ETSI TS 119 172-3 V1.1.1 (2019-12)

Electronic Signatures and Infrastructures (ESI);
Signature Policies;
Part 3: ASN.1 format for signature policies

Reference
DTS/ESI-0019172-3

Keywords
ASN.1, e-commerce, electronic signature,
policies, trust services

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

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables 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 (https://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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Electronic Signatures and Infrastructures (ESI).

The present document is part 3 of a multi-part deliverable. Full details of the entire series can be found in part 1 [i.2].

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document defines an ASN.1 format of machine readable signature policies based on the building blocks that define technical constraints on digital signatures and are specified in ETSI TS 119 172-1 [i.2].

Pure signature applicability rules, directly related to procedural constraints imposed by business processes, are out of the scope of the present document which does not define ASN.1 elements for the building blocks specified in ETSI TS 119 172-1 [i.2] defining only applicability rules.

For each element of the machine readable signature policy, the present document references to the semantics described in ETSI TS 119 172-2 [3] and defines the corresponding ASN.1 syntax.

The present document defines elements which can be used to describe technical constraints on signature creation, signature validation, and signature augmentation. These elements are designed in a way that it is possible to generate ASN.1 documents that include components of a signature generation policy, or/and signature validation policy, and/or signature augmentation policy.

An ASN.1 document conformant to the present specification, defines constraints (on generation, augmentation, validation, any combination of two of them, or the three of them) that one signature has to meet.

NOTE: Complex business processes, where several digital signatures need to be managed, having to meet different set of technical constraints, will require several ASN.1 documents conformant to the present document, each one defining one of these sets of technical constraints.

It is out of the scope to specify mechanisms for protecting the integrity of the machine-readable signature policy documents specified in the present document.

2 References

2.1 Normative references

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

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

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

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

[1] ETSI EN 319 122-1:"Electronic Signatures and Infrastructures (ESI); CAdES digital signatures; Part 1: Building blocks and CAdES baseline signatures".

[2] ETSI TS 119 612: "Electronic Signatures and Infrastructures (ESI); Trusted Lists".

[3] ETSI TS 119 172-2:"Electronic Signatures and Infrastructures (ESI); Signature Policies; Part 2: XML format for signature policies".

[4] Recommendation ITU-T X.680 (2015): "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation".

[5] Recommendation ITU-T X.690 (2015): "Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)".

[6] IETF RFC 5646: "Tags for Identifying Languages".

[7] IETF RFC 5912 (2010): "New ASN.1 Modules for the Public Key Infrastructure Using X.509 (PKIX)".

[8] W3C Recommendation: "XML Schema Part 2: Datatypes Second Edition". October 2004.

NOTE: See https://www.w3.org/TR/xmlschema-2/.

2.2 Informative references

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

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

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

[i.1] ETSI TS 119 102-1: "Electronic Signatures and Infrastructures (ESI); Procedures for Creation and Validation of AdES Digital Signatures; Part 1: Creation and Validation".

[i.2] ETSI TS 119 172-1:"Electronic Signatures and Infrastructures (ESI); Signature Policies; Part 1: Building blocks and table of contents for human readable signature policy documents".

[i.3] ETSI TS 119 192: "Electronic Signatures and Infrastructures (ESI); AdES related Uniform Resource Identifier".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI TS 119 172-1 [i.2] and the following apply:

signature applicability rules: set of rules, applicable to one or more digital signatures, that defines the requirements for determination of whether a signature is fit for a particular business or legal purpose

signature augmentation constraint: criteria used when augmenting a digital signature

signature augmentation policy: set of signature augmentation constraints

signature creation application: application within the signature creation system that creates the AdES digital signature and relies on the signature creation device to create a digital signature value

signature creation constraint: criteria used when creating a digital signature

signature creation policy: set of signature creation constraints processed or to be processed by the signature creation application

signature validation application: application that validates a signature against a signature validation policy, and that outputs a status indication (i.e. the signature validation status) and a signature validation report

signature validation constraint: technical criteria against which a digital signature can be validated

EXAMPLE: As specified in ETSI TS 119 102-1 [i.1].

signature validation policy: set of signature validation constraints processed or to be processed by the signature validation application

3.2 Symbols

Void.

3.3 Abbreviations

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

ASN.1	Abstract Syntax Notation One
BER	Basic Encoding Rules
DER	Distinguished Encoding Rules
OID	Object IDentifier
URI	Uniform Resource Identifier
XML	eXtensible Markup Language

4 ASN.1 syntax for machine processable signature policy document

4.1 Introduction

4.1.1 Technical approach

The present document takes as starting point the contents of ETSI TS 119 172-1 [i.2], which defines the building blocks of a human readable signature policy document. These building blocks are of two types:

- Building blocks defining applicability rules, which are the procedural constraints enforced by the business processes where the digital signatures are used. These procedural constraints, if not satisfied, could prevent further processing (in other words, accepting for the purpose of the business) a certain signed document even if the digital signature is technically valid.
- Building blocks defining technical constraints, related with technical aspects of the digital signature and its technical validation (signature format, signature attributes, constraints on certificates, time-stamp tokens, revocation material data, etc.).

