ETSI TS 119 172-2 V1.1.1 (2019-12)

Electronic Signatures and Infrastructures (ESI);
Signature Policies;
Part 2: XML format for signature policies

Reference
DTS/ESI-0019172-2

Keywords
e-commerce, electronic signature, policies, trust services, XML

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 2 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 XML 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 XML elements for the building blocks specified in ETSI TS 119 172-1 [i.2] that define only applicability rules.

For each element of the machine readable signature policy, the present document specifies the semantics and the how to implement it in XML 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 XML documents that include components of a signature generation policy, or/and signature validation policy, and/or signature augmentation policy.

An XML 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 XML 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 TS 119 612: "Electronic Signatures and Infrastructures (ESI); Trusted Lists".

[2] IETF RFC 3061: "A URN Namespace of Object Identifiers".

[3] IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

[4] IETF RFC 6960: "X.509 Internet Public Key Infrastructure. Online Certificate Status Protocol - OCSP".

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 119 612: "Electronic Signatures and Infrastructures (ESI); Trusted Lists".

[i.4] ETSI TS 119 192: "Electronic Signatures and Infrastructures (ESI); AdES related Uniform Resource Identifier".

[i.5] Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures. OJ L 13, 19.1.2000, p. 12-20.

[i.6] Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC. OJ L 257, 28.8.2014, p. 73-114.

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:

EU qualified certificate: qualified certificate as specified in Directive 1999/93/EC [i.5] or in Regulation (EU) No 910/2014 [i.6] whichever is in force at the time of issuance

EU qualified trust service provider: trust service provider that meets the requirements for qualified trust service providers laid down in Regulation (EU) 910/2014 [i.6]

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:

BSP	Business Scoping Parameter
CA	Certification Authority
CRL	Certificate Revocation List
EUMS	European Union Member State
IETF	Internet Engineering Task Force
LOTL	List Of Trusted Lists
OCSP	Online Certificate Status Protocol
OID	Object IDentifier
RFC	Request For Comments
TL	Trusted List
TSP	Trusted Service Provider
URI	Universal Resource Identifier
URN	Universal Resource Name
XML	eXtensible Markup Language

4 XML 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 a XML 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 XML 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 XML 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 XML namespaces

The present document uses the URI namespaces listed below:

- http://uri.etsi.org/19172/v1.1.1
- http://www.w3.org/2000/09/xmldsig#
- http://uri.etsi.org/02231/v2#
- http://www.w3.org/2001/XMLSchema

The present document defines one XML Schema file, namely: "19172xmlSchema.xsd". See Annex B for details on their locations.

Table 1 shows the URIs of the namespaces used in the XML Schema definitions, mapped to their corresponding prefixes.

Table 1: Namespaces with constant prefixes

XML Namespace URI	Prefix
http://www.w3.org/2000/09/xmldsig#	ds
http://www.w3.org/2001/XMLSchema	xs
http://uri.etsi.org/02231/v2#"	tsl

Below follows a copy of the xs:schema element of the XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and that defines the namespace whose URI is http://uri.etsi.org/19172/v1.1.1.

```
<xs:schema targetNamespace="http://uri.etsi.org/19172/v1.1.1#"
xmlns:tsl="http://uri.etsi.org/02231/v2#" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://uri.etsi.org/19172/v1.1.1#"
elementFormDefault="qualified" attributeFormDefault="unqualified">

    <xs:import namespace="http://uri.etsi.org/02231/v2#"
schemaLocation="https://uri.etsi.org/02231/v3.1.2/ts_102231v030102_xsd.xsd"/>
    <xs:import namespace="http://www.w3.org/2000/09/xmldsig#"
schemaLocation="http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/xmldsig-core-schema.xsd"/>
```

4.1.3 XML type to allow extensions: the AnyType type

Semantics

The AnyType type shall have a content model allowing a sequence of arbitrary XML elements that (mixed with text) is of unrestricted length.

The AnyType type shall have a content model allowing for text content only.

The AnyType type shall have a content model allowing an element of this data type to bear an unrestricted number of arbitrary attributes.

NOTE: The AnyType data type is used throughout the remaining parts of the present document wherever the content of an XML element has been left open.

Syntax

The AnyType element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
    <xs:complexType name="AnyType" mixed="true">
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:any namespace="##any" processContents="lax"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##any"/>
    </xs:complexType>
```

4.2 The `SignaturePolicy` element

4.2.1 Semantics

This element shall contain:

1) A digest for securing the contents of the signature policy document, and an identifier of the digest algorithm used for computing it, as specified in clause 4.3.

2) An element that contains all the machine-processable components of the signature policy, as specified in clause 4.4.

4.2.2 Syntax

The `SignaturePolicy` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="SignaturePolicy" type="SignaturePolicyType" />

    <xs:complexType name="SignaturePolicyType">
        <xs:sequence>
            <xs:element ref="Digest" />
            <xs:element ref="PolicyComponents" />
        </xs:sequence>
    </xs:complexType>
```

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

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

4.3 The `Digest` element

4.3.1 Semantics

This element shall contain:

1) One identifier of a digest algorithm.

2) One digest value.

In the case that the structured document is an XML document, this element shall also contain:

1) A canonicalization algorithm.

4.3.2 Syntax

The `Digest` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
    <xs:element name="Digest" type="DigestDetailsType"/>
    <xs:complexType name="DigestDetailsType">
        <xs:sequence>
            <xs:element ref="ds:DigestMethod" />
            <xs:element ref="ds:DigestValue" />
            <xs:element ref="ds:CanonicalizationMethod" />
        </xs:sequence>
    </xs:complexType>
```

The digest value shall be computed on the output of applying the canonicalization algorithm identified in this component, to the `PolicyComponents` element specified in clause 4.4.

4.4 The `PolicyComponents` element

4.4.1 Semantics

This element shall contain an element containing all the rules defined by the signature policy itself, as specified in clause 4.14.

This element should also contain an element including general details, as specified in clause 4.5.

4.4.2 Syntax

The `PolicyComponents` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="PolicyComponents" type="PolicyComponentsType"/>
<xs:complexType name="PolicyComponentsType">
    <xs:sequence>
        <xs:element ref="GeneralDetails"/>
        <xs:element ref="PolicyRules"/>
    </xs:sequence>
</xs:complexType>
```

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

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

4.5 The `GeneralDetails` element

4.5.1 Semantics

This element shall contain an element containing details on the signature policy itself, as specified in clause 4.6.

