NB-12**
Summary	Check that the ITS-S accepts a valid authorization response having correct fields and values.
Reference	IEEE P1609.2/D12 [1], clause 5.6.2.2
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid CertificateRequest set to MSG_AUTHREQ_IUT
        containing unsigned_csr.type_specific_data.permission.permissions_list
            set to array
                containing PSID_A
                containing PSID_B

}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateResponse set to MSG_AUTHRSP_TS
            containing certificate_chain[last].unsigned_certificate.type_specific_data.ANY_SCOPE
                containing permissions.permissions_list
                    set to array
                        not containing PSID_A
    }
    then {
        the IUT accepts the CertificateResponse
    }
}
```

TP Id	**TP/SEC/ITS-S/AUTH/NB-13-X**
Summary	Check that the ITS-S accepts a valid authorization response signed by ecdsa_signature with different public key types.
Reference	IEEE P1609.2/D12 [1], clause 6.2.17
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateResponse (AuthorizationResponse) to MSG_AUTHRSP_TS
            containing certificate_chain[last].signature.ecdsa_signature.R
                containing type set to X_PKT_SIGNATURE
    }
    then {
        the IUT accepts the CertificateResponse
    }
}
```

Variants

X	**PIC Selection**	**X_PKT_SIGNATURE**
A	PIC_Verify_CompressedKey	compressed_lsb_y_0/1
B	PIC_Verify_XCoordinateOnlyKey	x_coordinate_only
C	PIC_Verify_UncompressedKey	uncompressed

TP Id	**TP/SEC/ITS-S/AUTH/NB-14**
Summary	Check that the ITS-S accepts a valid authorization response with start_validity.
Reference	IEEE P1609.2/D12 [1], clause 6.3.2
Config Id	CF03
PICS Selection	PIC_Verify_StartValidity AND PIC_Verify_LifetimeIsDuration

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateResponse set to MSG_AUTHRSP_TS
            containing certificate_chain[last].unsigned_certificate
                containing cf
                    indicating 'use_start_validity'
                    not indicating 'lifetime_is_duration'
                containing start_validity
                not containing lifetime
    }
    then {
        the IUT accepts the CertificateResponse
    }
}
```

6.2.1.2.2 Exceptional Behavior

TP Id	**TP/SEC/ITS-S/AUTH/EB-01-X**
Summary	Check that the ITS-S discards an authorization response having a non-permitted subject_type.
Reference	ETSI TS 102 867 [3], clause 5.1.2.1, IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 5.6.1.2
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse (AuthorizationResponse) set to MSG_AUTHRSP_TS
            containing certificate_chain[last].unsigned_certificate.subject_type
                set to X_INVALID_SUBJECT_TYPE
    }
    then {
        the IUT discards the CertificateResponse
    }
}
```

Variants

X	**X_INVALID_SUBJECT_TYPE**
A	sec_data_exch_identified_not_localized (1)
B	sec_data_exch_csr (3)
C	wsa (4)
D	wsa_csr (5)
E	sec_data_exch_ca(6)
F	wsa_ca (7)
H	crl_signer(8)
I	root_ca (255)
G	ANY OTHER (128)

TP Id	**TP/SEC/ITS-S/AUTH/EB-02-X**
Summary	Check that the ITS-S discards an authorization response having a non-permitted cf.
Reference	ETSI TS 102 867 [3], clause 5.1.2.1, IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 5.6.1.2
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse set to MSG_AUTHRSP_TS
            containing certificate_chain[last].unsigned_certificate.cf
                indicating X_INVALID_CONTENT_FLAGS
    }
    then {
        the IUT discards the CertificateResponse
    }
}
```

Variants

X	**X_INVALID_SUBJECT_TYPE**	**PIC Selection**
A	use_start_validity (0)	NOT PIC_Verify_StartValidity
B	encryption_key (2)	
C	any value (3)	

TP Id	**TP/SEC/ITS-S/AUTH/EB-03-X**
Summary	Check that the ITS-S discards an authorization response having a non-permitted PsidArray.type.
Reference	ETSI TS 102 867 [3], clause 5.1.2.1, IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 5.6.1.2
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse (AuthorizationResponse) set to MSG_AUTHRSP_TS
            containing certificate_chain[last].unsigned_certificate.type_specific_data.ANY_SCOPE.permissions.type
                set to a X_INVALID_PERM_TYPE
    }
    then {
        the IUT discards the CertificateResponse
    }
}
```

Variants

X	**X_INVALID_PERM_TYPE**
A	from_issuer (0)
B	Any value (3)
C	Any value (255)

TP Id	**TP/SEC/ITS-S/AUTH/EB-04**
Summary	Check that the ITS-S discards an authorization response requesting acknowledgement.
Reference	ETSI TS 102 867 [3], clause 5.1.2.1, IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 5.6.1.2
Config Id	CF03
PICS Selection	
Initial conditions	

with {
the IUT in Enrolled state
the IUT has sent a valid AuthorizationRequest set to **MSG_AUTHREQ_IUT**
}	

Expected behaviour	

ensure that {
when {
the IUT receives a CertificateResponse (AuthorizationResponse) set to **MSG_AUTHRSP_TS**
containing f
indicating 'Requested'
}
then {
the IUT discards the CertificateResponse
}
}	

TP Id	**TP/SEC/ITS-S/AUTH/EB-05**
Summary	Check that the ITS-S discards an authorization response that does not comply with the authorization request.
Reference	ETSI TS 102 867 [3], clause 5.1.2.1, IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 5.6.1.2
Config Id	CF03
PICS Selection	
Initial conditions	

with {
the IUT in Enrolled state
the IUT has sent a valid AuthorizationRequest set to **MSG_AUTHREQ_IUT**
}	

Expected behaviour	

ensure that {
when {
the IUT receives a CertificateResponse (AuthorizationResponse) set to **MSG_AUTHRSP_TS**
containing fields that does not comply with the authorization request
}
then {
the IUT discards the CertificateResponse
}
}	

TP Id	**TP/SEC/ITS-S/AUTH/EB-06**
Summary	Check that the ITS-S discards an authorization response error with incorrect signerIdentifier_type.
Reference	ETSI TS 102 867 [3], clause 5.1.2.1, IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 5.6.1.2
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequestError set to MSG_AUTHERR_TS
            containing signer.type
                set to 'self'
    }
    then {
        the IUT discards the CertificateRequestError
    }
}
```

TP Id	**TP/SEC/ITS-S/AUTH/EB-07-X**
Summary	Check that the ITS-S discards an authorization response error having a non-permitted subject_type.
Reference	ETSI TS 102 867 [3], clause 5.1.2.1, IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 5.6.1.2
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequestError set to MSG_AUTHERR_TS
            containing signer.certificates[last].unsigned_certificate.subject_type
                set to X_INVALID_SUBJECT_TYPE
    }
    then {
        the IUT discards the CertificateRequestError
    }
}
```

Variants

X	**X_INVALID_SUBJECT_TYPE**
A	sec_data_exch_identified_not_localized (1)
B	sec_data_exch_csr (3)
C	wsa (4)
D	wsa_csr (5)
E	sec_data_exch_ca(6)
F	wsa_ca (7)
H	crl_signer(8)
I	root_ca (255)
G	ANY OTHER (128)

TP Id	**TP/SEC/ITS-S/AUTH/EB-08**
Summary	Check that the ITS-S discards an authorization response having the subordinate certificate's validity region not wholly contained in the issuing certificate's validity region.
Reference	ETSI TS 102 867 [3], clause 5.1.2.1, IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 5.6.1.2
Config Id	CF03
PICS Selection	
Initial conditions	

with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to **MSG_AUTHREQ_IUT**
}	

Expected behaviour	

ensure that {
    when {
        the IUT receives a CertificateResponse set to **MSG_AUTHRSP_TS**
            containing certificate_chain[n].scope.region
                set to **REGION_SMALL**
            containing certificate_chain[n+1].scope.region
                set to **REGION_INTERSECTING**
    }
    then {
        the IUT discards the CertificateResponse
    }
}	

TP Id	**TP/SEC/ITS-S/AUTH/EB-09**
Summary	Check that the ITS-S discards an authorization response error having the subordinate certificate's validity region not wholly contained in the issuing certificate's validity region.
Reference	ETSI TS 102 867 [3], clause 5.1.2.1, IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 5.6.1.2
Config Id	CF03
PICS Selection	
Initial conditions	

with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to **MSG_AUTHREQ_IUT**
}	

Expected behaviour	

ensure that {
    when {
        the IUT receives a CertificateRequestError set to **MSG_AUTHERR_TS**
            containing signer
                containing certificates[n].scope.region
                    set to **REGION_SMALL**
                containing certificates[n+1].scope.region
                    set to **REGION_INTERSECTING**
    }
    then {
        the IUT discards the CertificateRequestError
    }
}	

TP Id	**TP/SEC/ITS-S/AUTH/EB-10**
Summary	Check that the ITS-S discards an authorization response having the subordinate certificate's validity region not within in the issuing certificate's validity region.
Reference	ETSI TS 102 867 [3], clause 5.1.2.1, IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 5.6.1.2
Config Id	CF03
PICS Selection	
Initial conditions	

with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to **MSG_AUTHREQ_IUT**
}	

Expected behaviour	

ensure that {
    when {
        the IUT receives a CertificateResponse set to **MSG_AUTHRSP_TS**
            containing certificate_chain[n].scope.region
                set to **REGION_SMALL**
            containing certificate_chain[n+1].scope.region
                set to **REGION_OUTSIDE**
    }
    then {
        the IUT discards the CertificateResponse
    }
}	

TP Id	**TP/SEC/ITS-S/AUTH/EB-11**
Summary	Check that the ITS-S discards an authorization response error having the subordinate certificate's validity region not within in the issuing certificate's validity region.
Reference	ETSI TS 102 867 [3], clause 5.1.2.1, IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 5.6.1.2
Config Id	CF03
PICS Selection	
Initial conditions	

with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to **MSG_AUTHREQ_IUT**
}	

Expected behaviour	

ensure that {
    when {
        the IUT receives a CertificateRequestError set to **MSG_AUTHERR_TS**
            containing signer
                containing certificates[n].scope.region
                    set to **REGION_SMALL**
                containing certificates[n+1].scope.region
                    set to **REGION_OUTSIDE**
    }
    then {
        the IUT discards the CertificateRequestError
    }
}	

TP Id	**TP/SEC/ITS-S/AUTH/EB-12**
Summary	Check that the ITS-S discards an authorization response having the subordinate certificate operational permissions are not a subset ofthe issuing certificate operational permissions.
Reference	ETSI TS 102 867 [3], clause 5.1.2.1, IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 5.6.1.2
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse set to MSG_AUTHRSP_TS
            containing certificate_chain[n].scope.permissions
                not indicating PSID_A
            containing certificate_chain[n+1].scope.permissions
                indicating PSID_A
    }
    then {
        the IUT discards the CertificateResponse
    }
}
```

TP Id	**TP/SEC/ITS-S/AUTH/EB-13**
Summary	Check that the ITS-S discards an authorization response error having the subordinate certificate operational permissions are not a subset ofthe issuing certificate operational permissions.
Reference	ETSI TS 102 867 [3] clause 5.1.2.1, IEEE P1609.2/D12 [1], 5.5.3.3, 5.6.1.2
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequestError set to MSG_AUTHERR_TS
            containing signer
                containing certificates[n].scope.permissions
                    not indicating PSID_A
                containing certificates[n+1].scope.permissions
                    indicating PSID_A
    }
    then {
        the IUT discards the CertificateRequestError
    }
}
```