The present document specifies an ASN.1 format for the building blocks specified in ETSI TS 119 172-1 [i.2], which define technical constraints, and allows building documents which define technical constraints in a machine-readable format.

The ASN.1 elements defined within the present contain information that clearly signal whether the constraints that they define apply to the generation of a signature, the validation of a signature, the augmentation of a signature, any combination of two of the former, or to the three of them. Therefore, the ASN.1 documents built using the present document may contain components of signature generation policy, or/and signature validation policy, and/or signature augmentation policy.

4.1.2 ASN.1 module

Annex A defines the ASN.1 module which describes the elements defined in the present document.

4.1.3 ASN.1 encoding

4.1.3.1 DER

Distinguished Encoding Rules (DER) for ASN.1 types shall be as specified in Recommendation ITU-T X.690 [5].

4.1.3.2 BER

If Basic Encoding Rules (BER) are used for some ASN.1 types, it shall be as specified in Recommendation ITU-T X.690 [5].

4.1.4 ASN.1 type to allow extensions: the `Other` type

Semantics

The `Other` type shall contain an element of a type as defined by the `OTHER.&id`.

The `Other` type shall be parametized by a set of allowed `OTHER.&id` types, the `MyOtherSet`.

The `MyOtherSet` may be extensible, i.e. contains the "…".

NOTE: The `Other` type allows to extends specific types later with specific elements.

Syntax

The `OTHER` class and the `Other` type shall be as defined in annex A and is copied below for information.

```
OTHER ::= CLASS {
    &id OBJECT IDENTIFIER UNIQUE,
    &Value OPTIONAL }
WITH SYNTAX {
    OTHER-ID &id
    [OTHER-TYPE &Value] }

Other{OTHER:MyOtherSet} ::= SEQUENCE {
  otherId OTHER.&id({MyOtherSet}),
  otherValue OTHER.&Value({MyOtherSet}{@otherId}) OPTIONAL }
```

4.2 The `SignaturePolicy` type

4.2.1 Semantics

The semantics shall be as in clause 4.2.1 of ETSI TS 119 172-2 [3].

4.2.2 Syntax

The `SignaturePolicy` type shall be as defined in annex A and is copied below for information.

```
SignaturePolicy ::= SEQUENCE {
    digest Digest,
    policyComponents PolicyComponents}
```

The element of type `Digest` shall be as specified in clause 4.3.2.

The element of type `PolicyComponents` shall be as specified in clause 4.4.2.

4.3 The `Digest` type

4.3.1 Semantics

The semantics shall be as in clause 4.3.1 of ETSI TS 119 172-2 [3].

4.3.2 Syntax

The `Digest` type shall be as defined in annex A and is copied below for information.

```
Digest ::= OtherHashAlgAndValue
```

The type `OtherHashAlgAndValue` shall be as specified in ETSI EN 319 122-1 [1].

An element of type `OtherHashAlgAndValue` contains a `hashValue` element and a `hashAlgorithm` element. The `hashValue` element shall contain the result of applying the algorithm identified in the corresponding `hashAlgorithm` element on the binary encoding of the `PolicyComponents` element specified in clause 4.4.2.

4.4 The `PolicyComponents` type

4.4.1 Semantics

The semantics shall be as in clause 4.4.1 of ETSI TS 119 172-2 [3].

4.4.2 Syntax

The `PolicyComponents` type shall be as defined in annex A and is copied below for information.

```
PolicyComponents ::= SEQUENCE {
    geneneralDetails  GeneralDetails,
    policyRules PolicyRules }
```

The element of type `GeneralDetails` shall be as specified in clause 4.5.2.

The element of type `PolicyRules` shall be as specified in clause 4.14.2.

4.5 The `GeneralDetails` type

4.5.1 Semantics

The semantics shall be as in clause 4.5.1 of ETSI TS 119 172-2 [3].

4.5.2 Syntax

The `GeneralDetails` type shall be as defined in annex A and is copied below for information.

```
GeneralDetails ::= SEQUENCE {
    sigPolicyDetails  SigPolicyDetails,
    authorityDetails [0] AuthorityDetails OPTIONAL,
    otherDetails [1] OtherDetails OPTIONAL }
```

The element of type `SigPolicyDetails` shall be as specified in clause 4.6.2.

The element of type `AuthorityDetails` shall be as specified in clause 4.7.2.

The element of type `OtherDetails` shall be as specified in clause 4.13.2.

4.6 The `SigPolicyDetails` type

4.6.1 Semantics

The semantics shall be as in clause 4.6.1 of ETSI TS 119 172-2 [3].

4.6.2 Syntax

The `SigPolicyDetails` type shall be as defined in annex A and is copied below for information.

```
SigPolicyDetails ::= SEQUENCE {
    policyIdentifier OBJECT IDENTIFIER,
    policyName InternationalNames,
    distributionPoints DistributionPoints OPTIONAL }

InternationalNames ::= SEQUENCE OF MultiLangText

MultiLangText ::= SEQUENCE {
    lang PrintableString,
    text UTF8String }

DistributionPoints ::= SEQUENCE SIZE (1..MAX) OF IA5String
```

The `policyIdentifier` element shall contain an object-identifier that uniquely identifies the signature policy.