This element may also contain:

1) An element containing details of the responsible authority of the signature policy, as specified in clause 4.7.

2) An element containing other details not defined within the present document, as specified in clause 4.13.

4.5.2 Syntax

The `GeneralDetails` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="GeneralDetails" type="GeneralDetailsType"/>
<xs:complexType name="GeneralDetailsType">
    <xs:sequence>
        <xs:element ref="SigPolicyDetails"/>
        <xs:element ref="AuthorityDetails" minOccurs="0"/>
        <xs:element ref="OtherDetails" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
```

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

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

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

4.6 The SigPolicyDetails element

4.6.1 Semantics

This element shall contain:

1) An element identifying the signature policy itself.

2) An element whose value shall be the name of the signature policy itself.

This element may also contain an element containing one or more distribution points of the signature policy itself.

4.6.2 Syntax

The SigPolicyDetails element and the SigPolicyDetailsType type shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="SigPolicyDetails" type="SigPolicyDetailsType"/>

<xs:complexType name="SigPolicyDetailsType">
    <xs:sequence>
        <xs:element name="PolicyIdentifier" type="xs:anyURI"/>
        <xs:element name="PolicyName" type="tsl:InternationalNamesType"/>
        <xs:element ref="tsl:DistributionPoints" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
```

The PolicyIdentifier child element shall contain an URI identifying the signature policy. If the signature policy is identified by an OID, the value of PolicyIdentifier child element shall be the OID value encoded as an URN as specified by the IETF RFC 3061 [2].

The PolicyName child element shall contain one or more names of the signature policy. Its contents shall be as specified in ETSI TS 119 612 [1].

NOTE: Instances of tsl:InternationalNamesType specified in ETSI TS 119 612 [1] are qualified with an indication of a language.

If present, the tsl:DistributionPoints child element shall contain one or more URIs where the signature policy can be reached. Its contents shall be as specified in ETSI TS 119 612 [1].

4.7 The AuthorityDetails element

4.7.1 Semantics

This element shall contain details of the authority responsible for the signature policy.

This element shall contain at least, one of the two following components:

1) An element including the name of the authority as specified in clause 4.8.

2) An element as specified in clause 4.9, including an official registration identifier, unambiguously identifying the authority, as registered in official records, where such a registered identifier does exist.

This element shall also contain the following components:

1) An element including a sequence of one or more postal addresses of the authority as specified in clause 4.10.

2) An element including one or more electronic addresses of the authority as specified in clause 4.11.

This element may also contain:

1) An element for providing details of one or more natural persons acting as contact as specified in clause 4.12.

4.7.2 Syntax

The `Authority` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="AuthorityDetails" type="AuthorityDetailsType"/>
<xs:complexType name="AuthorityDetailsType">
    <xs:sequence>
        <xs:element ref="Name" minOccurs="0"/>
        <xs:element ref="TradeName" minOccurs="0"/>
        <xs:element ref="PostalAddresses" />
        <xs:element ref="ElectronicAddresses" />
        <xs:element ref="ContactPersons" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
```

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

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

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

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

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

4.8 The `Name` element

4.8.1 Semantics

This element shall meet the requirements specified in clause 5.3.4 ("Scheme operator name") of ETSI TS 119 612 [1].

4.8.2 Syntax

The `Name` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="Name" type="tsl:InternationalNamesType" />
```

4.9 The `TradeName` element

4.9.1 Semantics

This element shall meet the requirements specified in clause 5.4.2 ("TSP trade name") of ETSI TS 119 612 [1] for the contents of TSP trade names for legal persons.

4.9.2 Syntax

The `TradeName` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="TradeName" type="tsl:InternationalNamesType"/>
```

4.10 The `PostalAddresses` element

4.10.1 Semantics

This element shall meet the requirements specified in clause 5.3.5.1 ("Scheme operator postal address") of ETSI TS 119 612 [1].

4.10.2 Syntax

The `PostalAddresses` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="PostalAddresses" type="tsl:PostalAddressListType"/>
```

4.11 The `ElectronicAddresses` element

4.11.1 Semantics

This element shall meet the requirements specified in clause 5.3.5.2 ("Scheme operator electronic address") of ETSI TS 119 612 [1].

4.11.2 Syntax

The `ElectronicAddresses` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="ElectronicAddresses" type="tsl:ElectronicAddressType"/>
```

NOTE: Instances of `tsl:ElectronicAddressType` specified in ETSI TS 119 612 [1] have an indication of the language.

4.12 The `ContactPersons` element

4.12.1 Semantics

This element shall include the details of one or more contact persons.

For each contact person this element shall contain:

1) one name;

2) one or more electronic addresses where to reach that contact person; and

3) one or more telephone numbers.

4.12.2 Syntax

The `ContactPersons` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="ContactPersons" type="ContactPersonsListType"/>
<xs:complexType name="ContactPersonsListType">
    <xs:sequence>
        <xs:element ref="ContactPerson" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="ContactPerson" type="ContactPersonType"/>
<xs:complexType name="ContactPersonType">
    <xs:sequence>
        <xs:element name="Name" type="xs:string"/>
        <xs:element ref="ElectronicAddresses"/>
        <xs:element name="PhoneNumbers" type="StringListType"/>
    </xs:sequence>
</xs:complexType>
<xs:simpleType name="StringListType">
    <xs:list itemType="xs:string"/>
</xs:simpleType>
```

Each string 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.