TP Id	**TP/SEC/ITS-S/AUTH/EB-14-X**
Summary	Check that the ITS-S discards an authorization response encapsulated into 1609Dot2Data with protocol_version not egal to 2.
Reference	IEEE P1609.2/D12 [1], clause 6.2.1
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data structure
            containing protocol_version
                set to X_INVALID_VERSION_NUMBER
            containing type
                set to 'encrypted'
            containing encrypted_data.ciphertext
/----------------- After deciphering process -------------------
                containing type
                    set to 'certificate_response'
                containing request
                    set to MSG_AUTHRSP_TS
/----------------------------------------------------------------------
    }
    then {
        the IUT discards the CertificateResponse
    }
}
```

Variants

X	**X_INVALID_VERSION_NUMBER**
A	0
B	1
C	3
D	255

TP Id	**TP/SEC/ITS-S/AUTH/EB-15-X**
Summary	Check that the ITS-S discards an authorization request error encapsulated into 1609Dot2Data with protocol_version not egal to 2.
Reference	IEEE P1609.2/D12 [1], clause 6.2.1
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data structure
            containing protocol_version
                set to X_INVALID_VERSION_NUMBER
            containing type
                set to 'encrypted'
            containing encrypted_data.ciphertext
/----------------- After deciphering process -------------------
                containing type
                    set to 'certificate_request_error'
                containing request
                    set to MSG_AUTHERR_TS
/---------------------------------------------------------------------
    }
    then {
        the IUT discards the CertificateResponse
    }
}
```

Variants

X	**X_INVALID_VERSION_NUMBER**
A	0
B	1
C	3
D	255

TP Id	**TP/SEC/ITS-S/AUTH/EB-16**
Summary	Check that the ITS-S discards an authorization response with zero value in all expiration fields.
Reference	IEEE P1609.2/D12 [1], clause 6.3.2
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse (AuthorizationResponse) set to MSG_AUTHRSP_TS
            containing certificate_chain[last].unsigned_certificate
                containing expiration
                    set to 0
    }
    then {
        the IUT discards the CertificateResponse
    }
}
```

TP Id	**TP/SEC/ITS-S/AUTH/EB-17**
Summary	Check that the ITS-S discards an authorization response with duplicate PSIDs.
Reference	IEEE P1609.2/D12 [1], clause 6.3.9
Config Id	CF03
PICS Selection	
Initial conditions	

with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to **MSG_AUTHREQ_IUT**
}	

Expected behaviour	

ensure that {
    when {
        the IUT receives a CertificateResponse (AuthorizationResponse) set to **MSG_AUTHRSP_TS**
            containing certificate_chain[last].unsigned_certificate
                containing type_specific_data.**ANY_SCOPE**.permissions.permissions_list
                    set to array[2]
                        containing **PSID_A**
                        containing **PSID_A**
    }
    then {
        the IUT discards the CertificateResponse
    }
}	

TP Id	**TP/SEC/ITS-S/AUTH/EB-18-X**
Summary	Check that the ITS-S discards an authorization response with wrongly encoded latitude field.
Reference	IEEE P1609.2/D12 [1], clause 6.3.18
Config Id	CF03
PICS Selection	
Initial conditions	

with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to **MSG_AUTHREQ_IUT**
}	

Expected behaviour	

ensure that {
    when {
        the IUT receives a CertificateResponse (AuthorizationResponse) set to **MSG_AUTHRSP_TS**
            containing certificate_chain[last].unsigned_certificate
                containing scope.region.circular_region.center.latitude
                    set to **X_INVALID_LATITUDE**
    }
    then {
        the IUT discards the CertificateResponse
    }
}	

Variants	
X	**X_INVALID_LATITUDE**
A	900000001
B	-900000001

TP Id	TP/SEC/ITS-S/AUTH/EB-19-X
Summary	Check that the ITS-S discards an authorization response with wrongly encoded longitude field.
Reference	IEEE P1609.2/D12 [1], clause 6.3.18
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse set to MSG_AUTHRSP_TS
            containing certificate_chain[last].unsigned_certificate
                containing scope.region.circular_region.center.longitude
                    set to X_INVALID_LONGITUDE
    }
    then {
        the IUT discards the CertificateResponse
    }
}
```

Variants

X	X_INVALID_LONGITUDE
A	1800000001
B	-1800000001

TP Id	TP/SEC/ITS-S/AUTH/EB-20
Summary	Check that the ITS-S discards an authorization response with an empty PsidSspArray.
Reference	IEEE P1609.2/D12 [1], clause 6.3.23
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse (AuthorizationResponse) set to MSG_AUTHRSP_TS
            containing certificate_chain[last].unsigned_certificate
                containing ANY_SCOPE.permissions.permissions_list
                    set to array of length 0
    }
    then {
        the IUT discards the CertificateResponse
    }
}
```

TP Id	**TP/SEC/ITS-S/AUTH/EB-21**
Summary	Check that the ITS-S discards an authorization response with a certificate having a too long service_specific_permission field.
Reference	IEEE P1609.2/D12 [1], clause 6.3.24
Config Id	CF03
PICS Selection	

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateResponse set to MSG_AUTHRSP_TS
            containing certificate_chain[last].unsigned_certificate
                containing type_specific_data.ANY_SCOPE.permissions.permissions_list
                    set to array[1]
                        containing a PsidSpp (V_PSIDSSPP_A)
                            containing service_specific_permission
                                longer than 31 octets
                containing a service_specific_permission
                    having a length > 32 octets
    }
    then {
        the IUT discards the CertificateResponse
    }
}
```

TP Id	**TP/SEC/ITS-S/AUTH/EB-22**
Summary	Check that the ITS-S discards an authorization request error with having a wrongly calculated request_hash.
Reference	IEEE P1609.2/D12 [1], clause 6.3.1
Config Id	CF03
PICS Selection	PIC_Verify_UncompressedKey

Initial conditions

```
with {
    the IUT in Enrolled state
    the IUT has sent a valid AuthorizationRequest set to MSG_AUTHREQ_IUT
        containing unsigned_csr.verification_key.public_key.type (V_PKT_VK)
            set to 'uncompressed'
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequestError set to MSG_AUTHERR_TS
            containing request_hash
                calculated using uncompressed representation of V_PKT_VK
    }
    then {
        the IUT discards the CertificateRequestError
    }
}
```

6.2.1.3 Sending Data

TP Id	**TP/SEC/ITS-S/S-DATA/NB-01**
Summary	Check that ITS-S sends a correctly signed message with payload.
Reference	IEEE P1609.2/D12 [1], clause 6.2.7
Config Id	CF04
PICS Selection	PIC_Generate_SignPayload
Initial conditions	

with {
    the IUT in Authorized state
}	

Expected behaviour	

ensure that {
    when {
        the IUT is requested to send a signed message
    }
    then {
        the IUT sends a valid 1609Dot2Data set to **MSG_SIGNED_IUT**
    }
}	

TP Id	**TP/SEC/ITS-S/S-DATA/NB-02**
Summary	Check that ITS-S sends correctly signed message with partial payload.
Reference	IEEE P1609.2/D12 [1], clause 6.2.7
Config Id	CF04
PICS Selection	PIC_Generate_SignPartialPayload
Initial conditions	

with {
    the IUT in Authorized state
}	

Expected behaviour	

ensure that {
    when {
        the IUT is requested to send a signed message with partial data
    }
    then {
        the IUT sends a valid 1609Dot2Data set to **MSG_SIGNED_IUT**
            containing type
                set to 'signed_partial_payload'
            containing signed_data.unsigned_data
                containing data
    }
}	

TP Id	**TP/SEC/ITS-S/S-DATA/NB-03**
Summary	Check that ITS-S sends correctly signed message with external payload.
Reference	IEEE P1609.2/D12 [1], clause 6.2.7
Config Id	CF04
PICS Selection	PIC_Generate_SignExternalPayload

Initial conditions

```
with {
    the IUT in Authorized state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send a signed message with external data
    }
    then {
        the IUT sends a valid 1609Dot2Data set to MSG_SIGNED_IUT
            containing type
                set to 'signed_external_payload'
            containing signed_data.unsigned_data
                not containing data
    }
}
```

TP Id	**TP/SEC/ITS-S/S-DATA/NB-04**
Summary	Check that if ITS-S generates correctly a signed message containing the generation time.
Reference	IEEE P1609.2/D12 [1], clause 6.2.7
Config Id	CF04
PICS Selection	PIC_Generate_SignPayload AND PIC_Generate_GenerationTime

Initial conditions

```
with {
    the IUT in Authorized state and
    the IUT is configured to include generation time when signing a message
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send a signed message
    }
    then {
        the IUT sends a valid 1609Dot2Data set to MSG_SIGNED_IUT
            containing signed_data.unsigned_data
                containing tf
                    indicating 'use_generation_time'
                containing generation_time
    }
}
```

TP Id	**TP/SEC/ITS-S/S-DATA/NB-05**
Summary	Check that if ITS-S generates correctly multiple signed messages containing the generation time.
Reference	IEEE P1609.2/D12 [1], clause 6.2.7
Config Id	CF04
PICS Selection	PIC_Generate_SignPayload AND PIC_Generate_GenerationTime

Initial conditions

```
with {
    the IUT in Authorized state and
    the IUT is configured to include generation time when signing a message and
    the IUT has previously sent a signed message (V_MSG_0)
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send a new signed message
    }
    then {
        the IUT sends a valid 1609Dot2Data set to MSG_SIGNED_IUT
            containing signed_data.unsigned_data
                containing tf
                    indicating 'use_generation_time'
                containing generation_time
                    set to a value > V_MSG_0.signed_data.unsigned_data.generation_time and < CLT
    }
}
```

TP Id	**TP/SEC/ITS-S/S-DATA/NB-06**
Summary	Check that if ITS-S generates correctly a ToBeSignedData containing the expiry time.
Reference	IEEE P1609.2/D12 [1], clause 6.2.7
Config Id	CF04
PICS Selection	PIC_Generate_SignPayload AND PIC_Generate_ExpirationTime

Initial conditions

```
with {
    the IUT in Authorized state and
    the IUT is configured to include expiry_time when signing a message
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send a signed message
    }
    then {
        the IUT sends a valid 1609Dot2Data set to MSG_SIGNED_IUT
            containing signed_data.unsigned_data
                containing tf
                    indicating 'expires'
                containing expiry_time
    }
}
```

TP Id	**TP/SEC/ITS-S/S-DATA/NB-07**
Summary	Check that if ITS-S generates correctly a ToBeSignedData containing the generation location.
Reference	IEEE P1609.2/D12 [1], clause 6.2.7
Config Id	CF04
PICS Selection	PIC_Generate_SignPayload AND PIC_Generate_GenerationLocation

Initial conditions

```
with {
    the IUT in Authorized state and
    the IUT is configured to include generation_location when signing a message
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send a signed message
    }
    then {
        the IUT sends a valid 1609Dot2Data set to MSG_SIGNED_IUT
            containing signed_data.unsigned_data
                containing tf
                    indicating 'use_location'
                containing generation_location
    }
}
```