The `policyName` element shall contain one or more names of the signature policy. Each name shall be qualified with an indication of a language, as per the definition of `MultiLangText`.

The `lang` element shall contain a tag conformant to IETF RFC 5646 [6] and in lower case, that identifies the language in which the content of the `text` element is expressed.

The `distributionPoints` child element shall contain one or more URIs where the signature policy can be reached.

4.7 The `AuthorityDetails` type

4.7.1 Semantics

The semantics shall be as in clause 4.7.1 of ETSI TS 119 172-2 [3].

4.7.2 Syntax

The `AuthorityDetails` type shall be as defined in annex A and is copied below for information.

```
AuthorityDetails ::= SEQUENCE {
    name [0] Name OPTIONAL,
    tradeName [1] TradeName OPTIONAL,
    postalAddresses PostalAddresses,
    electronicAddresses ElectronicAddresses,
    contactPersons [2] ContactPersons OPTIONAL}
```

The element of type `Name` shall be as specified in clause 4.8.2.

The element of type `TradeName` shall be as specified in clause 4.9.2.

The element of type `PostalAddresses` shall be as specified in clause 4.10.2.

The element of type `ElectronicAddresses` shall be as specified in clause 4.11.2.

The element of type `ContactPersons` shall be as specified in clause 4.12.2.

4.8 The `Name` type

4.8.1 Semantics

The semantics shall be as in clause 4.8.1 of ETSI TS 119 172-2 [3].

4.8.2 Syntax

The `Name` type shall be as defined in annex A and is copied below for information.

```
Name ::= InternationalNames
```

The element of type `InternationalNames` shall be as specified in clause 4.6.2.

4.9 The `TradeName` type

4.9.1 Semantics

The semantics shall be as in clause 4.9.1 of ETSI TS 119 172-2 [3].

4.9.2 Syntax

The `TradeName` type shall be as defined in annex A and is copied below for information.

```
TradeName::= InternationalNames
```

The type `InternationalNames` shall be as specified in clause 4.6.2.

4.10 The `PostalAddresses` type

4.10.1 Semantics

The semantics shall be as in clause 4.10.1 of ETSI TS 119 172-2 [3].

4.10.2 Syntax

The `PostalAddresses` type shall be as defined in annex A and is copied below for information.

```
PostalAddresses ::= SEQUENCE OF PostalAddress
```

The type `PostalAddress` shall be as specified in ETSI EN 319 122-1 [1].

4.11 The `ElectronicAddresses` type

4.11.1 Semantics

The semantics shall be as in clause 4.11.1 of ETSI TS 119 172-2 [3].

4.11.2 Syntax

The `ElectronicAddresses` type shall be as defined in annex A and is copied below for information.

```
ElectronicAddresses::= SEQUENCE OF ElectronicAddresse

ElectronicAddresse ::= MultiLangURI

MultiLangURI ::= SEQUENCE {
    lang PrintableString,
    uri IA5String }
```

Each element of type `ElectronicAddresses` shall be qualified with an indication of a language, as per the definition of `MultiLangURI`.

The `lang` element shall contain a tag conformant to IETF RFC 5646 [6] and in lower case, that identifies the language in which the content pointed-to by the URI contained in the `uri` element is expressed.

4.12 The `ContactPersons` type

4.12.1 Semantics

The semantics shall be as in clause 4.12.1 of ETSI TS 119 172-2 [3].

4.12.2 Syntax

The `ContactPersons` type shall be as defined in annex A and is copied below for information.

```
ContactPersons ::= SEQUENCE OF ContactPerson

ContactPerson ::= SEQUENCE {
    name UTF8String,
    electronicAddresses ElectronicAddresses,
    phoneNumbers SEQUENCE OF PrintableString}
```

An element of type `ElectronicAddresses` shall be as specified in clause 4.11.2.

Each `PrintableString` value in the `phoneNumbers` element shall contain exactly one phone number of the contact person. Each phone number shall start with the '+' character followed by the country prefix.

4.13 The `OtherDetails` type

4.13.1 Semantics

The semantics shall be as in clause 4.13.1 of ETSI TS 119 172-2 [3].

4.13.2 Syntax

The `OtherDetails` type shall be as defined in annex A and is copied below for information.

```
OtherDetails ::= SEQUENCE {
    dateOfIssue GeneralizedTime,
    signingPeriod [0] SigningPeriod OPTIONAL,
    others [1] SEQUENCE SIZE (1..MAX) OF Other{{OtherDetailsOtherSet}} OPTIONAL}

SigningPeriod ::= SEQUENCE {
    notBefore   GeneralizedTime,
    notAfter    GeneralizedTime OPTIONAL }

OtherDetailsOtherSet OTHER ::= {...}
```

The elements of type `GeneralizedTime` shall be as specified in Recommendation ITU-T X.680 [4].