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

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

4.13 The `OtherDetails` element

4.13.1 Semantics

This element shall contain the date and time of issue of the signature policy.

This element may also contain:

1) The signing period, which identifies the date and time before which, and an optional date and time after which, the signature policy should not be used for creating new signatures.

NOTE: While this component impacts the creation and validation of a digital signature, it does not impact the augmentation of a digital signature, as this augmentation aims at preserving the status of the signature regardless when the signature was created.

2) A list of other general details, not defined in the present document.

4.13.2 Syntax

The `OtherDetails` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="OtherDetails" type="OtherDetailsType"/>
<xs:complexType name="OtherDetailsType">
    <xs:sequence>
        <xs:element name="DateOfIssue" type="xs:dateTime"/>
        <xs:element name="SigningPeriod" type="TimePeriodType" minOccurs="0"/>
        <xs:element name="Other" type="AnyType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="TimePeriodType">
    <xs:sequence>
        <xs:element name="NotBefore" type="xs:dateTime"/>
        <xs:element name="NotAfter" type="xs:dateTime" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
```

4.14 The `PolicyRules` element

4.14.1 Semantics

This element shall contain all the rules defined within the signature policy.

This element shall contain at least one of the following elements:

1) One element containing rules on commitments, corresponding to technical counterpart of BSP (g) in ETSI TS 119 172-1 [i.2]. This element shall be as specified in clause 4.15.

2) One element containing rules that apply to the data to be signed, corresponding to technical counterpart of BSP (b) in ETSI TS 119 172-1 [i.2]. This element shall be as specified in clause 4.16.

3) One element containing rules on the relationship between the signature and the data to be signed, corresponding to technical counterpart of BSP (c) in ETSI TS 119 172-1 [i.2]. This element t shall be as specified in clause 4.17.

4) One element containing rules about the responsibility of validating and/or augmenting the signature, corresponding to technical counterpart of BSP (e) in ETSI TS 119 172-1 [i.2]. This element shall be as specified in clause 4.21.

5) One element containing rules on the level of assurance required for timing evidence, corresponding to technical counterpart of BSP (h) in ETSI TS 119 172-1 [i.2]. This element shall be as specified in clause 4.25.

6) An element containing rules that apply to the identity and the attributes of the signer, corresponding to technical counterpart of BSP (l) in ETSI TS 119 172-1 [i.2]. This element shall be as specified in clause 4.24.

7) An element containing rules that apply to the required level of assurance on signer authentication, including trust conditions on X509 certificates, and trust conditions on revocation status information data objects, corresponding to technical counterpart of BSP (m) in ETSI TS 119 172-1 [i.2]. This element shall be as specified in clause 4.26.

8) An element containing rules on mandated signed and unsigned qualifying properties/attributes incorporated to the signature, corresponding to technical counterpart of BSP (o) in ETSI TS 119 172-1 [i.2]. This element shall be as specified in clause 4.27.

9) An element containing rules on the required level of assurance of the signature creation device, corresponding to technical counterpart of BSP (n) in ETSI TS 119 172-1 [i.2]. This element shall be as specified in clause 4.28.

10) An element containing rules on the cryptographic suites, corresponding to technical counterpart of BSP (p) in ETSI TS 119 172-1 [i.2]. This element shall be as specified in clause 4.29.

11) An element acting as placeholder for rules that have not been defined in ETSI TS 119 172-1 [i.2]. Its syntax is defined in clause 4.14.2.

NOTE: The set of rules defined in the present document builds a framework for defining policies, as users can select those rules that better fit the purposes of the signature policies that they are defining.

Each element in the bulleted list above may contain a component indicating a recommended scope for the rules contained in that element (recommended scope component hereinafter). The scope may be any combination of: generation, validation, and augmentation of digital signatures. Users of the signature policy should use the policy rule in the scopes identified in this component.

Absence of the recommended scope component shall mean that users of the signature policy can use the policy rules in any scope.

4.14.2 Syntax

The `PolicyRules` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="PolicyRules" type="PolicyRulesListType"/>
<xs:complexType name="PolicyRulesListType">
    <xs:sequence>
        <xs:element ref="PolicyRule" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:element name="PolicyRule" type="PolicyRuleType"/>

<xs:group name="BasicRule">
    <xs:choice>
        <xs:element ref="DataToBeSignedRules"/>
        <xs:element ref="SigToDTBSRelationRules"/>
        <xs:element ref="SigFormatsAndLevels"/>
        <xs:element ref="AugmentationRules"/>
        <xs:element ref="SigningCertRules"/>
        <xs:element ref="TimeEvidencesRules"/>
        <xs:element ref="SignerAttributesConstraints"/>
        <xs:element ref="QualifyingPropertiesRules"/>
        <xs:element ref="SCDLoARules"/>
        <xs:element ref="CryptoSuitesRules"/>
        <xs:element ref="OtherRule"/>
    </xs:choice>
</xs:group>
<xs:element name="OtherRule" type="AnyType"/>
<xs:complexType name="PolicyRuleType">
    <xs:choice>
        <xs:element ref="CommitmentRules"/>
```

```
            <xs:group ref="BasicRule"/>
        </xs:choice>
        <xs:attribute name="RecommendedScope" type="ScopeListType" use="optional"/>
    </xs:complexType>



    <xs:simpleType name="ScopeListType">
        <xs:list itemType="ScopeType"/>
    </xs:simpleType>

    <xs:simpleType name="ScopeType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Generation">
                <xs:annotation>
                    <xs:documentation>The rule applies for signature generation</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="Validation">
                <xs:annotation>
                    <xs:documentation>The rule applies for signature validation</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="Augmentation">
                <xs:annotation>
                    <xs:documentation>The rule applies for signature augmentation</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
        </xs:restriction>
    </xs:simpleType>
```

NOTE 1: A `PolicyRule` element can have as child either one of the elements referenced in `BasicRule` or the `CommitmentRules` element.