TP Id	**TP/SEC/ITS-S/S-DATA/NB-08**
Summary	Check that the ITS-S can generate valid signed data with ecdsa_nistp256_with_sha256.
Reference	IEEE P1609.2/D12 [1], clause 6.2.15
Config Id	CF04
PICS Selection	PIC_Generate_SignPayload AND PIC_Generate_Ecdsa256

Initial conditions

```
with {
    the IUT in Authorized state and
    the IUT is configured to use 'ecdsa_nistp256_with_sha256' as PKAlgorithm when signing a message
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send a signed message
    }
    then {
        the IUT sends a valid 1609Dot2Data set to MSG_SIGNED_IUT
            containing signed_data
                containing signer
                    containing type
                        set to 'certificate_digest_with_ecdsap256'
                    containing digest
                containing signature.algorithm
                    set to 'ecdsa ecdsa_nistp256_with_sha256'
    }
}
```

TP Id	**TP/SEC/ITS-S/S-DATA/NB-09**
Summary	Check that the ITS-S can generate valid signed data with ecdsa_nistp224_with_sha224.
Reference	IEEE P1609.2/D12 [1], clause 6.2.15
Config Id	CF04
PICS Selection	PIC_Generate_SignPayload AND PIC_Generate_Ecdsa224

Initial conditions

```
with {
    the IUT in Authorized state and
    the IUT is configured to use ecdsa_nistp224_with_sha224 as PKAlgorithm when signing a message
}
```

Expected behaviour

```
ensure that {
    when {      the IUT is requested to send a signed message
    }
    then {
        the IUT sends a valid 1609Dot2Data set to MSG_SIGNED_IUT
            containing signed_data
                containing signer
                    containing type
                        set to 'certificate_digest_with_ecdsap224'
                    containing digest
                containing signature
                    containing algorithm
                        set to 'ecdsa ecdsa_nistp224_with_sha224'
    }
}
```

TP Id	**TP/SEC/ITS-S/S-DATA/NB-10-X**
Summary	Check that ITS-S generates signed data with signature with different public key types.
Reference	IEEE P1609.2/D12 [1], clause 6.2.15
Config Id	CF04
PICS Selection	PIC_Generate_SignPayload

Initial conditions

```
with {
    the IUT in Authorized state
    the IUT is configured to sign messages using signatures with public key type of form X_PKT_SIGNATURE
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send a signed message
    }
    then {
        the IUT sends a valid 1609Dot2Data set to MSG_SIGNED_IUT
            containing signed_data
                containing signature.ecdsa_signature.R.type
                    set to X_PKT_SIGNATURE
    }
}
```

Variants

X	**PIC Selection**	**X_PKT_SIGNATURE**
A	PIC_Generate_CompressedKeyPublicKey	compressed_lsb_y_0 or compressed_lsb_y_1
B	PIC_Generate_XCoordinateOnlyPublicKey	x_coordinate_only
C	PIC_Generate_UncompressedKeyPublicKey	uncompressed

TP Id	**TP/SEC/ITS-S/S-DATA/NB-11**
Summary	Check that ITS-S generates valid signed data with a certificate containing lifetime field when cf flag is set to lifetime_is_duration.
Reference	IEEE P1609.2/D12 [1], clause 6.3.2
Config Id	CF04
PICS Selection	PIC_Generate_SignPayload AND PIC_Generate_StartValidity AND PIC_Generate_LifetimeIsDuration

Initial conditions

```
with {
    the IUT in Authorized state and
    the IUT is configured to put certificate in each of the signed message
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send a signed message
    }
    then {
        the IUT sends a valid 1609Dot2Data set to MSG_SIGNED_IUT
            containing signed_data.signer
                containing type
                    set to 'certificate'
                containing certificate.unsigned_certificate
                    containing cf
                        indicating 'lifetime_is_duration'
                    containing lifetime
    }
}
```

TP Id	**TP/SEC/ITS-S/S-DATA/NB-12**
Summary	Check that ITS-S generates valid signed data with a certificate containing start_validity field.
Reference	IEEE P1609.2/D12 [1], clause 6.3.2
Config Id	CF04
PICS Selection	PIC_Generate_SignPayload AND PIC_Generate_StartValidity AND NOT PIC_Generate_LifetimeIsDuration

Initial conditions

```
with {
    the IUT in Authorized state and
    the IUT is configured to put certificate in each of the signed message
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send a signed message
    }
    then {
        the IUT sends a valid 1609Dot2Data set to MSG_SIGNED_IUT
            containing signed_data.signer
                containing type
                    set to 'certificate'
                containing certificate.unsigned_certificate
                    containing cf
                        indicating 'use_start_validity'
                    containing start_validity
    }
}
```

TP Id	**TP/SEC/ITS-S/S-DATA/NB-13**
Summary	Check that ITS-S generates valid signed data with a certificate containing encryption_key field.
Reference	IEEE P1609.2/D12 [1], clause 6.3.2
Config Id	CF04
PICS Selection	PIC_Generate_SignPayload AND PIC_Generate_EncryptionKey

Initial conditions

```
with {
    the IUT in Authorized state and
    the IUT is configured to put certificate in each of the signed message
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send a signed message
    }
    then {
        the IUT sends a valid 1609Dot2Data set to MSG_SIGNED_IUT
            containing signed_data.signer
                containing type
                    set to 'certificate'
                containing certificate.unsigned_certificate
                    containing cf
                        indicating 'encryption_key'
                    containing encryption_key
    }
}
```

TP Id	**TP/SEC/ITS-S/S-DATA/NB-14**
Summary	Check that ITS-S generates valid signed data with a certificate containing more than 8 entries in the permissions_list field.
Reference	IEEE P1609.2/D12 [1], clause 6.3.9
Config Id	CF04
PICS Selection	PIC_Generate_SignPayload AND PIC_Generate_PsidArrayWithMoreThan8Entries

Initial conditions

```
with {
    the IUT in Authorized state and
    the IUT is configured to put certificate in each of the signed message
    the CERT_AUTH_TS.scope.permissions.permissions_list contains 9 PSID items
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT is requested to send a signed message
    }
    then {
        the IUT sends a valid 1609Dot2Data set to MSG_SIGNED_IUT
            containing signed_data.signer
                containing type
                    set to 'certificate'
                containing certificate.unsigned_certificate.scope.permissions.permissions_list
                    containing 9 entries
    }
}
```

6.2.1.4 Receiving Data

6.2.1.4.1 Normal Behavior

6.2.1.4.1.1 Signature verification

TP Id	**TP/SEC/ITS-S/R-DATA/NB-01-X**
Summary	Check that ITS-S accepts valid signed data from another ITS-S when the Signer Identifier is a Certificate Digest and the signature contains public key with various types.
Reference	ETSI TS 102 867 [3], clause 5.1.4
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data
                containing signer.digest
                    set to certificate_digest_with_ecdsa_p256 of CERT_AUTH_TS
                containing a valid signature
                    containing ecdsa_signature.R.type
                        set to X_PKT_SIGNATURE
    }
    then {
        the IUT accepts the message
    }
}
```

Variants

X	**PIC Selection**	**X_PKT_SIGNATURE**
A	PIC_Verify_CompressedKeyPublicKey	compressed_lsb_y_0 or compressed_lsb_y_1
B	PIC_Verify_ XCoordinateOnlyPublicKey	x_coordinate_only
C	PIC_Verify_ UncompressedPublicKey	uncompressed

TP Id	**TP/SEC/ITS-S/R-DATA/NB-02-X**
Summary	Check that ITS-S accepts valid signed data from another ITS-S when the Signer Identifier is a Certificate Chain and the signature contains public key with various types.
Reference	ETSI TS 102 867 [3], clause 5.1.4
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data
                containing signer
                    containing type set to 'certificate_chain'
                    containing certificates
                containing a valid signature
                    containing ecdsa_signature.R.type
                        set to X_PKT_SIGNATURE
    }
    then {
        the IUT accepts the message
    }
}
```

Variants

X	**PIC Selection**	**X_PKT_SIGNATURE**
A	PIC_Verify_CompressedKeyPublicKey	compressed_lsb_y_0 or compressed_lsb_y_1
B	PIC_Verify_XCoordinateOnlyPublicKey	x_coordinate_only
C	PIC_Verify_UncompressedKeyPublicKey	uncompressed

6.2.1.4.1.2 Signer verification

TP Id	**TP/SEC/ITS-S/R-DATA/NB-03**
Summary	Check that ITS-S accepts valid signed data from another ITS-S when the Signer Identifier is a Certificate with a lifetime set to duration.
Reference	IEEE P1609.2/D12 [1], clause 6.3.2
Configuration	CF04
PICS Selection	PIC_Verify_StartValidity AND PIC_Verify_LifetimeIsDuration

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type set to 'certificate'
                containing certificate.unsigned_certificate
                    containing cf
                        indicating 'use_start_validity'
                        indicating 'lifetime_is_duration'
    }
    then {
        the IUT accepts the message
    }
}
```

TP Id	**TP/SEC/ITS-S/R-DATA/NB-04**
Summary	Check that ITS-S accepts valid signed data from another ITS-S when the Signer Identifier is a Certificate without a lifetime set to duration.
Reference	IEEE P1609.2/D12 [1], clause 6.3.2
Configuration	CF04
PICS Selection	PIC_Verify_StartValidity AND PIC_Verify_StartValidityIsATimestamp

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type set to 'certificate'
                containing certificate.unsigned_certificate
                    containing cf
                        indicating 'use_start_validity'
                        not indicating 'lifetime_is_duration'
    }
    then {
        the IUT accepts the message
    }
}
```

TP Id	**TP/SEC/ITS-S/R-DATA/NB-05-X**
Summary	Check that ITS-S accepts valid signed data from another ITS-S when the Signer Identifier is a Certificate containing *list_size* PSIDs.
Reference	IEEE P1609.2/D12 [1], clause 6.3.23
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type set to 'certificate'
                containing certificate.unsigned_certificate
                    containing a subject_type
                        set to 'sec_data_exch_ca'
                    containing scope.permissions.permissions_list
                        containing X_LIST_SIZE PSID items
    }
    then {
        the IUT accepts the message
    }
}
```

Variants

X	**X_LIST_SIZE**	**PIC Selection**
A	0	
B	1	
C	4	
D	8	
E	9	PIC_Verify_PsidArrayWithMoreThan8Entries

TP Id	**TP/SEC/ITS-S/R-DATA/NB-06**
Summary	Check that ITS-S accepts valid signed data from another ITS-S when signed with a certificate **containing a**n *IdentifiedNotLocalizedScope* and a zero-length *subject_name* field.
Reference	IEEE P1609.2/D12 [1], clause 6.3.22
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type set to 'certificate'
                containing certificate.unsigned_certificate
                    containing a subject_type
                        set to 'sec_data_exch_identified_not_localized'
                    containing id_not_loc_scope.subject_name
                        set to an empty string
    }
    then {
        the IUT accepts the message
    }
}
```

TP Id	**TP/SEC/ITS-S/R-DATA/NB-07**
Summary	Check that ITS-S accepts valid signed data from another ITS-S when signed with a certificate **containing a**n *IdentifiedNotLocalizedScope* and a non-zero-length *subject_name* field.
Reference	IEEE P1609.2/D12 [1], clause 6.3.22
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type set to 'certificate'
                containing certificate.unsigned_certificate
                    containing subject_type
                        indicating 'sec_data_exch_identified_not_localized'
                    containing id_not_loc_scope.subject_name
                        set to non empty string
    }
    then {
        the IUT accepts the message
    }
}
```

6.2.1.4.2 Exceptional behavior

6.2.1.4.2.1 Generic message verification

TP Id	**TP/SEC/ITS-S/R-DATA/EB-01-X**
Summary	Check that ITS-S discards a 1609.2 secured message if the protocol version is invalid.
Reference	IEEE P1609.2/D12 [1], clause 6.2.1
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing protocol_version
                set to X_INVALID_VERSION_NUMBER
    }
    then {
        the IUT discards the message
    }
}
```