NOTE: For the moment there are no `OTHER.&id` defined for the `OtherDetailsOtherSet`.

4.14 The `PolicyRules` type

4.14.1 Semantics

The semantics shall be as in clause 4.14.1 of ETSI TS 119 172-2 [3].

4.14.2 Syntax

The `PolicyRules` type shall be as defined in annex A and is copied below for information.

```
PolicyRules  ::= SEQUENCE OF PolicyRuleWithScope

PolicyRuleWithScope ::= SEQUENCE {
    rule   SigPolicyRule,
    scope  SigPolicyScope OPTIONAL}

SigPolicyRule ::= CHOICE {
    commitmentRules            [0]   CommitmentRules,
    basicRule                  [1]   BasicRule }

BasicRule ::= CHOICE {
    dataToBeSignedRules        [0]   DataToBeSignedRules,
    sigToDTBSRelationRules     [1]   SigToDTBSRelationRules,
    dTBSCardinality            [2]   DTBSCardinality,
    sigDTBSRelativePosition    [3]   SigDTBSRelativePosition,
    sigFormatsAndLevels        [4]   SigFormatsAndLevels,
    augmentationRules          [5]   AugmentationRules,
    signingCertRules           [6]   SigningCertRules,
    timeEvidencesRules         [7]   TimeEvidencesRules,
    signerAttributeConstraints [8]   SignerAttributeConstraints,
    qualifyingAttributesRules  [9]   QualifyingAttributesRules,
    sCDLoARules                [10]  SCDLoARules,
    cryptoSuitesRules          [11]  CryptoSuitesRules,
    otherRules                 [12]  Other{{BasicRuleOtherSet}}
}


SigPolicyScope  ::= ENUMERATED {
    generation      (0),
    validation      (1),
    augmentation    (2) }

BasicRuleOtherSet OTHER ::= {...}
```

The element of type `CommitmentRule`s shall be as specified in clause 4.15.2.

The element of type `DataToBeSignedRules` shall be as specified in clause 4.16.2.

The element of type `SigToDTBSRelationRules` shall be as specified in clause 4.17.2.

The element of type `DTBSCardinality` shall be as specified in clause 4.18.2.

The element of type `SigDTBSRelativePosition` shall be as specified in clause 4.19.2.

The element of type `SigFormatAndLevels` shall be as specified in clause 4.20.2.

The element of type `AugmentationRules` shall be as specified in clause 4.21.2.

The element of type `SigningCertRules` shall be as specified in clause 4.24.4.2.

The element of type `TimeEvidLoARules` shall be as specified in clause 4.25.2.

The element of type `SignerAttributeConstraints` shall be as specified in clause 4.26.2.

The element of type `QualifyingAttributesRules` shall be as specified in clause 4.27.2.

The element of type `SCDLoARules` shall be as specified in clause 4.28.2.

The element of type `CryptoSuitesRules` shall be as specified in clause 4.29.2.

NOTE: For the moment there are no `OTHER.&id` defined for the `BasicRuleOtherSet`.

4.15 The CommitmentRules type

4.15.1 Semantics

The semantics shall be as in clause 4.15 of ETSI TS 119 172-2 [3].

4.15.2 Syntax

The CommitmentRules type shall be as defined in annex A and is copied below for information.

```
CommitmentRules ::= SEQUENCE OF CommitmentRule

CommitmentRule ::= SEQUENCE {
    commitments                    SEQUENCE OF Commitment,
    matchingIndicator               MatchingIndicator,
    basicRules                      SEQUENCE OF BasicRule}

Commitment ::= SEQUENCE {
    commitmentIdentifier           CommitmentTypeIdentifier,
    details                        SEQUENCE SIZE (1..MAX) OF MultiLangString OPTIONAL}

MultiLangString ::= SEQUENCE {
    lang PrintableString,
    uri UTF8String }

MatchingIndicator ::= ENUMERATED {
    all       (0),
    none      (1),
    atLeastOne(2) }
```

Each Commitment element shall contain an identifier of one commitment taken by the signer (CommitmentTypeIdentifier), and may also contain additional textual details in different languages.

Each entry of details shall be qualified with an indication of a language, as per the definition of MultiLangURI, in clause 4.11.2.

The element of type CommitmentTypeIdentifier shall be as specified in ETSI EN 319 122-1 [1].

The element of type BasicRule shall be as specified in clause 4.14.2.

4.16 The DataToBeSignedRules type

4.16.1 Semantics

The semantics shall be as in clause 4.16.1 of ETSI TS 119 172-2 [3].

4.16.2 Syntax

The DataToBeSignedRules type shall be as defined in annex A and is copied below for information.

```
DataToBeSignedRules ::= SEQUENCE OF DataToBeSignedRule

DataToBeSignedRule ::= SEQUENCE {
    anyOfMimeType  [0] SEQUENCE SIZE (1..MAX) OF UTF8String OPTIONAL,
    noneOfMimeType [1] SEQUENCE SIZE (1..MAX) OF UTF8String OPTIONAL }
```

The DataToBeSignedRule shall not be empty.