NOTE 2: The group of elements `BasicRule` groups all elements that can appear as children of the `CommitmentRules` element. It will also be referenced in the definition of the `CommitmentRulesType` in clause 4.15.2.

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

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

The element `SigFormatsAndLevels` shall be as specified in clause 4.20.2.

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

The element `SigningCertRules` shall be as specified in clause 4.24.2.

The element `TimeEvidencesRules` shall be as specified in clause 4.17.2.

The element `SignerAttributesConstraints` shall be as specified in clause 4.26.2.

The element `QualifyingPropertiesRules` shall be as specified in clause 4.27.2.

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

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

4.15 The `CommitmentRules` element

4.15.1 Semantics

This element shall contain a sequence of components. Each one shall contain:

1) A non-empty list of commitments identifiers.

2) On or more instance of the rules contained in the `PolicyRules` element as specified in clause 4.14, with the only exception that it shall not include the `CommitmentRules` child element.

3) A matching value indicator ("all", "atLeastOne", or "none") that identifies to which commitments the rules in the previous bullet apply.

4.15.2 Syntax

The `CommitmentRules` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
    <xs:element name="CommitmentRules" type="CommitmentRulesType"/>

    <xs:complexType name="CommitmentRulesType">
        <xs:sequence>
            <xs:element ref="ComitmentRulesComponent" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:element name="ComitmentRulesComponent" type="CommitmentRulesComponentType"/>

    <xs:complexType name="CommitmentRulesComponentType">
        <xs:sequence>
            <xs:element ref="Commitment" maxOccurs="unbounded"/>
            <xs:sequence>
                <xs:group ref="BasicRule" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:sequence>
        <xs:attribute name="MatchingIndicator" type="MatchingIndicatorType" use="required"/>
    </xs:complexType>

    <xs:element name="Commitment" type="CommitmentDetailsType"/>
    <xs:complexType name="CommitmentDetailsType">
        <xs:sequence>
            <xs:element name="Identifier" type="xs:anyURI" />
            <xs:element name="Details" type="tsl:MultiLangStringType" minOccurs="0"
maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>

    <xs:simpleType name="MatchingIndicatorType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="All"/>
            <xs:enumeration value="None"/>
            <xs:enumeration value="AtLeastOne"/>
        </xs:restriction>
    </xs:simpleType>
```

Each `Commitment` element shall contain an identifier of one commitment taken by the signer (`Identifier` child), and may also contain additional textual details in different languages (each `Detail` child element has the `lang` attribute for identifying them).

4.16 The `DataToBeSignedRules` element

4.16.1 Semantics

This element shall contain a sequence of 1 or 2 components. Each component:

1) Shall have as value a list of mime type values, identifying one or more formats for the signed data objects.

2) Shall be a list indicating either:

   - that none of the signed data objects shall have a mime type present in the list (NoneOf list); or

   - that all the signed data objects shall have one of the mime types present in the list (AnyOf list).

If this element contains a sequence of 2 components, one of them shall be a NoneOf list and the other shall be an AnyOf list as specified above.

4.16.2 Syntax

The `DataToBeSignedRules` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="DataToBeSignedRules" type="DataToBeSignedRulesType"/>

<xs:complexType name="DataToBeSignedRulesType">
    <xs:sequence>
        <xs:element name="AnyOfMimeType" type="StringListType" minOccurs="0"/>
        <xs:element name="NoneOfMimeType" type="StringListType" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
```

The `AnyOfMimeType` list shall include all the valid mime types for the signed data objects.

The `NoneOfMimeType` list shall include the invalid mime types for signed data objects.

The `DataToBeSignedRules` element shall not be empty.

4.17 The `SigToDTBSRelationRules` element

4.17.1 Semantics

This element shall contain:

1) An element providing information on the allowed number ranges for the data objects to be signed, as specified in clause 4.18.

This element may also contain:

1) A component indicating the relative position of the signature and the signed data object(s), as specified in clause 4.19.

2) A component indicating the format and the level of the signature as specified in clause 4.20.

4.17.2 Syntax

The `SigToDTBSRelationRules` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="SigToDTBSRelationRules" type="SigToDTBSRelationRulesType"/>

<xs:complexType name="SigToDTBSRelationRulesType">
    <xs:sequence>
        <xs:element ref="DTBSCardinality"/>
        <xs:element ref="SigDTBSRelativePosition" minOccurs="0"/>
        <xs:element ref="SigFormatsAndLevels" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
```

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

The element `SigToDTBSRelativePosition` shall be as specified in clause 4.19.2.

The element `SigToFormatsAndLevels` shall be as specified in clause 4.20.2.

4.18 The DTBSCardinality element

4.18.1 Semantics

This element shall contain at least one of the two following components:

1) One component whose value is an integer value, indicating a maximum value of data objects to be signed. This component shall also include a qualifier that shall take one of the following values: "LessThan", "LessOrEqualTo", or "Equal". If the qualifier has the value "LessThan", the number of data objects signed by the signature shall be less than the integer value in the component. If the qualifier has the value "LessOrEqualTo", the number of data objects signed by the signature shall be less than or equal to the integer value in the component. If the qualifier has the value "Equal", the number of data objects signed by the signature shall equal to the integer value in the component.

2) One component whose value is an integer value, indicating a minimum value of data objects to be signed. This component shall also include a qualifier that shall take one of the following values: "HigherThan", "HigherOrEqualTo", or "Equal". If the qualifier has the value "HigherThan", the number of data objects signed by the signature shall be higher than the integer value in the component. If the qualifier has the value "HigherOrEqualTo", the number of data objects signed by the signature shall be higher than or equal to the integer value in the component. If the qualifier has the value "Equal", the number of data objects signed by the signature shall equal to the integer value in the component.

4.18.2 Syntax

The DTBSCardinality element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
    <xs:element name="DTBSCardinality" type="CardinalityType"/>

    <xs:complexType name="CardinalityType">
        <xs:sequence>
            <xs:element name="MaximumValue" minOccurs="0">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension base="xs:int">
                            <xs:attribute name="qualifier" type="MaxValueQualEnumeratedType"
use="required"/>
                        </xs:extension>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
            <xs:element name="MinimumValue" minOccurs="0">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension base="xs:int">
                            <xs:attribute name="qualifier" type="MinValueQualEnumeratedType"
use="required"/>
                        </xs:extension>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>

    <xs:simpleType name="MaxValueQualEnumeratedType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="LessThan"/>
            <xs:enumeration value="LessOrEqualTo"/>
            <xs:enumeration value="Equal"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="MinValueQualEnumeratedType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="HigherThan"/>
            <xs:enumeration value="HigherOrEqualTo"/>
            <xs:enumeration value="Equal"/>
        </xs:restriction>
    </xs:simpleType>
```