Variants

#	**X_INVALID_VERSION_NUMBER**
A	0
B	1
C	3
D	255

TP Id	**TP/SEC/ITS-S/R-DATA/EB-02-X**
Summary	Check that ITS-S discards a 1609.2 secured message if the content type is not supported.
Reference	IEEE P1609.2/D12 [1], clause 6.2.1
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing type
                set to X_INVALID_CONTENT_TYPE
    }
    then {
        the IUT discards the message
    }
}
```

Variants

X	X_INVALID_CONTENT_TYPE
A	unsecured (0)
B	encrypted(2)
C	certificate_request(3)
D	certificate_response(4)
E	anonymous_certificate_response(5)
F	certificate_request_error(6)
G	crl_request(7)
H	crl(8)
I	signed_wsa(11)
J	certificate_response_acknowledgment (12)
K	ANY_VALUE(128)

6.2.1.4.2.2 Data fields verification

TP Id	**TP/SEC/ITS-S/R-DATA/EB-03**
Summary	Check that ITS-S discards valid signed data from another ITS-S when the expiry time of the received data is before the current time.
Reference	ETSI TS 102 867 [3], clause 5.1.11
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.unsigned_data
                containing tf
                    indicating 'expires'
                containing expiry_time
                    set to value < CLT
    }
    then {
        the IUT discards the message
    }
}
```

TP Id	TP/SEC/ITS-S/R-DATA/EB-04
Summary	Check that ITS-S discards valid signed data which expires before generation time.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.2.1
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data
                containing generation_time
                    set to V_GEN_TIME
                containing expiry_time
                    set to V_GEN_TIME - 1min
    }
    then {
        the IUT discards the message
    }
}
```

TP Id	TP/SEC/ITS-S/R-DATA/EB-05
Summary	Check that ITS-S discards valid signed data generated early then the validity period of the signing certificate.
Reference	IEEE P1609.2/D12 [1], 5.5.3.2.1
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data
                containing generation_time
                    set to V_GEN_TIME
                containing signer
                    containing type
                        set to 'certificate_chain'
                    containing certificates[last].unsigned_certificate
                        containing a start_validity
                            set to V_GEN_TIME + 1min  (V_START_VALIDITY_TIME)
                        containing an expiration
                            set to V_START_VALIDITY_TIME + 1Y
    }
    then {
        the IUT discards the message
    }
}
```

TP Id	**TP/SEC/ITS-S/R-DATA/EB-06**
Summary	Check that ITS-S discards valid signed data generated later then the validity period of the signing certificate.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.2.1
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data
                containing generation_time
                    set to V_GEN_TIME
                containing signer
                    containing type
                        set to 'certificate_chain'
                    containing certificates[last].unsigned_certificate
                        containing an expiration
                            set to V_GEN_TIME – 1min
    }
    then {
        the IUT discards the message
    }
}
```

TP Id	**TP/SEC/ITS-S/R-DATA/EB-07**
Summary	Check that ITS-S discards valid signed data which expires early then the validity period of the signing certificate.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.2.1
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data
                containing expiry_time
                    set to V_EXP_TIME
                containing signer
                    containing type
                        set to 'certificate_chain'
                    containing certificates[last].unsigned_certificate
                        containing a start_validity
                            set to V_EXP_TIME + 1min  (V_START_VALIDITY_TIME)
                        containing an expiration
                            set to V_START_VALIDITY_TIME + 1Y
    }
    then {
        the IUT discards the message
    }
}
```

TP Id	**TP/SEC/ITS-S/R-DATA/EB-08**
Summary	Check that ITS-S discards valid signed data which expires later then the validity period of the signing certificate.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.2.1
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data
                containing expiry_time
                    set to V_EXP_TIME
                containing signer
                    containing type
                        set to 'certificate_chain'
                    containing certificates[last].unsigned_certificate
                        containing an expiration
                            set to V_EXP_TIME – 1min
    }
    then {
        the IUT discards the message
    }
}
```

TP Id	**TP/SEC/ITS-S/R-DATA/EB-09**
Summary	Check that ITS-S discards valid signed data from another ITS-S when the generation location of the received data is beyond the range considered valid by the IUT.
Reference	ETSI TS 102 867 [3], clause 5.1.11
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.unsigned_data
                containing tf
                    indicating 'use_location'
                containing generation_location
                    containing latitude
                        set to PARIS_LAT
                    containing longitude
                        set to PARIS_LON
    }
    then {
        the IUT discards the message
    }
}
```

<table>
<tr><td><b>TP Id</b></td><td><b>TP/SEC/ITS-S/R-DATA/EB-10</b></td></tr>
<tr><td><b>Summary</b></td><td>Check that ITS-S discards valid signed data when the generated location is outside the validity region of the signer's certificate.</td></tr>
<tr><td><b>Reference</b></td><td>IEEE P1609.2/D12 [1], clause 5.5.3.2.1</td></tr>
<tr><td><b>Configuration</b></td><td>CF04</td></tr>
<tr><td><b>PICS Selection</b></td><td></td></tr>
<tr><td colspan="2"><b>Initial conditions</b></td></tr>
<tr><td colspan="2">with {
    IUT in the operational state
}</td></tr>
<tr><td colspan="2"><b>Expected behaviour</b></td></tr>
<tr><td colspan="2">ensure that {
    when {
        the IUT receives a 1609Dot2Data set to <b>MSG_SIGNED_TS</b>
            containing signed_data
                containing signer
                    containing type set to 'certificate'
                    containing certificate.unsigned_certificate.scope.region
                    set to <b>REGION_SMALL</b>
                containing unsigned_data.generation_location
                    containing latitude
                        set to <b>PARIS_LAT</b>
                    containing longitude
                        set to <b>PARIS_LON</b>
    }
    then {
        the IUT discards the message
    }
}</td></tr>
</table>

6.2.1.4.2.3 Signature verification

<table>
<tr><td><b>TP Id</b></td><td><b>TP/SEC/ITS-S/R-DATA/EB-11</b></td></tr>
<tr><td><b>Summary</b></td><td>Check that ITS-S discards data with a cryptographically invalid signature.</td></tr>
<tr><td><b>Reference</b></td><td>IEEE P1609.2/D12 [1], clause 6.2.3</td></tr>
<tr><td><b>Configuration</b></td><td>CF04</td></tr>
<tr><td><b>PICS Selection</b></td><td></td></tr>
<tr><td colspan="2"><b>Initial conditions</b></td></tr>
<tr><td colspan="2">with {
    IUT in the operational state
}</td></tr>
<tr><td colspan="2"><b>Expected behaviour</b></td></tr>
<tr><td colspan="2">ensure that {
    when {
        the IUT receives a 1609Dot2Data set to <b>MSG_SIGNED_TS</b>
            containing signed_data
                containing signature.ecdsa_signature
                    set to the invalid signature value
    }
    then {
        the IUT discards the message
    }
}</td></tr>
</table>

6.2.1.4.2.4 Signer verification

TP Id	**TP/SEC/ITS-S/R-DATA/EB-12-X**
Summary	Check that ITS-S discards a signed 1609.2 message if the signer type is not set to a permitted value.
Reference	IEEE P1609.2/D12 [1], clause 6.2.1
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type
                    set to X_INVALID_SIGNER_TYPE
    }
    then {
        the IUT discards the message
    }
}
```

Variants

X	**X_INVALID_VERSION_NUMBER**	**Comments**
A	'self' (0)	Self-signed certificates are not allowed
B	6	Invalid value
C	255	Invalid value

TP Id	**TP/SEC/ITS-S/R-DATA/EB-13**
Summary	Check that ITS-S discards received data signed with a revoked certificate.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.2.1
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type
                    set to 'certificate'
                containing certificate
                    set to revoked Certificate
    }
    then {
        the IUT discards the message
    }
}
```

TP Id	**TP/SEC/ITS-S/R-DATA/EB-14-X**
Summary	Check that ITS-S discards valid signed data when the signer is a certificate chain in which the region of validity of a subordinate certificate overlaps but is not wholly contained by the region of validity of its issuing certificate.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.2.3
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type set to 'certificate_chain'
                containing certificates[n].scope.region
                    set to REGION_SMALL
                containing certificates[n+1].scope.region
                    set to X_REGION
    }
    then {
        the IUT discards the message
    }
}
```

Variants

X	**X_REGION**
A	**REGION_INTERSECTING**
B	**REGION_OUTSIDE**
C	**REGION_MEDIUM**

TP Id	**TP/SEC/ITS-S/R-DATA/EB-15-X**
Summary	Check that ITS-S discards valid signed data when the signer is a certificate chain in which the validity period of a subordinate certificate is outside that of its issuing certificate.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.2.3
Configuration	CF04
PICS Selection	PIC_Verify_StartValidity AND PIC_Verify_StartValidityIsATimestamp

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type set to 'certificate_chain'
                containing certificates[last-1].unsigned_certificate
                    containing cf
                        set to 'use_start_validity'
                    containing an expiration
                        set to X_TIME_EXP1
                    containing start_validity
                        set to X_TIME_START1
                containing certificates[last].unsigned_certificate
                    containing cf
                        set to 'use_start_validity'
                    containing an expiration
                        set to X_TIME_EXP2
                    containing start_validity
                        set to X_TIME_START2
    }
    then {
        the IUT discards the message
    }
}
```

Variants

X	X_TIME_START1	X_TIME_EXP1	X_TIME_START2	X_TIME_EXP2	Comment
A	CLT+2Y	CLT+3Y	CLT-1Y	CLT+1Y	Subordinate certificate validity period is totaly before the issuing one
B	CLT-1Y	CLT+2Y	CLT-2Y	CLT+1Y	Subordinate certificate validity period is intersecting the issuing one
C	CLT-2Y	CLT+1Y	CLT-1Y	CLT+2Y	Subordinate certificate validity period is intersecting the issuing one
D	CLT-1Y	CLT+1Y	CLT+2Y	CLT+3Y	Subordinate certificate validity period is totaly after the issuing one

TP Id	TP/SEC/ITS-S/R-DATA/EB-16
Summary	Check that ITS-S discards valid signed data when the signer is a certificate chain in which the operational permissions of a subordinate certificate are not a subset of the permissions of its issuing certificate.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.3
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type set to 'certificate_chain'
                containing certificates[last-1].unsigned_certificate
                    containing scope.permissions.permissions_list
                        set to array[1]
                            containing PSID_A
                containing certificates[last].unsigned_certificate
                    containing scope.permissions.permissions_list
                        set to array[1]
                            containing PSID_B
    }
    then {
        the IUT discards the message
    }
}
```

TP Id	TP/SEC/ITS-S/R-DATA/EB-17
Summary	Check that ITS-S discards valid signed data when the signer is a certificate chain in which the subordinate certificate has a valid signature which is not the signature of its issuing certificate.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.3
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type set to 'certificate_chain'
                containing certificates[last]
                    containing valid signature
                        verifiable using verification key of the certificate pointed by signer_id
                    containing signer_id
                        set to the value not equal to the 8-byte hash of the certificates[last-1]
    }
    then {
        the IUT discards the message
    }
}
```