4.17 The SigToDTBSRelationRules type

4.17.1 Semantics

The semantics shall be as in clause 4.17.1 of ETSI TS 119 172-2 [3].

4.17.2 Syntax

The SigToDTBSRelationRules type shall be as defined in annex A and is copied below for information.

```
SigToDTBSRelationRules::= SEQUENCE {
    dTBSCardinality            DTBSCardinality,
    sigDTBSRelativePosition [0] SigDTBSRelativePosition OPTIONAL,
    sigFormatsAndLevels     [1] SigFormatsAndLevels OPTIONAL }
```

The element of type DTBSCardinality shall be as specified in clause 4.18.2.

The element of type SigDTBSRelativePosition shall be as specified in clause 4.19.2.

The element of type SigFormatsAndLevels shall be as specified in clause 4.20.2.

4.18 The DTBSCardinality type

4.18.1 Semantics

The semantics shall be as in clause 4.18.1 of ETSI TS 119 172-2 [3].

4.18.2 Syntax

The DTBSCardinality type shall be as defined in annex A and is copied below for information.

```
DTBSCardinality::= SEQUENCE {
    maxDTBSNumber [0] MaxDTBSNumber OPTIONAL,
    minDTBSNumber [1] MinDTBSNumber OPTIONAL }

MaxDTBSNumber ::= SEQUENCE {
    dTBSNumber INTEGER,
    maxValueQualifier MaxValueQualifier }

MaxValueQualifier ::= ENUMERATED {
    lessThan       (0),
    lessOrEqualTo  (1),
    equal          (2)  }

MinDTBSNumber ::= SEQUENCE {
    dTBSNumber INTEGER,
    minValueQualifier MinValueQualifier }


MinValueQualifier ::= ENUMERATED {
    higherThan      (0),
    higherOrEqualTo (1),
    equal           (2)  }
```

4.19 The SigDTBSRelativePosition type

4.19.1 Semantics

The semantics shall be as in clause 4.20.1 of ETSI TS 119 172-2 [3].

4.19.2 Syntax

The `SigDTBSRelativePosition` type shall be as defined in annex A and is copied below for information.

```
SigDTBSRelativePosition ::= SEQUENCE OF SigDTBSRelativePositionValue

SigDTBSRelativePositionValue::= ENUMERATED {
    envelopingSig             (0),
    envelopedSig              (1),
    detachedSig               (2),
    aSiC                      (3) }
```

The value of `SigDTBSRelativePosition` element shall be a list of one or more SigDTBSRelativePositionValue entries. Each entry in the list shall indicate either that the signature is enclosed in an ASiC container (value `"ASiC"`), or the relative position of the signature and one of the data object it signs (`"EnvelopingSig"` for enveloping signatures, `"EnvelopedSig"` for enveloped signatures, and `"DetachedSig"` for detached signatures).

NOTE: As a result of their characteristics, a policy for XML and XAdES digital signatures could include up to three values, other than ASiC. An XML or a XAdES digital signature can be enveloped in one of the data objects it signs, enveloping a second signed data object, and detached from a third signed data object.

If the value `"ASiC"` is present in the list, no other value shall be present. The presence of this value means that the signature shall appear in an ASiC container.

For each value within the list other than `"ASiC"`, the signature shall be placed with respect to one or more signed digital objects as indicated by that value.

EXAMPLE: If the value of `SigDTBSRelativePosition` contains `"EnvelopingSig”` and `“DetachedSig",` the signature policy establishes that the signature will be enveloping one or more signed data objects and will be detached from one or more signed data objects.

4.20 The `SigFormatsAndLevels` type

4.20.1 Semantics

The semantics shall be as in clause 4.20.1 of ETSI TS 119 172-2 [3].

4.20.2 Syntax

The `SigFormatsAndLevels` type shall be as defined in annex A and is copied below for information.

```
SigFormatsAndLevels ::= SEQUENCE {
    sigFormats [0] SEQUENCE SIZE (1..MAX) OF IA5String OPTIONAL,
    sigLevels [1] SEQUENCE SIZE (1..MAX) OF IA5String OPTIONAL}
```

Instances of `SigFormatAndLevels` shall not be empty.

Absence of the `sigFormats` element shall indicate that any format is allowed.

The `sigFormats` shall contain URIs identifying the admitted signature formats.

Absence of the `sigLevels` element shall indicate that any level is allowed.

The `sigLevels` shall contain URIs identifying the admitted signature levels.

EXAMPLE: ETSI TS 119 192 [i.3] defines URIs for different signature formats and levels.

4.21 The AugmentationRules type

4.21.1 Semantics

The semantics shall be as in clause 4.21.1 of ETSI TS 119 172-2 [3].

4.21.2 Syntax

The DTBSCardinality type shall be as defined in annex A and is copied below for information.

```
AugmentationRules ::= SEQUENCE {
    previousValidationRequired BOOLEAN,
    levelID IA5String,
    augQualifier AugmentationQualifier }

AugmentationQualifier ::= ENUMERATED {
    thisLevel     (0),
    minLevel      (1),
    maxLevel      (2)  }
```

The element levelID shall contain an URI describing the augmentation level.