4.19 The SigDTBSRelativePosition element

4.19.1 Semantics

This component shall indicate either:

a) that the signature shall be enclosed within an ASiC container (which also encloses the signed data objects); or

b) that the signature shall be enveloping, enveloped, or detached from the signed data objects; or

c) that the relative position between the signature and the signed data objects shall be any combination of two of enveloping, enveloped, and detached; or

d) that the relative position between the signature and the signed data objects shall be enveloping, enveloped, and detached.

NOTE: Cases c) and d) can only happen in XML and XAdES digital signatures (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).

Absence of this component shall mean that any relative position is valid for the signature policy.

4.19.2 Syntax

The DTBSCardinality element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="SigDTBSRelativePosition" type="SigDTBSRelativePositionType"/>

<xs:simpleType name="SigDTBSRelativePositionType">
    <xs:list itemType="RelativePositionValuesType" />
</xs:simpleType>

<xs:simpleType name="RelativePositionValuesType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="EnvelopingSig"/>
        <xs:enumeration value="EnvelopedSig"/>
        <xs:enumeration value="DetachedSig"/>
        <xs:enumeration value="ASiC"/>
    </xs:restriction>
</xs:simpleType>
```

The value of SigDTBSRelativePosition element shall be a list of one or more strings. Each string 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).

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.

If the value "ASiC" is not present in the list, this list shall contain 1, 2 or 3 string values.

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).

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

EXAMPLE: If the value of SigDTBSRelativePosition element is the string list "EnvelopingSig 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 element

4.20.1 Semantics

This component shall include:

1) One component for identifying the admitted signature formats, or for indicating that any format is admitted.

2) One component for indicating the admitted levels of the signature, or for indicating that any level is admitted.

EXAMPLE: ETSI TS 119 192 [i.4] provides URI values for identifying the different formats and the different levels.

4.20.2 Syntax

The SigFormatsAndLevels element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="SigFormatsAndLevels" type="SigFormatsAndLevelsType"/>

<xs:complexType name="SigFormatsAndLevelsType">
    <xs:sequence>
        <xs:element ref="SigFormats" minOccurs="0"/>
        <xs:element ref="SigLevels" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:element name="SigFormats" type="SigFormatsType"/>

<xs:complexType name="SigFormatsType">
    <xs:sequence>
        <xs:element name="Format" type="xs:anyURI" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:element name="SigLevels" type="SigLevelsType"/>

<xs:complexType name="SigLevelsType">
    <xs:sequence>
        <xs:element name="SigLevel" type="xs:anyURI" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

Instances of SigFormatsAndLevelsType type shall not be empty.

Absence of SigFormats child shall indicate that any format is allowed.

Absence of SigLevels child shall indicate that any level is allowed.

4.21 The AugmentationRules element

4.21.1 Semantics

This element shall contain:

1) A component indicating whether validation of the signature is required before proceeding to augment it.

2) A component identifying the constraints for the augmentation of the signature. This component shall include:

    a) The identifier of one level or an identifier indicating that there are no constraints on the augmentation levels.

    b) A qualifier indicating whether the indicated level is the exact level, a minimum level, or the maximal level the signature has to be augmented to.

4.21.2 Syntax

The `AugmentationRules` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="AugmentationRules" type="AugmentationRulesType"/>

<xs:complexType name="AugmentationRulesType">
    <xs:sequence>
        <xs:element name="PreviousValidationRequired" type="xs:boolean"/>
        <xs:element name="LevelId">
            <xs:complexType>
                <xs:simpleContent>
                    <xs:extension base="xs:anyURI">
                        <xs:attribute name="Qualifier">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:enumeration value="ThisLevel"/>
                                    <xs:enumeration value="MinimumLevel"/>
                                    <xs:enumeration value="MaximumLevel"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:attribute>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>
```

4.22 Types for defining constraints on certificates' trust

4.22.1 Introduction

The present clause defines four types:

1) `TrustAnchorsListType` which defines the trust anchors.

2) `NameConstraintsType` which defines constraints on the names of entities.

3) `PolicyConstraintsType` for defining constraints on certificate policies.

4) `CertificateTrustTreesType`, which defines constraints on the trust conditions required to certificates.

4.22.2 TrustAnchorsListType type

4.22.2.1 Semantics

Instances of this type shall contain a list of trust anchors. Each trust anchor may be provided in the different ways indicated below:

a) As a tuple formed by a self-signed X.509 certificate and an optional time instant indicating until when the trust anchor is reliable, its absence meaning that the trust anchor is reliable.

b) As a tuple that:

- Shall contain a reference to a Trusted List as specified in ETSI TS 119 612 [1], which may be either a specific Trusted List (TL) or a List of Trusted Lists (LOTL). A specific Trusted List lists trusted services. A List of Trusted Lists (LOTL) is a list that does not list trusted services but pointers to specific Trusted Lists.

- May contain an optional time instant indicating until when the trust anchors in the Trusted List are reliable, its absence meaning that the usage of these trust anchors are reliable as long as the Trusted List itself is reliable.

- May contain a list of identifiers of types of services.