TP Id	**TP/SEC/ITS-S/R-DATA/EB-18**
Summary	Check that ITS-S discards valid signed data when the signer is a certificate chain in which an issuing certificate is not permitted to issue certificates of its subordinate certificate's type.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.3
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type set to 'certificate_chain'
                containing certificates[last-1].unsigned_certificate
                    containing a scope
                        containing permitted_subject_types
                            set to 'sec_data_exch_identified_localized'
                containing certificates[last].unsigned_certificate
                    containing a subject_type
                        set to 'sec_data_exch_anonymous'
    }
    then {
        the IUT discards the message
    }
}
```

TP Id	**TP/SEC/ITS-S/R-DATA/EB-19-X**
Summary	Check that ITS-S discards a signed 1609.2 message if the version_and_type field is not set to the value 2.
Reference	IEEE P1609.2/D12 [1], clause 6.3.2
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type set to 'certificate_chain'
                containing certificates[last].version_and_type
                    set to INVALID_CERT_VERSION_AND_TYPE
    }
    then {
        the IUT discards the message
    }
}
```

Variants

Y	**INVALID_CERT_VERSION_AND_TYPE**
A	0
B	1
C	3
D	255

<table>
<tr><th>TP Id</th><th>TP/SEC/ITS-S/R-DATA/EB-20</th></tr>
<tr><td><b>Summary</b></td><td>Check that ITS-S discards a signed 1609.2 message if the signature is calculated over the hash of the <i>version_and_type</i> and the <i>unsigned_certificate</i> fields if the calculation does not use the compressed representation of all public keys and reconstruction values contained in the certificate.</td></tr>
<tr><td><b>Reference</b></td><td>IEEE P1609.2/D12 [1], clause 6.3.1</td></tr>
<tr><td><b>Configuration</b></td><td>CF04</td></tr>
<tr><td><b>PICS Selection</b></td><td>PIC_Verify_UncompressedKey</td></tr>
<tr><td colspan="2"><b>Initial conditions</b></td></tr>
<tr><td colspan="2">with {
    IUT in the operational state
}</td></tr>
<tr><td colspan="2"><b>Expected behaviour</b></td></tr>
<tr><td colspan="2">ensure that {
    when {
        the IUT receives a 1609Dot2Data set to <b>MSG_SIGNED_TS</b>
            containing signed_data.signer
                containing type
                    set to 'certificate_chain'
                containing certificates[last].unsigned_certificate
                    containing verification_key.public_key.type (<b>V_PKT_VK</b>)
                        set to 'uncompressed'
                    containing signature.ecdsa_signature
                        calculated using uncompressed representation of <b>V_PKT_VK</b>
    }
    then {
        the IUT discards the message
    }
}</td></tr>
</table>

<table>
<tr><th>TP Id</th><th>TP/SEC/ITS-S/R-DATA/EB-21</th></tr>
<tr><td><b>Summary</b></td><td>Check that ITS-S discards a signed 1609.2 message if both the <i>crl_series</i> and the <i>expiration</i> fields in the <i>unsigned_certificate</i> are empty.</td></tr>
<tr><td><b>Reference</b></td><td>IEEE P1609.2/D12 [1], clause 6.3.1</td></tr>
<tr><td><b>Configuration</b></td><td>CF04</td></tr>
<tr><td><b>PICS Selection</b></td><td></td></tr>
<tr><td colspan="2"><b>Initial conditions</b></td></tr>
<tr><td colspan="2">with {
    IUT in the operational state
}</td></tr>
<tr><td colspan="2"><b>Expected behaviour</b></td></tr>
<tr><td colspan="2">ensure that {
    when {
        the IUT receives a 1609Dot2Data set to <b>MSG_SIGNED_TS</b>
            containing signed_data.signer
                containing type
                    set to 'certificate_chain'
                containing certificate[last].unsigned_certificate
                    containing crl_series
                        set to 0
                    containing expiration
                        set to 0
    }
    then {
        the IUT discards the message
    }
}</td></tr>
</table>

TP Id	**TP/SEC/ITS-S/R-DATA/EB-22**
Summary	Check that ITS-S discards a signed 1609.2 message if the permissions requested in the end-user certificate contains duplicate PSIDs.
Reference	IEEE P1609.2/D12 [1], clause 6.3.9
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type
                    set to 'certificate_chain'
                containing certificates[last].unsigned_certificate.scope.permissions.permissions_list
                    set to array[2]
                        containing PSID_A
                        containing PSID_A
    }
    then {
        the IUT discards the message
    }
}
```

TP Id	**TP/SEC/ITS-S/R-DATA/EB-23-X**
Summary	Check that ITS-S discards a signed 1609.2 message if the *latitude* specified in the *region* associated with the signers certificate scope is outside the limits of ±90˚.
Reference	IEEE P1609.2/D12 [1], clause 6.3.9
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type
                    set to 'certificate_chain'
                containing certificates[last].unsigned_certificate.scope.region
                    containing latitude
                        set to X_INVALID_LATITUDE
    }
    then {
        the IUT discards the message
    }
}
```

Variants

X	**X_INVALID_LATITUDE**
A	900000001
B	-900000001

TP Id	**TP/SEC/ITS-S/R-DATA/EB-24-X**
Summary	Check that ITS-S discards a signed 1609.2 message if the *longitude* specified in the *region* associated with the signers certificate scope is outside the limits of ±180˚.
Reference	IEEE P1609.2/D12 [1], clause 6.3.9
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type
                    set to 'certificate_chain'
                containing certificates[last].unsigned_certificate.scope.region
                    containing longitude
                        set to X_INVALIT_LONGITUDE
    }
    then {
        the IUT discards the message
    }
}
```

Variants

X	**X_INVALID_LONGITUDE**
A	1800000001
B	-1800000001

TP Id	**TP/SEC/ITS-S/R-DATA/EB-25**
Summary	Check that ITS-S discards a signed 1609.2 message if it contains a secured data exchange, identified not localized scope with zero PSID SSPs in its permissions list.
Reference	IEEE P1609.2/D12 [1], clause 6.3.23
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type
                    set to 'certificate_chain'
                containing certificates[last].unsigned_certificate.scope.permissions.permissions_list
                    set to array[0]
                        not containing any PSID SSP
    }
    then {
        the IUT discards the message
    }
}
```

TP Id	**TP/SEC/ITS-S/R-DATA/EB-26**
Summary	Check that ITS-S discards a signed 1609.2 message if it contains a secured data exchange, identified not localized scope with a PSID SSPs of more than 31 octets.
Reference	IEEE P1609.2/D12 [1], clause 6.3.23
Configuration	CF04
PICS Selection	

Initial conditions

```
with {
    IUT in the operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data set to MSG_SIGNED_TS
            containing signed_data.signer
                containing type
                    set to 'certificate_chain'
                containing certificates[last].unsigned_certificate
                    containing scope.permissions.permissions_list
                        set to array[1]
                            containing V_PSIDSSPP_A
                                containing service_specific_permission
                                    longer than 31 octets
    }
    then {
        the IUT discards the message
    }
}
```

6.2.2 Certificate Authority

6.2.2.1 Normal Behavior

6.2.2.1.1 Generic message verification

TP Id	**TP/SEC/CA/NB-01**
Summary	Check that CA correctly decrypts a Certificate Request.
Reference	IEEE P1609.2/D12 [1], clause 5.6.2.1
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest
    }
    then {
        the IUT decrypts the request
    }
}
```

TP Id	**TP/SEC/CA/NB-02-X**
Summary	Check that CA generates certificate response encoded using the key stored in response_encryption_key field in the request.
Reference	IEEE P1609.2/D12 [1], clause 6.3.34
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
            containing unsigned_csr.response_encryption_key (V_RESPONSE_ENC_KEY)
    }
    then {
        the IUT sends a CertificateResponse set to X_RESPONSE
            encrypted using V_RESPONSE_ENC_KEY
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRRSP_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHRSP_IUT**

TP Id	**TP/SEC/CA/NB-03-X**
Summary	Check that CA generates certificate response.
Reference	IEEE P1609.2/D12 [1], clause 6.2.17
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateRequest set to X_REQUEST
    }
    then {
        the IUT sends a CertificateResponse set to X_RESPONSE
            containing certificate_chain[last].signature
                verifiable using CERT_CA.unsigned_certificate.verification_key
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRRSP_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHRSP_IUT**

TP Id	**TP/SEC/CA/NB-04-X**
Summary	Check that the CA accepts a valid certificate request having correct fields and values, signed by a signer_id with type set to 'certificate'.
Reference	IEEE P1609.2/D12 [1], clause 6.2.4
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateRequest set to X_REQUEST
            containing signer
                containing type
                    set to 'certificate'
                containing certificate
    }
    then {
        the IUT sends a CertificateResponse set to X_RESPONSE
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRRSP_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHRSP_IUT**

TP Id	**TP/SEC/CA/NB-05-X**
Summary	Check that the CA accepts a valid certificate request having correct fields and values, signed by a signer_id with type set to 'certificate_chain'.
Reference	IEEE P1609.2/D12 [1], clause 6.2.4
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
            containing signer
                containing type
                    set to 'certificate_chain'
                containing certificates
                    set to array of certificates
    }
    then {
        the IUT sends a CertificateResponse set to X_RESPONSE
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRRSP_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHRSP_IUT**

6.2.2.1.2 Key Compression

TP Id	TP/SEC/CA/NB-06-X-Y
Summary	Check that an CA accepts a certificate request, signed by a valid certificate chain and containing various public key types.
Reference	IEEE P1609.2/D12 [1], clause 6.2.17
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {

    the IUT in operational state
    the IUT is configured to use signature of type Y_PKT_RES_SIGN
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateRequest set to X_REQUEST
            containing signer.type
                set to 'certificate_chain'
            containing signer.certificate_chain[last]
                containing signature.ecdsa_signature.R.type
                    set to Y_PKT_SIG_SIGN
                containing unsigned_certificate.verification_key.public_key.type
                    set to Y_PKT_SIG_VK
            containing unsigned_csr.verification_key.public_key.type
                set to Y_PKT_VK
            containing unsigned_csr.response_encryption_key.public_key.type
                set to Y_PKT_REK
            containing signature.ecdsa_signature
                calculated using compressed representation of Y_PKT_VK and Y_PKT_REK
                containing R.type
                    set to Y_PKT_REQ_SIGN
    }
    then {
        the IUT sends a valid CertificateResponse set to X_RESPONSE
            containing certificates[last]
                containing unsigned_certificate.verification_key.public_key.type
                    set to Y_PKT_VK
                containing signature.ecdsa_signature
                    calculated using compressed representation of Y_PK_TYPE_VK and Y_PK_TYPE_REK
                    containing R.type
                        set to Y_PKT_RES_SIGN
    }
}
```

Variants

X	X_REQUEST	X_RESPONSE
A	**MSG_ENRREQ_TS**	**MSG_ENRRSP_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHRSP_IUT**

Possible values:
Comp : compressed_lsb_y_0 or compressed_lsb_y_1
X_co : x_coordinate_only
Uncomp: uncompressed