EXAMPLE: ETSI TS 119 192 [i.3] defines URIs for different signature formats and levels.

4.22 Types for defining constraints on certificates' trust and certificates revocation status

4.22.1 Introduction

The present clause defines four types:

1) TrustAnchorsList which defines the trust anchors.

2) NameConstraints which defines constraints on the names of entities.

3) PolicyConstraints for defining constraints on certificate policies.

4) CertificateTrustTrees which defines constraints on the trust conditions required to certificates.

4.22.2 The TrustAnchors type

4.22.2.1 Semantics

The semantics shall be as in clause 4.22.2.1 of ETSI TS 119 172-2 [3].

4.22.2.2 Syntax

The TrustAnchors type shall be as defined in annex A and is copied below for information.

```
TrustAnchors ::= SEQUENCE OF TrustAnchor

TrustAnchor ::= CHOICE  {
    certificate CertAndReliableTime,
    tAInTrustedList TAsInTrustedList }

CertAndReliableTime ::= SEQUENCE {
    cert Certificate,
    reliableUntil GeneralizedTime OPTIONAL }

TAsInTrustedList::= SEQUENCE {
    uri IA5String,
    serviceTypes [0] SEQUENCE SIZE (1..MAX) OF IA5String OPTIONAL,
    serviceStatuses [1] SEQUENCE SIZE (1..MAX) OF IA5String OPTIONAL }
```

The element of type `Certificate` shall be as specified in IETF RFC 5912 [7].

The element `uri` shall contain a reference to the Trusted List.as defined in ETSI TS 119 612 [2].

The element `serviceType`, when present, shall contain a non-empty list of URIs expressing acceptable service types.

The element `serviceStatuses`, when present, shall contain a non-empty list of URIs expressing acceptable service statuses.

4.22.3 The NameConstraints type

4.22.3.1 Semantics

The semantics shall be as in clause 4.22.3.1 of ETSI TS 119 172-2 [3].

4.22.3.2 Syntax

The `NameConstraints` type shall be as defined in annex A and is copied below for information.

```
NameConstraints ::= SEQUENCE {
          permittedSubtrees       [0]     GeneralSubtrees OPTIONAL,
          excludedSubtrees        [1]     GeneralSubtrees OPTIONAL }

      GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

      GeneralSubtree ::= SEQUENCE {
           base                    GeneralName,
           minimum         [0]     BaseDistance DEFAULT 0,
           maximum         [1]     BaseDistance OPTIONAL }

      BaseDistance ::= INTEGER (0..MAX)
```

The element of type `GeneralName` shall be as specified in IETF RFC 5912 [7].

4.22.4 The PolicyConstraints type

4.22.4.1 Semantics

The semantics shall be as in clause 4.22.4.1 of ETSI TS 119 172-2 [3].

4.22.4.2 Syntax

The `PolicyConstraints` type shall be as defined in annex A and is copied below for information.

```
PolicyConstraints ::= SEQUENCE {
        requireExplicitPolicy           [0] SkipCerts OPTIONAL,
        inhibitPolicyMapping            [1] SkipCerts OPTIONAL }

SkipCerts ::= INTEGER (0..MAX)
```

4.22.5 The CertificateTrustTrees type

4.22.5.1 Semantics

The semantics shall be as in clause 4.22.5.1 of ETSI TS 119 172-2 [3].

4.22.5.2 Syntax

The `CertificateTrustTrees` type shall be as defined in annex A and is copied below for information.

```
CertificateTrustTrees ::=   SEQUENCE OF CertificateTrustPoint
```

```
CertificateTrustPoint ::= SEQUENCE {
    trustAnchors        TrustAnchors,
    pathLenConstraint   [0] PathLenConstraint   OPTIONAL,
    acceptablePolicySet [1] AcceptablePolicySet OPTIONAL,
    nameConstraints     [2] NameConstraints     OPTIONAL,
    policyConstraints   [3] PolicyConstraints   OPTIONAL,
    userCertPath        [4] UserCertPath        OPTIONAL }


PathLenConstraint    ::=   INTEGER (0..MAX)

AcceptablePolicySet ::= SEQUENCE OF CertPolicyId

CertPolicyId ::= OBJECT IDENTIFIER

UserCertPath ::= CHOICE {
    asInSignature [0] BOOLEAN,
    path          [1] SEQUENCE OF Certificate }
```

The element of type `TrustAnchors` shall be as specified in clause 4.22.2.2.

The element of type `NameConstraints` shall be as specified in clause 4.22.3.2.

The element of type `PolicyConstraints` shall be as specified in clause 4.22.4.2.

4.23 Types for defining constraints on certificates' revocation status

4.23.1 Introduction

The present clause defines two types:

1) `CertificateRevReq` which defines constraints on the certificate revocation checks procedures.

2) `CertificateRevStatus` which defines constraints on the trust conditions required on the certificates' revocation data.

4.23.2 The CertificateRevReq type

4.23.2.1 Semantics

The semantics shall be as in clause 4.23.2.1 of ETSI TS 119 172-2 [3].