- May contain the list of services statuses.

The set of trust anchors, in the case a tuple of case 2) is found, shall be determined as indicated below:

1) Take an initial set of certificates as follows:

   - If the referenced TL is a specific TL, the initial set of the certificates shall include all the certificates present in the TL.

   - If the referenced TL is a List of Trusted Lists, the initial set of the certificates shall include all the certificates listed in all the specific Trusted Lists pointed from the List of Trusted Lists.

2) If the list of identifiers of types of services is present, keep only those certificates in the initial list of certificates for which the corresponding service has a type contained in the list of identifiers of types of services mentioned in the third bullet in sub-list of bullet b) above.

EXAMPLE 1: ETSI TS 119 612 [1] defines as http://uri.etsi.org/TrstSvc/Svctype/CA/QC as the type for certificate authorities issuing EU qualified certificates.

3) If the list of identifiers of statuses is present, keep only those certificates for which the corresponding service has at current time a status contained in the list of services statuses mentioned in the fourth bullet in sub-list of bullet b) above.

EXAMPLE 2: ETSI TS 119 612 [1] defines http://uri.etsi.org/TrstSvc/TrustedList/Svcstatus/granted" the status for EU qualified service with a granted qualification status.

EXAMPLE 3: If the TL of Spain (https://sede.minetur.gob.es/Prestadores/TSL/TSL.xml) is referenced, then the initial set of certificates shall include all the certificates corresponding to the trusted services listed in the Spanish TL. If the European Union LOTL (https://ec.europa.eu/tools/lotl/eu-lotl.xml) is referenced then the initial set of certificates shall include all the certificates corresponding to the trusted services listed in any EUMS TL pointed in the EU LOTL. After that, certificates in the initial list are filtered out as per the lists mentioned respectively in third and fourth bullets in sub-list of bullet b) abovefor obtaining the list of trust anchors.

4.22.2.2 Syntax

The `TrustAnchors` element and the `TrustAnchorsType` type shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="TrustAnchors" type="TrustAnchorListType"/>

<xs:complexType name="TrustAnchorListType">
    <xs:sequence maxOccurs="unbounded">
        <xs:choice>
            <xs:element ref="X509CertificateBased"/>
            <xs:element ref="TAsInTrustedList"/>
        </xs:choice>
    </xs:sequence>
</xs:complexType>

<xs:element name ="X509CertificateBased" type="Base64BinaryAndTimeType"/>
<xs:complexType name="Base64BinaryAndTimeType">
    <xs:simpleContent>
        <xs:extension base="xs:base64Binary">
            <xs:attribute name="reliableUntil" type="xs:dateTime" use="optional"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

<xs:element name ="TAsInTrustedList" type="TrustAnchorsInTLType"/>
<xs:complexType name="TrustAnchorsInTLType">
    <xs:sequence>
        <xs:element name="TLReference" type="xs:anyURI"/>
        <xs:element name="ServiceTypes" type="tsl:NonEmptyURIListType" minOccurs="0"/>
        <xs:element name="ServiceStatuses" type="tsl:NonEmptyURIListType" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="reliableUntil" type="xs:dateTime" use="optional"/>
</xs:complexType>
```

4.22.3 NameConstraintsType type

4.22.3.1 Semantics

Instances of this type shall specify constraints on the names within the certificates (constraints on the subtrees).

Instances of this type shall define a name space within which all subject names in subsequent certificates in a certification path shall be located. The restrictions may apply either to the subject distinguished name or subject alternative names. The restrictions shall apply only when the specified name form is present. If no name of the type is present in the certificate, the certificate shall be acceptable.

Instances of this type shall have two sub-components, namely:

1) One subcomponent containing the list of permitted subtrees.

2) One subcomponent containing the list of excluded subtrees. Any name matching a restriction in this sub-component shall be considered invalid regardless of information appearing in the subcomponent that contains the list of permitted subtrees.

One names subtree shall be defined by:

- One instance of the type GeneralName type defined in IETF RFC 5280 [3] (the base of the subtree).
- One integer that shall indicate the minimum distance of the names in the subtree to the name acting as base. The default value of this integer shall be 0.
- One optional integer that shall indicate the maximum distance of the names in the subtree to the name acting as base.

4.22.3.2 Syntax

The `NameConstranitsType` type shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:complexType name="NameConstraintsType">
    <xs:sequence>
        <xs:element ref="PermittedSubtrees" minOccurs="0"/>
        <xs:element ref="ExcludedSubtrees" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<xs:element name="PermittedSubtrees" type="GeneralSubTreesListType" />
<xs:element name="ExcludedSubtrees" type="GeneralSubTreesListType" />

<xs:complexType name="GeneralSubTreesListType">
    <xs:sequence>
        <xs:element ref="GeneralSubTree" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:element name="GeneralSubTree" type="GeneralSubTreeType"/>

<xs:complexType name="GeneralSubTreeType">
    <xs:sequence>
        <xs:element name="Base" type="xs:base64Binary"/>
        <xs:element name="Minimum" type="xs:integer" default="0"/>
        <xs:element name="Maximum" type="xs:integer" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
```

The `Base` child element of GeneralSubTreeType shall contain the the base-64 encoding of one DER-encoded instance of type `GeneralName` type defined in IETF RFC 5280 [3].

4.22.4 PolicyConstraintsType type

4.22.4.1 Semantics

Instances of this type shall constrain path processing in two ways. They can be used to prohibit policy mapping, or to require that each certificate in a path contain an acceptable policy identifier.

Instances of this type shall contain at least one of the following two components:

1) One component for restricting policy mapping. It shall have an integer value, which shall indicate the number of additional certificates that may appear in the path (including the trustpoint's self-certificate) before policy mapping is no longer permitted.

EXAMPLE: A value of one indicates that policy mapping may be processed in certificates issued by the subject of this certificate, but not in additional certificates in the path.