Y	Y_PKT_SIG_VK	Y_PKT_SIG_SIGN	Y_PKT_REQ_SIGN	Y_PKT_VK	Y_PKT_REK	Y_PKT_RES_SIGN
A	Comp	X_co	X_co	Comp	Comp	Comp
B	X_co	X_co	X_co	X_co	X_co	X_co
C	Uncomp	Uncomp	Uncomp	Uncomp	Uncomp	Uncomp
D	Comp	U Uncomp	Uncomp	Comp	X_co	Uncomp
E	X_co	Uncomp	Uncomp	X_co	X_co	X_co
F	Uncomp	Comp	Comp	Uncomp	Uncomp	Comp
G	Y	Comp	Comp	X_co	Comp	Uncomp
H	X_co	Comp	Comp	X_co	X_co	X_co

6.2.2.1.3 Permissions

TP Id	**TP/SEC/CA/NB-07-X-Y**
Summary	Check that an CA responds to a certificate request with the list of permissions fully contained in the request signer certificate.
Reference	IEEE P1609.2/D12 [1], clauses 6.3.9 and 6.3.23,
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
    the IUT is configured to provide certificates with permissions {PSID_A, PSID_B, PSID_C, PSID_D, PSID_E,
    PSID_F, PSID_G, PSID_H, PSID_I}
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateRequest set to X_REQUEST
            containing signer.certificate.unsigned_certificate.sec_data_exch_ca_scope.permissions.permissions_list
                set to Y_PSID_LIST_SIGNER
            containing unsigned_csr.type_specific_data.V_REQ_SCOPE
                containing permissions.permissions_list
                    set to Y_PSIDSSP_LIST_REQUEST
    }
    then {
        the IUT sends a valid CertificateResponse set to X_RESPONSE
            containing certificates[last].unsigned_certificate
                containing V_REQ_SCOPE
                    containing permissions.permissions_list
                        set to Y_PSIDSSP_LIST_RES
    }
}
```

Variants

X	X_REQUEST	X_RESPONSE
A	**MSG_ENRREQ_TS**	**MSG_ENRRSP_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHRSP_IUT**

Variants

Y	PICS Selection	Y_PSID_LIST_SIGNER	Y_PSIDSSP_LIST_REQUEST	Y_PSIDSSP_LIST_RES
A		{**PSID_A**}	{**PSID_A**}	{**PSID_A**}
B		{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**}	{**PSID_A**}	{**PSID_A**}
C		{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**}	{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**}	{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**}
D		{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**, **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H**}	{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**, **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H**}	{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**, **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H**}
E		{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**}	{ **PSID_C**, **PSID_D**, **PSID_E**, **PSID_F** }	{**PSID_A**, **PSID_B**}
F	PIC_Verify_PsidArrayWithMoreThan8Entries	{**PSID_A**}	{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**, **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H**, **PSID_I**}	{**PSID_A**}
G	PIC_Verify_PsidArrayWithMoreThan8Entries	{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**, **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H**, **PSID_I**}	{**PSID_A**}	{**PSID_A**}

TP Id	**TP/SEC/CA/NB-08-X-Y**
Summary	Check that an CA responds to a certificate request with the list of permissions set to the intersection between requested permissions and CA certificate permissions.
Reference	**I**EEE P1609.2/D12 [1], clauses 6.3.9 and 6.3.23
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
    the IUT is configured with an CA certificate
        containing certificate.unsigned_certificate.sec_data_exch_ca_scope.permissions.permissions_list
            set to Y_PSID_LIST_CA_CERT
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateRequest set to X_REQUEST
            containing unsigned_csr.type_specific_data.REQ_SCOPE
                containing permissions.permissions_list
                    set to Y_PSIDSSP_LIST_REQUEST
    }
    then {
        the IUT sends a valid CertificateResponse set to X_RESPONSE
            containing certificates[last].unsigned_certificate
                containing REQ_SCOPE
                    containing permissions.permissions_list
                        set to Y_PSIDSSP_LIST_RES
    }
}
```

Note: Request signing certificate fully covers **Y_PSIDSSP_LIST_REQUEST**

Variants

X	X_REQUEST	X_RESPONSE
A	**MSG_ENRREQ_TS**	**MSG_ENRRSP_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHRSP_IUT**

Variants

Y	PICS Selection	Y_PSID_LIST_CA_CERT	Y_PSIDSSP_LIST_REQUEST	Y_PSIDSSP_LIST_RES
A		{**PSID_A**}	{**PSID_A**}	{**PSID_A**}
B		{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**}	{**PSID_A**}	{**PSID_A**}
C		{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**}	{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**}	{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**}
D		{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**, **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H**}	{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**, **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H**}	{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**, **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H**}
E		{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**}	{ **PSID_C**, **PSID_D**, **PSID_E**, **PSID_F** }	{**PSID_A**, **PSID_B**}
F	PIC_Verify_PsidArrayWithMoreThan8Entries	{**PSID_A**}	{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D** , **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H**, **PSID_I**}	{**PSID_A**}
G	PIC_Verify_PsidArrayWithMoreThan8Entries	{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D** , **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H**, **PSID_I**}	{**PSID_A**}	{**PSID_A**}

6.2.2.1.4 Expiration

TP Id	**TP/SEC/CA/NB-09-X**
Summary	Check that the CA accepts a valid certificate request having specified start_validity time.
Reference	IEEE P1609.2/D12 [1], clause 6.2.17
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
            containing unsigned_csr
                containing cf
                    indicating use_start_validity
                    and not indicating lifetime_is_duration
                containing start_validity
                    set to 1 Jan 2010
    }
    then {
        the IUT sends a CertificateResponse set to X_RESPONSE
            containing certificates[last].unsigned_certificate
                valid from 1 Jan 2010
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRRSP_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHRSP_IUT**

TP Id	**TP/SEC/CA/NB-10-X**
Summary	Check that the CA accepts a valid certificate request with lifetime set to 0.
Reference	IEEE P1609.2/D12 [1], clause 6.3.34 ETSI TS 102 941 [2] Table 1 : Contents of ITS-S EnrolmentRequest message ETSI TS 102 941 [2] Table 2 : Contents of ITS-S AuthorizationRequest message
Config Id	CF01,CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
            containing unsigned_csr
                containing cf
                    indicating use_start_validity and lifetime_is_duration
                containing lifetime
                    set to 0
    }
    then {
        the IUT sends a valid CertificateResponse set to X_RESPONSE
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRRSP_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHRSP_IUT**

TP Id	**TP/SEC/CA/NB-11-X**
Summary	Check that the CA accepts a valid certificate request with start_validity set to 0.
Reference	IEEE P1609.2/D12 [1], clause 6.3.34 ETSI TS 102 941 [2] Table 1 : Contents of ITS-S EnrolmentRequest message ETSI TS 102 941 [2] Table 2 : Contents of ITS-S AuthorizationRequest message
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
            containing unsigned_csr
                containing cf
                    indicating use_start_validity
                    and not indicating lifetime_is_duration
                containing start_validity
                    set to 0
    }
    then {
        the IUT sends a valid CertificateResponse set to X_RESPONSE
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRRSP_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHRSP_IUT**

TP Id	**TP/SEC/CA/NB-12-X**
Summary	Check that CA generates valid certificate response with a certificate containing the field start_validity.
Reference	IEEE P1609.2/D12 [1], clause 6.3.2
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
    the IUT is configured to use start_validity flag
    the IUT is configured not to use a lifetime_is_duration flag
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateRequest set to X_REQUEST
    }
    then {
        the IUT sends a CertificateResponse set to X_RESPONSE
            containing certificate_chain[last].unsigned_certificate
                containing cf
                    indicating use_start_validity
                    and not indicating lifetime_is_duration
                containing start_validity
                    set to the timestamp < certificate_chain[last].unsigned_certificate.expiration
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRRSP_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHRSP_IUT**

6.2.2.1.5 Regions

TP Id	**TP/SEC/CA/NB-13-X-Y**
Summary	Check that an CA responds to a certificate request with the region which is fully containing in the request region and in the signer region.
Reference	IEEE P1609.2/D12 [1], clauses 6.3.13, 6.3.15 and 5.5.3.3
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateRequest set to X_REQUEST
        containing signer.certificate.unsigned_certificate.ANY_SCOPE.region
            set to Y_REGION_SIGNER
        containing unsigned_csr.type_specific_data.ANY_SCOPE.region
            set to Y_REGION_REQUEST
    }
    then {
        the IUT sends a valid CertificateResponse set to X_RESPONSE
        containing certificates[last].unsigned_certificate.ANY_SCOPE.region
            containing region_type
                set to 'circle'
            containing circular_region inside Y_REGION_RES
    }
}
```

Variants

X	X_REQUEST	X_RESPONSE
A	**MSG_ENRREQ_TS**	**MSG_ENRRSP_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHRSP_IUT**

Variants

Y	Y_REGION_SIGNER	Y_REGION_REQUEST	Y_REGION_RES
A	REGION_LARGE	REGION_MEDIUM	REGION_MEDIUM
B	REGION_LARGE	REGION_LARGE	REGION_LARGE
C	REGION_MEDIUM	REGION_SMALL	REGION_SMALL

6.2.2.2 Exceptional Behavior

6.2.2.2.1 Invalid Message Fields

TP Id	**TP/SEC/CA/EB-01-X**
Summary	Check that CA discards certificate requests if the message content type is different than "encrypted".
Reference	IEEE P1609.2/D12 [1], clause 6.2.1
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data structure
            containing type
                set to X_INVALID_CONTENT_TYPE
    }
    then {
        the IUT discards the received message
    }
}
```

Variants

X	**X_INVALID_CONTENT_TYPE**
A	unsecured (0)
B	signed(1)
C	certificate_request(3)
D	certificate_response(4)
E	anonymous_certificate_response(5)
F	certificate_request_error(6)
G	crl_request(7)
H	crl(8)
I	signed_partial_payload(9)
J	signed_external_payload(10)
K	signed_wsa(11)
L	certificate_response_acknowledgment (12)
M	ANY_VALUE(128)

TP Id	**TP/SEC/CA/EB-02-X**
Summary	Check that CA discards certificate requests if the protocol_version is not 2.
Reference	IEEE P1609.2/D12 [1], clause 6.2.1.1
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data structure
            containing protocol_version
                set to X_INVALID_VERSION_NUMBER
    }
    then {
        the IUT discards the received message
    }
}
```

Variants

#	X_INVALID_VERSION_NUMBER
A	0
B	1
C	3
D	255

TP Id	**TP/SEC/CA/EB-03-X**
Summary	Check that CA discards messages others than certificate request.
Reference	IEEE P1609.2/D12 [1], clause 6.2.1.1
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a 1609Dot2Data structure
            containing encrypted_data
                containing encrypted_data (ToBeEncrypted data structure)
                    containing type
                        set to X_INVALID_CONTENT_TYPE
    }
    then {
        the IUT discards the received message
    }
}
```

Variants

X	X_INVALID_CONTENT_TYPE
A	unsecured (0)
B	signed(1)
C	encrypted(2)
D	certificate_response(4)
E	anonymous_certificate_response(5)
F	certificate_request_error(6)
G	crl_request(7)
H	crl(8)
I	signed_partial_payload(9)
J	signed_external_payload(10)
K	signed_wsa(11)
L	certificate_response_acknowledgment (12)
M	ANY_VALUE(128)