4.23.2.2 Syntax

The `CertificateRevReq` type shall be as defined in annex A and is copied below for information.

```
CertificateRevReq ::= SEQUENCE {
    endCertRevReq   EnuRevReq,
    caCerts         EnuRevReq }

EnuRevReq  ::= ENUMERATED {
    clrCheck    (0),
    ocspCheck   (1),
    bothCheck   (2),
    eitherCheck (3),
    noCheck     (4),
    other       (5)}
```

4.23.3 The CertificateRevTrust type

4.23.3.1 Semantics

The semantics shall be as in clause 4.23.3.1 of ETSI TS 119 172-2 [3].

4.23.3.2 Syntax

The `CertificateRevTrust` type shall be as defined in annex A and is copied below for information.

```
CertificateRevTrust ::= SEQUENCE {
    certificateRevReq CertificateRevReq,
    freshness         [0] Freshness OPTIONAL,
    sigCertIssuedByCAKeepsExpiredRevokedCertsInfo [1] Duration OPTIONAL }

Freshness ::= CHOICE {
    maxDifferenceRevocationAndValidation [0] Duration,
    timeAfterSignature                   [1] Duration }

Duration ::= PrintableString
```

The element of type `CertificateRevReq` shall be as specified in clause 4.23.2.2.

The value of the `timeAfterSignature` element shall be the time after the best-signature time as computed in ETSI TS 119 102-1 [i.1].

The value of the `sigCertIssuedByCAKeepsExpiredRevokedCertsInfo` element shall be a time indication.

If the `sigCertIssuedByCAKeepsExpiredRevokedCertsInfo` element is present then it is mandated that the signing certificate has been issued by a certification authority that keeps revocation notices for revoked certificates even after they have expired, for a period of time exceeding the value of this child element. If the `SigCertIssuedByCAKeepsExpiredRevokedCertsInfo` child is absent, this constraint is not mandated.

The content of an element of `Duration` type shall contain a printable string expressing the duration as defined for the `duration` type as defined in clause 3.2.6 of the description of the XML Schema [8].

NOTE: The `duration` type in the XMLSchema [8] is specified in the following way "PnYnMnDTnHnMnS" where:

- P indicates the period (required)
- nY indicates the number of years
- nM indicates the number of months
- nD indicates the number of days
- T indicates the start of a time section (required when specifying hours, minutes, or seconds)
- nH indicates the number of hours
- nM indicates the number of minutes
- nS indicates the number of seconds

4.24 The `SigningCertRules` type

4.24.1 Semantics

The semantics shall be as in clause 4.24.1 of ETSI TS 119 172-2 [3].

4.24.2 Syntax

The `SigningCertRules` type shall be as defined in annex A and is copied below for information.

```
SigningCertRules ::= SEQUENCE {
    signingCertTrustConditions      SigningCertTrustConditions,
    mandatedSigningCertInfo         MandatedSigningCertInfo OPTIONAL }
```

The element of type `CertificateRevReq` shall be as specified in clause 4.24.4.2.

The element of type `MandatedSigningCertInfo` shall be as defined in clause 4.24.3.2.

4.24.3 The `MandatedSigningCertInfo` type

4.24.3.1 Semantics

The semantics shall be as in clause 4.24.3.1 of ETSI TS 119 172-2 [3].

4.24.3.2 Syntax

The `MandatedSigningCertInfo` element shall be as defined in annex A and is copied below for information.

```
MandatedSigningCertInfo ::= ENUMERATED {
    signingCertOnly (0),
    fullPath        (1) }
```

4.24.4 The `SigningCertTrustConditions` type

4.24.4.1 Semantics

The semantics shall be as in clause 4.24.4.1 of ETSI TS 119 172-2 [3].

4.24.4.2 Syntax

The `SigningCertTrustConditions` type shall be as defined in annex A and is copied below for information.

```
SigningCertTrustConditions ::= SEQUENCE {
    signerTrustTrees CertificateTrustTrees,
    signerRevTrust   CertificateRevTrust }
```

The element of type `CertificateTrustTrees` shall be as specified in clause 4.22.5.2.

The element of type `CertificateRevTrust` shall be as specified in clause 4.23.3.2.

4.25 The `TimeEvidencesRules` type

4.25.1 Semantics

The semantics shall be as in clause 4.25.1 of ETSI TS 119 172-2 [3].

4.25.2 Syntax

The `TimeEvidLoARules` type shall be as defined in annex A and is copied below for information.

```
TimeEvidencesRules::= SEQUENCE OF RulesForSetOfEvidences

RulesForSetOfEvidences ::= SEQUENCE {
    evidenceIdentifiers     SEQUENCE OF IA5String,
    levelOfAssurance        IA5String,
    timeStampTrustCondition TimestampTrustCondition OPTIONAL }

TimestampTrustCondition ::= SEQUENCE {
    ttsCertificateTrustTrees    [0]     CertificateTrustTrees   OPTIONAL,
    ttsRevReq                   [1]     CertificateRevReq       OPTIONAL,
    ttsNameConstraints          [2]     NameConstraints         OPTIONAL,
    signatureTimestampDelay     [3]     DeltaTime               OPTIONAL }

DeltaTime ::= SEQUENCE {
    deltaSeconds    INTEGER,
    deltaMinutes    INTEGER,
    deltaHours      INTEGER,
```

```
    deltaDays       INTEGER }
```

The element of type `CertificateTrustTrees` shall be as specified in clause 4.22.5.2.