2) One component for requiring the presence of an acceptable policy identifier within a set of certificates. It shall have an integer value, which shall indicate the number of additional certificates that may appear in the path (including the trustpoint's self-certificate) before an explicit policy is required.

An acceptable policy identifier shall be either the identifier of a policy required by the user of the certification path or the identifier of a policy that has been declared equivalent through policy mapping.

4.22.4.2 Syntax

The `PolicyConstranitsType` type shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:complexType name="PolicyConstraintsType">
    <xs:sequence>
         <xs:element name="RequireExplicitPolicy" type="xs:integer" minOccurs="0"/>
         <xs:element name="InhibitExplicitPolicy" type="xs:integer" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
```

4.22.5 CertificateTrustTreesType type

4.22.5.1 Semantics

Instances of this type shall contain an instance of `TrustAnchorsListType` as specified in clause 4.22.2.

Instances of this type may also contain the following components:

1) A component for specifying constraints on the certification path. Its value shall be an integer value greater than or equal to zero. If greater than zero, it shall indicate the maximum number of CA certificates that may be in a certification path following the trustpoint. A value of zero indicates that only the given trustpoint certificate and an end-entity certificate may be used. Absence of this component indicates that there is no limit to the allowed length of the certification path.

2) The initial set of acceptable certificate policies: a sequence of identifiers of certificate policies, each of which are acceptable under the signature policy.

3) One instance of `PolicyConstraintsType` type as specified in clause 4.22.4.

4) One instance of `NameConstraintsType` type as specified in clause 4.22.3.

5) One component that shall indicate the certification path that shall be used. This may be given explicitly, or implicitly, by means of an indication that the certification path provided in the signature shall be used. If this component is present, and its content contradicts requirements in any of the previous components listed in the bullets above, the requirements in the present component shall have preference.

4.22.5.2 Syntax

The `CertificateTrustTreesType` type shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
    <xs:complexType name="CertificateTrustTreesType">
        <xs:sequence>
             <xs:element ref="CertificateTrustPoint" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:element name="CertificateTrustPoint" type="CertificateTrustPointType"/>

    <xs:complexType name="CertificateTrustPointType">
        <xs:sequence>
            <xs:element ref="TrustAnchors"/>
            <xs:element name="PathLenConstraint" type="xs:integer" minOccurs="0"/>
            <xs:element name="AcceptablePolicySet" type="AcceptablePoliciesListType" minOccurs="0"/>
            <xs:element name="NameConstraints" type="NameConstraintsType" minOccurs="0"/>
            <xs:element name="PolicyConstraints" type="PolicyConstraintsType" minOccurs="0"/>
            <xs:element ref="UseCertPath" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="AcceptablePoliciesListType">
        <xs:sequence>
             <xs:element name="AcceptablePolicy" type="xs:anyURI" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:element name="UseCertPath">
        <xs:complexType>
            <xs:choice>
                <xs:element name="AsInSignature"/>
                <xs:sequence>
                    <xs:element name="X509Certificate" type="xs:base64Binary"
maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:choice>
        </xs:complexType>
    </xs:element>
```

The `TrustPoint` child element of `CertificateTrustPointType` and the `X509Certificate` child element of `UseCertPath` shall contain one DER-encoded X509 certificate conformant to IETF RFC 6960 [4].

The element `PolicyConstraints`, of type `PolicyConstraintsType`, 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) `CertificateRevReqType` which defines constraints on the certificate revocation checks procedures.

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

4.23.2 CertificateRevReqType type

4.23.2.1 Semantics

Instances of this type shall contain rules that specify requirements for the using certificate revocation status information (for instance CRLs, OCSP responses) to check the validity of a certificate.

The entity that validates the signature may take into account information in the certificate for deciding how best to check the revocation status but if the information in the certificate contradicts the requirements expressed in the present component, the requirements of the present document shall take precedence.

The rules specified in this component may apply to different types of certificates (e.g. the signing certificate, CA certificates, OCSP responders' certificates, CRLs issuers' certificates, Time Stamping Units' certificates, etc.).

Instances of this type shall have the following components:

1) One component for specifying requirements on the revocation checking procedures for end entity certificates.

2) One component for specifying requirements on the revocation checking procedures for CA certificates.

Each component shall indicate whether the revocation status of the certificate shall be verified or not, and, in case yes, whether the check shall take place using OCSP, CRL, both of them, either of them, or using other means.

Revocation requirements shall be specified in terms of:

- **clrcheck:** Checks shall be made against current CRLs (or authority revocation lists);
- **ocspcheck:** The revocation status shall be checked using the Online Certificate Status Protocol as specified in IETF RFC 6960 [4];
- **bothcheck:** Both OCSP and CRL checks shall be carried out;
- **eithercheck:** Either OCSP or CRL checks shall be carried out;
- **nocheck:** No check is mandated;
- **Other:** Other mechanism as defined by signature policy extension.

4.23.2.2 Syntax

The `CertificateRevReqType` type shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="CertificateRevReq" type="CertificateRevReqType"/>

<xs:complexType name="CertificateRevReqType">
    <xs:sequence>
         <xs:element name="EndRevReq" type="RevocationReqType"/>
         <xs:element name="CACerts" type="RevocationReqType"/>
    </xs:sequence>
</xs:complexType>