TP Id	**TP/SEC/CA/EB-04-X-Y**
Summary	Check that CA discards certificate request if the certificate is not an explicit one.
Reference	IEEE P1609.2/D12 [1], clause 6.3.1
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
            containing unsigned_csr.version_and_type
                set to Y_INVALID_CERT_VERSION_AND_TYPE
    }
    then {
        the IUT sends a CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'verification_failure'
    }
}
```

Variants

X	X_REQUEST	X_RESPONSE
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

Variants

Y	Y_INVALID_CERT_VERSION_AND_TYPE
A	0
B	1
C	3
D	255

TP Id	**TP/SEC/CA/EB-05-X**
Summary	Check that CA generates a certificate request error with valid fields when it receives the request with cryptographically invalid signature.
Reference	IEEE P1609.2/D12 [1], clause 6.2.17
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
            containing a criptogtaphicaly invalid signature
    }
    then {
        the IUT sends a CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'verification_failure'
    }
}
```

Variants

X	X_REQUEST	X_RESPONSE
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

6.2.2.2.2 Invalid Certificate or Certificate Chain

TP Id	**TP/SEC/CA/EB-06-X**
Summary	Check that an CA discards an certificate request with an cryptographically invalid signing certificate.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.3
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
        containing signer
            containing type
                set to 'certificate_chain'
            containing certificates[last]
                containing cryptographically invalid signature
    }
    then {
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'verification_failure'
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

TP Id	**TP/SEC/CA/EB-07-X**
Summary	Check that an CA discards an certificate request containing a signer containing an invalid certificate (unknown root certificate).
Reference	IEEE P1609.2/D12 [1], clauses 5.6.1.2 and 6.3.37
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
        containing signer
            containing type
                set to 'certificate_chain'
            containing certificates[0] (root certificate)
                set to an unknown root certificate
    }
    then {
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'verification_failure'
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

TP Id	**TP/SEC/CA/EB-08-X**
Summary	Check that an CA discards an certificate request containing a signer containing an invalid certificate chain (expired root certificate).
Reference	IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 6.3.37
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
        containing signer
            containing type
                set to 'certificate_chain'
            containing certificates[0] (root certificate)
                containing unsigned_certificate.expiration < CLT
    }
    then {
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'verification_failure'
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

TP Id	**TP/SEC/CA/EB-09-X**
Summary	Check that an CA discards an certificate request containing a signer containing an invalid certificate chain (cryptographically invalid root certificate).
Reference	IEEE P1609.2/D12 [1], clauses 5.6.1.2 and 6.3.37
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
        containing signer
            containing type
                set to 'certificate_chain'
            containing certificates[0] (root certificate)
                containing invalid signature
    }
    then {
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'verification_failure'
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

TP Id	**TP/SEC/CA/EB-10-X**
Summary	Check that an CA discards an certificate request containing a signer containing an incomplete certificate chain (missing root certificate).
Reference	IEEE P1609.2/D12 [1], clauses 5.6.1.2 and 6.3.37
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
        containing signer
            containing type
                set to 'certificate_chain'
            containing certificates
                not containing a root certificate (CERT_ROOT)
    }
    then {
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'verification_failure'
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

TP Id	**TP/SEC/CA/EB-11-X**
Summary	Check that an CA discards an certificate request containing an unknown signer.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.3
Config Id	CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
        containing signer
            containing type set to 'certificate'
            containing certificate
                set to unknown certificate (see note)
    }
    then {
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'csr_cert_revoked'
    }
}
```

NOTE: A certificate that does not belong to a chain that leads to a known trust anchor.

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

TP Id	**TP/SEC/CA/EB-12-X**	
Summary	Check that an CA discards an certificate request containing a revoked signer certificate.	
Reference	IEEE P1609.2/D12 [1], clause 6.3.37	
Config Id	CF01, CF02	
PICS Selection		
Initial conditions		

with {
the IUT in operational state
}		

Expected behaviour		

ensure that {
when {
the IUT receives a CertificateRequest set to **X_REQUEST**
containing signer
containing type
set to 'certificate'
containing certificate
set to revoked certificate
}
then {
the IUT sends a valid CertificateRequestError set to **X_RESPONSE**
containing reason
set to 'csr_cert_revoked'
}
}		

Variants		
X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

6.2.2.2.3 Invalid Certificate Fields

TP Id	**TP/SEC/CA/EB-13-X-Y**
Summary	Check that an CA discards an certificate request with certificate content flags other than 'use_start_validity' or 'lifetime_is_duration'.
Reference	ETSI TS 102 867 [3], clause 5.1.2.2, IEEE P1609.2/D12 [1], clauses 6.3.2 and 6.3.34
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
            containing unsigned_csr.cf
                set to Y_INVALID_FLAGS
    }
    then {
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'request_denied'
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

Variants

	Y_INVALID_FLAGS		
Y	**use_start_validity (0)**	**lifetime_is_duration(1)**	**encryption_key (2)**
A	Yes	Yes	Yes
B	No	Yes	Yes
C	Yes	No	Yes
D	No	No	Yes

TP Id	**TP/SEC/CA/EB-14-X**
Summary	Check that an CA discards an certificate request signed with expired credentials.
Reference	IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 6.3.37
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
        containing signer
            containing type
                set to 'certificate_chain'
            containing certificates[last]
                containing unsigned_certificate.expiration < CLT
    }
    then {
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'verification_failure'
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

TP Id	**TP/SEC/CA/EB-15-X**
Summary	Check that CA generates certificate request error with valid fields and with signature of various public key types. Check that CA calculate request hash using compressed representation of all public keys.
Reference	IEEE P1609.2/D12 [1], clause 6.2.17
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
    the IUT is configured to use signature of type Y_PK_TYPE_SIGNATURE
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
            containing a criptogtaphicaly invalid signature
            containing unsigned_csr.verification_key.public_key.type (V_PK_REQ_VK)
                set to 'uncompressed'
            containing unsigned_csr.response_encryption_key.public_key.type (V_PK_REQ_REK)
                set to 'uncompressed'
    }
    then {
        the IUT sends a CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'verification_failure'
            containing request_hash
                set to the hash calculated using compressed representation of the V_PK_REQ_VK and
                    V_PK_REQ_REK
            containing signature.ecdsa_signature
                containing R.type
                    set to Y_PK_TYPE_SIGNATURE
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

Variants

Y	**Y_PK_TYPE_SIGNATURE**
A	compressed_lsb_y_0/1
B	x_coordinate_only
C	uncompressed

6.2.2.2.4 Invalid Permissions

TP Id	**TP/SEC/CA/EB-16-X-Y**
Summary	Check that an CA discards an certificate request with an invalid PsidArray.type.
Reference	ETSI TS 102 867 [3], clause 5.1.2.2, IEEE P1609.2/D12 [1], clause 6.3.7
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
        containing unsigned_csr.type_specific_data.ANY_SCOPE
            containing permissions.type
                set to Y_INVALID_ARRAY_TYPE
    }
    then {
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'request_denied'
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

Variants

Y	**Y_INVALID_ARRAY_TYPE**
A	from_issuer(0)
B	ANY OTHER (128)

TP Id	**TP/SEC/CA/EB-17-X**
Summary	Check that an CA discards an certificate request signed by the certificate with the permissions list which is not a superset of requested permissions list.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.3
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
        containing signer.certificate.unsigned_certificate.ANY_SCOPE.permissions.permissions_list
            set to X_PSID_LIST_SIGNER
        containing unsigned_csr.type_specific_data.ANY_SCOPE.permissions.permissions_list
                set to X_PSID_LIST_REQ
    }
    then {
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'request_denied'
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

Variants

Y	**PICS Selection**	**X_PSID_LIST_SIGNER**	**X_PSID_LIST_REQ**
A		{**PSID_B**}	{**PSID_A**}
B		{ **PSID_B**, **PSID_C**, **PSID_D** , **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H**, **PSID_I**}	{**PSID_A**}
C	PIC_Verify_PsidArrayWithMoreThan8Entries	{ **PSID_B**, **PSID_C**, **PSID_D** , **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H**, **PSID_I**, **PSID_J**}	{**PSID_A**}
D	PIC_Verify_PsidArrayWithMoreThan8Entries	{**PSID_A**}	{**PSID_B**, **PSID_C**, **PSID_D** , **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H**, **PSID_I**}
E		{**PSID_A**}	{**PSID_B**, **PSID_C**, **PSID_D** , **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H**, **PSID_I**, **PSID_J**}
F		{**PSID_A**, **PSID_B**, **PSID_C**, **PSID_D**}	{ **PSID_E**, **PSID_F**, **PSID_G**, **PSID_H** }

TP Id	**TP/SEC/CA/EB-18-X**
Summary	Check that an CA discards an certificate request if it has duplicated PSIDs.
Reference	IEEE P1609.2/D12 [1], clause 6.3.9
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
    the IUT containing CA_CERT
        containing unsigned_certificate.scope.permissions.permissions_list(V_PERM_LIST)
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
            containing unsigned_csr.type_specific_data.scope
                containing permissions.permissions_list
                    set to array[2]{
                        containing V_PERM_LIST[0]
                        containing V_PERM_LIST[0]
                    }
    }
    then {
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'verification_failure'
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

6.2.2.2.5 Invalid Regions

TP Id	**TP/SEC/CA/EB-19-X-Y**
Summary	Check that an CA discards a certificate request signed by the certificate containing a scope with a circular region (REGION_SIGNER) and an unsigned csr with a circular region (REGION_REQUEST) that is not fully contained in the signer region.
Reference	IEEE P1609.2/D12 [1], clause 5.5.3.3
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
        containing signer
            containing type
                set to 'certificate'
            containing certificate.unsigned_certificate.ANY_SCOPE
                containing region
                    set to Y_REGION_SIGNER
        containing unsigned_csr.type_specific_data.ANY_SCOPE
            containing region
                set to Y_REGION_REQUEST
    }
    then {
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'request_denied'
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

Variants

Y	**Y_REGION_SIGNER**	**Y_REGION_REQUEST**
A	**REGION_SMALL**	**REGION_OUTSIDE**
B	**REGION_SMALL**	**REGION_LARGE**
C	**REGION_SMALL**	**REGION_INTERSECTING**

6.2.2.2.6 Expiration

TP Id	**TP/SEC/CA/EB-20-X**
Summary	Check that an CA discards a certificate request containing an expired signer certificate.
Reference	IEEE P1609.2/D12 [1], clauses 6.3.2, 6.3.37 and 6.2.7, ETSI TS 102 867 [3], clause 5.1.2.1
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
        containing signer
            containing type
                set to 'certificate'
            containing certificate.unsigned_certificate
                containing expiration
                    set to CLT – '1Y'
                containing lifetime
                    set to '1Y'
    }
    then {
        the IUT sends a CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'csr_cert_expired'
    }
}
```