The element of type `CertificateRevReq` shall be as specified in clause 4.23.2.2.

The element of type `NameConstraints` shall be as specified in clause 4.22.3.2.

4.26 The `SignerAttributeConstraints` type

4.26.1 Semantics

The semantics shall be as in clause 4.26.1 of ETSI TS 119 172-2 [3].

4.26.2 Syntax

The `SignerAttributeConstraints` type shall be as defined in annex A and is copied below for information.

```
SignerAttributeConstraints ::= SEQUENCE {
    noSignerAttributesAllowed BOOLEAN,
    constraintsOnOneSetOfAttributes SEQUENCE OF AttributeSetConstraints }

AttributeSetConstraints ::= SEQUENCE {
    howCertAttribute             HowCertAttribute,
    attrCertificateTrustTrees   [0] CertificateTrustTrees   OPTIONAL,
    attrRevReq                  [1] CertificateRevTrust     OPTIONAL,
    attributeConstraints        [2] AttributeConstraints    OPTIONAL }

HowCertAttribute ::= ENUMERATED {
    claimedAttribute    (0),
    certifiedAttribtes  (1),
    signedAssertions    (2),
    any                 (3)

AttributeConstraints ::= SEQUENCE {
    attributeIdMustBePresent    [0] OBJECT IDENTIFIER  OPTIONAL,
    attributeValueConstraints   [1] SEQUENCE SIZE (1..MAX) OF
                                    Other{{AttributeValueConstraintsOtherSet}} OPTIONAL }

AttributeValueConstraintsOtherSet OTHER ::= {...}
```

The element of type `CertificateTrustTrees` shall be as specified in clause 4.22.5.2.

The element of type `CertificateRevTrust` shall be as specified in clause 4.23.3.2.

NOTE: For the moment there are no `OTHER.&id` defined for the `AttributeValueConstraintsOtherSet`.

4.27 The `QualifyingAttributesRules` type

4.27.1 Semantics

The semantics shall be as in clause 4.27.1 of ETSI TS 119 172-2 [3].

4.27.2 Syntax

The `QualifyingAttributesRules` type shall be as defined in annex A and is copied below for information.

```
QualifyingAttributesRules ::= SEQUENCE OF LevelAttributesRules

LevelAttributesRules ::= SEQUENCE {
    levelIdentifier    [0] IA5String,
    signedAttributes   [1] SignatureAttributes OPTIONAL,
```

```
    unsignedAttributes [2] SignatureAttributes OPTIONAL }

SignatureAttributes ::= SEQUENCE OF CHOICE {
    choice  [0] SEQUENCE OF SignatureAttribute,
    sigAttr [1] SignatureAttribute }

SignatureAttribute ::= SEQUENCE {
    identifier    OBJECT IDENTIFIER,
    mandatory     BOOLEAN }
```

4.28 The SCDLoARules type

4.28.1 Semantics

The semantics shall be as in clause 4.28.1 of ETSI TS 119 172-2 [3].

4.28.2 Syntax

The SCDLoARules type shall be as defined in annex A and is copied below for information.

```
SCDLoARules ::= IA5String
```

The SCDDLoARules shall be a URI value indicating the Level of Assurance of the signature creation device.

4.29 The CryptoSuitesRules type

4.29.1 Semantics

The semantics shall be as in clause 4.29.1 of ETSI TS 119 172-2 [3].

4.29.2 Syntax

The CryptoSuitesRules type shall be as defined in annex A and is copied below for information.

```
CryptoSuitesRules ::= SEQUENCE OF AlgConstraints

AlgConstraints ::= SEQUENCE {
    algID             OBJECT IDENTIFIER,
    usages            SEQUENCE SIZE (1..MAX) OF IA5String,
    minKeyLength  [0] INTEGER   OPTIONAL,
    minHashLength [1] INTEGER   OPTIONAL,
    other             Other{{AlgConstraintsOtherSet}} OPTIONAL
}

AlgConstraintsOtherSet OTHER ::= {...}
```

The usages element shall contain URIs as in the UsageType in clause 4.29.2 of ETSI TS 119 172-2 [3].

NOTE: For the moment there are no OTHER.&id defined for the AlgConstraintsOtherSet.

Annex A (normative): Signature policy ASN.1 module using X.680 ASN.1 syntax

The file at https://forge.etsi.org/rep/esi/x19_17203_ASN1_sign-policies/raw/v1.1.1/19172asn1Module.txt contains ASN.1 module `ETSI-SigPolicy-ASN1 { itu-t(0) identified-organization(4) etsi(0) sigpolicy-asn1(19172) id-mod(0) sigpolicy-syntax680(1)}`.

History

Document history		
V1.1.1	December 2019	Publication