Variants

X	X_REQUEST	X_RESPONSE
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

TP Id	**TP/SEC/CA/EB-21-X-Y**
Summary	Check that an CA discards a certificate request with invalid expiration time.
Reference	IEEE P1609.2/D12 [1], clauses 6.3.2, 6.3.37 and 6.2.7, ETSI TS 102 867 [3], clause 5.1.2.1
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to X_REQUEST
        containing signer.certificate.unsigned_certificate
            containing expiration
                set to Y_EXP_SIGNER
            containing lifetime
                set to Y_LT_SIGNER
        containing unsigned_csr
            containing expiration
                set to Y_EXP_REQUEST
            containing lifetime
                set to Y_LT_REQUEST
    }
    then{
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'request_denied'
    }
}
```

Variants

X	**X_REQUEST**	**X_RESPONSE**
A	**MSG_ENRREQ_TS**	**MSG_ENRERR_IUT**
B	**MSG_AUTHREQ_TS**	**MSG_AUTHERR_IUT**

Variants

Y	**Y_EXP_SIGNER**	**Y_LT_SIGNER**	**Y_EXP_REQUEST**	**Y_EXP_REQUEST**
A	CLT+1Y	1Y	CLT+2Y	1M
B	CLT+1Y	1Y	CLT+2Y	1Y
C	CLT+1Y	1Y	CLT+2Y	2Y
D	CLT+2Y	1M	CLT+1Y	1M
E	CLT+2Y	1Y	CLT+1Y	1M
F	CLT+3Y	2Y	CLT+2Y	2Y

6.2.3 Enrolment Authority

6.2.3.1 Normal Behavior

TP Id	**TP/SEC/EA/ENR/NB-01**
Summary	Check that the EA accepts a valid self-signed enrolment request having correct fields and values.
Reference	IEEE P1609.2/D12 [1], clause 6.2.4
Config Id	CF01
PICS Selection	PIC_Generate_SelfSigned

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateRequest set to MSG_ENRREQ_TS
            containing signer
                containing type
                    set to 'self'
    }
    then {
        the IUT sends a CertificateResponse set to MSG_ENRRSP_IUT
    }
}
```

6.2.3.2 Exceptional Behavior

TP Id	**TP/SEC/EA/ENR/EB-02-X**
Summary	Check that an EA discards a enrolment request signed by a signer_id with type set to an other value than 'self', 'certificate' or ' certificate_chain'.
Reference	ETSI TS 102 941 [2], clause 6.2.2.3
Config Id	CF01, CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to MSG_ENRREQ_TS
            containing signer
                containing type
                    set to X_INVALID_SUBJECT_TYPE
    }
    then {
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'request_denied'
    }
}
```

Variants

X	**PICS**	**X_INVALID_SUBJECT_TYPE**
A	Not PIC_Verify_SelfSigned	self(0)
B		certificate_digest_with_ecdsap224(1)
C		certificate_digest_with_ecdsap256(2)
D		certificate_digest_with_other_algorithm(5)
E		ANY OTHER (128)

TP Id	**TP/SEC/EA/ENR/EB-03-X**
Summary	Check that an EA discards an enrolment request with a subject type other than 'sec_data_exch_csr'.
Reference	IEEE P1609.2/D12 [1], 5.5.3.3, ETSI TS 102 867 [3], clause 5.1.2.1, IEEE P1609.2/D12 [1], clause 6.3.7
Config Id	CF01
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to MSG_ENRREQ_TS
            containing unsigned_csr
                containing subject_type
                    set to X_INVALID_SUBJECT_TYPE
                containing type_specific_data
                    containing X_INVALID_SCOPE
    }
    then {
        the IUT sends a valid CertificateRequestError set to X_RESPONSE
            containing reason
                set to 'request_denied'
    }
}
```

Variants

#	**X_INVALID_SUBJECT_TYPE**	**X_INVALID_SCOPE**
A	sec_data_exch_anonymous (0)	AnonymousScope
B	sec_data_exch_identified_not_localized (1)	IdentifiedNotLocalizedScope
C	sec_data_exch_identified _localized (2)	IdentifiedLocalizedScope
D	wsa (4)	WsaCaScope
E	wsa_csr (5)	WsaCaScope
F	sec_data_exch_ca(6)	SecDataExchCaScope
G	wsa_ca (7)	WsaCaScope
H	crl_signer(8)	CRLSeries
I	root_ca (255)	RootCaScope
J	ANY OTHER (128)	omit

6.2.4 Authorization Authority

6.2.4.1 Normal Behavior

6.2.4.1.1 Scopes (Scope Kind and Scope Name)

TP Id	**TP/SEC/AA/AUTH/NB-01**
Summary	Check that an AA responds to an authorization request with ***an anonymous scope*** with a valid authorization certificate.
Reference	IEEE P1609.2/D12 [1], clauses 6.2.22, 6.3.6, 6.3.7 and 6.3.19
Config Id	CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateRequest (AuthorisationRequest) set to MSG_AUTHREQ_TS
            containing unsigned_csr
                containing subject_type
                    set to 'sec_data_exch_anonymous'
                containing type_specific_data
                    containing anonymous_scope
                    containing additional_data
                        set to 0x00 (length 0)
    }
    then {
        the IUT sends a valid CertificateResponse (AuthorisationResponse) set to MSG_AUTHRSP_IUT
            containing certificates[last].unsigned_certificate
                containing subject_type
                    set to 'sec_data_exch_anonymous'
                containing type_specific_data
                    containing anonymous_scope
                    containing additional_data
                        set to 0x00 (length 0)
    }
}
```

TP Id	**TP/SEC/AA/AUTH/NB-02**
Summary	Check that an AA responds to an authorization request with ***a localized scope with a name of different size*** with a valid authorization certificate.
Reference	IEEE P1609.2/D12 [1], clauses 6.2.22, 6.3.6, 6.3.7 and 6.3.19
Config Id	CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateRequest (AuthorisationRequest) set to MSG_AUTHREQ_TS
            containing unsigned_csr
                containing subject_type
                    set to 'sec_data_exch_identified_localized'
                containing type_specific_data.id_scope.name
                    set to SCOPE_NAME
    }
    then {
        the IUT sends a valid CertificateResponse (AuthorisationResponse) set to MSG_AUTHRSP_IUT
            containing certificates[last].unsigned_certificate
                containing subject_type
                    set to 'sec_data_exch_identified_localized'
                containing id_scope.name
                    set to ANY_VALUE_OR_NONE
    }
}
```

Variants

X	**SCOPE_NAME**
A	of length > 0 and < 32
B	of length 0
C	of length 1
D	of length 32

6.2.4.1.2 Expiration

TP Id	**TP/SEC/AA/AUTH/NB-03-X**
Summary	Check that AA responds to an authorization request with the validity period conformed to the request and to the enrolment certificate.
Reference	IEEE P1609.2/D12 [1], clauses 6.3.2 and 6.3.34
Config Id	CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a valid CertificateRequest (AuthorisationRequest) set to MSG_AUTHREQ_TS
        containing signer.certificate.unsigned_certificate
            containing expiration
                set to EXP_ENR_Cert
            containing lifetime
                set to LT_ENR_Cert
        containing unsigned_csr
            containing expiration
                set to EXP_AR
            containing lifetime
                set to LT_AR
    }
    then {
        the IUT sends a valid AuthorizationResponse
        containing certificates[last].unsigned_certificate
            containing expiration
                set to EXP_ARResp
            containing lifetime
                set to LT_AResp
    }
}
```

Variants

X	**EXP_ENR_Cert**	**LT_ENR_Cert**	**EXP_AR**	**LT_AR**
A	CLT+2Y	1M	CLT+2Y	1M
B	CLT+2Y	1Y	CLT+2Y	1M
C	CLT+2Y	1Y	CLT+1Y + 1M	1M
D	CLT+2Y	2Y	CLT+2Y	2Y
E	CLT+2Y	2Y	CLT + 1M	1M
F	CLT+2Y	4Y	CLT+2Y	2Y
G	CLT+2Y	4Y	CLT + 1M	1M

EXP_AResp	with EXP_AResp =< EXP_ENR_Cert AND EXP_AResp <=EXP_AR
LT_AResp	EXP_AResp - LT_AResp >= CLT and EXP_AResp - LT_AResp >= EXP_ENR_Cert – LT_ENR_Cert AND EXP_AResp - LT_AResp >= EXP_AR – LT_AR

6.2.4.2 Exceptional Behavior

6.2.4.2.1 Invalid Certificates or Certificate Chain Fields

TP Id	**TP/SEC/AA/AUTH/EB-01-X**
Summary	Check that an AA discards an authorization request signed **by a signer_id with type set to an other value than 'certificate' or ' certificate_chain'.**
Reference	ETSI TS 102 941 [2] (V1.1.1), clause 6.2.2.3
Config Id	CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to MSG_AUTHREQ_TS
            containing signer
                containing type
                    set to X_INVALID_SIGNER_TYPE
    }
    then {
        the IUT sends a valid CertificateRequestError set to MSG_AUTHERR_IUT
            containing reason
                set to 'request_denied'
    }
}
```

Variants

X	**X_INVALID_SIGNER_TYPE**
A	self(0)
B	certificate_digest_with_ecdsap224(1)
C	certificate_digest_with_ecdsap256(2)
D	certificate_digest_with_other_algorithm(5)
E	ANY OTHER (128)

6.2.4.2.2 Invalid Scopes (Subject Type and Scope Name)

TP Id	**TP/SEC/AA/AUTH/EB-02-X**
Summary	Check that an AA discards an authorization request with a subject type other than 'sec_data_exch_anonymous' or 'sec_data_exch_identified_localized'.
Reference	IEEE P1609.2/D12 [1], clauses 5.5.3.3 and 6.3.7, ETSI TS 102 867 [3], clause 5.1.2.1
Config Id	CF02
PICS Selection	

Initial conditions

```
with {
    the IUT in operational state
}
```

Expected behaviour

```
ensure that {
    when {
        the IUT receives a CertificateRequest set to MSG_AUTHREQ_TS
            containing signer.certificate.unsigned_certificate.sec_data_exch_ca_scope.permitted_subject_types
                set to X_PERMITTED_SUBJECT_TYPES
            containing unsigned_csr
                containing subject_type
                    set to X_INVALID_SUBJECT_TYPE
                containing type_specific_data
                    containing X_INVALID_SCOPE
    }
    then {
        the IUT sends a valid CertificateRequestError set to MSG_AUTHERR_IUT
            containing reason
                set to 'request_denied'
    }
}
```

Variants

X	X_PERMITTED_SUBJECT_TYPES	X_INVALID_SUBJECT_TYPE	X_INVALID_SCOPE
A	sec_data_exch_identified_localized **and** sec_data_exch_anonymous	sec_data_exch_identified_not_localized (1)	IdentifiedNotLocalizedScope
B	sec_data_exch_identified_localized **and** sec_data_exch_anonymous	sec_data_exch_csr (3)	SecDataExchCaScope
C	sec_data_exch_identified_localized **and** sec_data_exch_anonymous	wsa (4)	WsaCaScope
D	sec_data_exch_identified_localized **and** sec_data_exch_anonymous	wsa_csr (5)	WsaCaScope
E	sec_data_exch_identified_localized **and** sec_data_exch_anonymous	sec_data_exch_ca(6)	SecDataExchCaScope
F	sec_data_exch_identified_localized **and** sec_data_exch_anonymous	wsa_ca (7)	WsaCaScope
G	sec_data_exch_identified_localized **and** sec_data_exch_anonymous	crl_signer(8)	CRLSeries
H	sec_data_exch_identified_localized **and** sec_data_exch_anonymous	root_ca (255)	RootCaScope
I	sec_data_exch_identified_localized **and** sec_data_exch_anonymous	ANY OTHER (128)	omit
J	sec_data_exch_identified_localized	sec_data_exch_anonymous (0)	AnonymousScope
K	sec_data_exch_anonymous	sec_data_exch_identified _localized (2)	IdentifiedLocalizedScope

History

<table>
<tr><th colspan="3">Document history</th></tr>
<tr><td>V1.1.1</td><td>July 2013</td><td>Publication</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>