<xs:simpleType name="RevocationReqType">
    <xs:restriction base="xs:string">
         <xs:enumeration value="clrcheck"/>
             <xs:enumeration value="ocspcheck"/>
             <xs:enumeration value="bothcheck"/>
             <xs:enumeration value="eithercheck"/>
             <xs:enumeration value="nocheck"/>
             <xs:enumeration value="other"/>
    </xs:restriction>
</xs:simpleType>
```

4.23.3 CertificateRevTrustType type

4.23.3.1 Semantics

Instances of this type shall contain rules that specify requirements for using certificate revocation status information (for instance CRLs, OCSP responses) to check the validity of a certificate.

The entity that validates the signature may take into account information in the certificate for deciding how best to check the revocation status but if the information in the certificate contradicts the requirements expressed in the present component, the requirements of the present document shall take precedence.

The rules specified in this component may apply to different types of certificates (e.g. the signing certificate, CA certificates, OCSP responders' certificates, CRLs issuers' certificates, Time Stamping Units' certificates, etc.).

Instances of this type shall have an instance of type `CertificateRevReqType` as specified in clause 4.23.2.

Instances of this type may also have the following components:

1) A component defining a constraint for the freshness of the revocation status data that shall be used in the validation. This component shall either:

   - indicate the maximum accepted difference between the issuance date of the revocation status data of a certificate and the time of validation; or

   - require to accept only revocation status data issued a certain time after the best-signature time as computed in ETSI TS 119 102-1 [i.1].

For the special case when an instance of this type defines rules for certificate revocation status of the signing certificate used for validating the signature, it may also have the following component:

2) A component mandating 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 exceeding a lower bound indicated in this component.

4.23.3.2 Syntax

The `CertificateRevTrustType` type shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
    <xs:complexType name="CertificateRevTrustType">
        <xs:sequence>
            <xs:element ref="CertificateRevReq"/>
            <xs:element name="Freshness" minOccurs="0">
                <xs:complexType>
                    <xs:choice>
                        <xs:element name="MaxDifferenceRevocationAndValidation" type="xs:duration"/>
                        <xs:element name="TimeAfterSignature" type="xs:duration"/>
                    </xs:choice>
                </xs:complexType>
            </xs:element>
            <xs:element name="SigCertIssuedByCAKeepsExpiredRevokedCertsInfo" type="xs:duration"
minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
```

The value of `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` child shall be a time indication.

If the `SigCertIssuedByCAKeepsExpiredRevokedCertsInfo` child 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.

4.24 The `SigningCertRules` element

4.24.1 Semantics

The `SigningCertRules` element shall include a component defining the trust conditions for the signing certificate.

The `SigningCertRules` element may also contain a component specifying whether the signing certificate or all the certificates in the certification path shall be included in the signature.

4.24.2 Syntax

The `SigningCertRules` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="SigningCertRules" type="SigningCertRulesType"/>
<xs:complexType name="SigningCertRulesType">
    <xs:sequence>
        <xs:element ref="MandatedSigningCertInfo" minOccurs="0"/>
        <xs:element ref="SigningCertTrustConditions" />
    </xs:sequence>
</xs:complexType>
```

The element `MandatedSigningCertInfo` shall be as specified in clause 4.24.3.2.

The element `SigningCertTrustConditions` shall be as specified in clause 4.24.4.2.

4.24.3 The `MandatedSigningCertInfo` element

4.24.3.1 Semantics

The `MandatedSigningCertInfo` element shall indicate whether the signature shall include only the signing certificate, in which case its value shall be `"signingCertOnly"`, or the full signing certificate path, in which case its value shall be `"fullPath"`.

4.24.3.2 Syntax

The `MandatedSigningCertInfo` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="MandatedSigningCertInfo">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="signingCertOnly"/>
            <xs:enumeration value="fullPath"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

4.24.4 The `SigningCertTrustConditions` element

4.24.4.1 Semantics

This element shall contain:

1) a subcomponent instance of `CertificateTrustTreesType` type as specified in clause 4.22.5; and

2) a subcomponent instance of `CertificateRevReqType` type as specified in clause 4.23.3.

4.24.4.2 Syntax

The `SigningCertTrustConditions` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="SigningCertTrustConditions" type="SigningCertTrustConditionsType"/>
<xs:complexType name="SigningCertTrustConditionsType">
    <xs:sequence>
        <xs:element name="SignerTrustTrees" type="CertificateTrustTreesType"/>
        <xs:element name="SignerRevTrust" type="CertificateRevTrustType"/>
    </xs:sequence>
</xs:complexType>
```

The element `SignerTrustTrees`, of type `CertificateTrustTreesType`, shall be as specified in clause 4.22.5.2.

The element `SignerRevTrust`, of type `CertificateRevTrustType`, shall be as specified in clause 4.23.3.2.

4.25 The `TimeEvidencesRules` element

4.25.1 Semantics

This element shall include a sequence of tuples. Each tuple:

1) Shall have a component that identifies one or more specific time evidences.

2) Shall have a component indicating the Level of Assurance, as specified in ETSI TS 119 172-1 [i.2], on the time evidences identified in the first component of the tuple.

3) May also have a component identifying trust conditions for the certificate path processing used for authenticating the time-stamping authority(ies) that generate the time evidences, and constraints on its name. This component may contain:

   a) One indication of the maximum elapsed time period permitted between a time instant t0 and the signature time-stamp token generation. In signature creation t0 shall be the signature creation instant. In signature validation t0 shall be the claimed signing time, if present within the signature. As above this sub-component indicates a delta time with capability for indicating deltas for days, hours, minutes, and seconds.

   b) One instance of `CertificateTrustTreesType` type as specified in clause 4.22.5.

   c) One instance of `CertificateRevTrustType` type as specified in clause 4.23.3.

   d) One instance of `NameConstraintsType` type as specified in clause 4.22.3.

4.25.2 Syntax

The `TimeEvidencesRules` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
    <xs:element name="TimeEvidencesRules" type="TimeEvidencesRulesType"/>

    <xs:complexType name="TimeEvidencesRulesType">
        <xs:sequence>
            <xs:element ref="RulesForSetOfEvidences" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:element name="RulesForSetOfEvidences" type="RulesForSetOfEvidencesType"/>

    <xs:complexType name="RulesForSetOfEvidencesType">
        <xs:sequence>
            <xs:element name="EvidenceIdentifiers">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Identifier" type="xs:anyURI" maxOccurs="unbounded"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="LevelOfAssurance" type="xs:anyURI"/>
            <xs:element ref="TimeStampTrustCondition" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>

    <xs:element name="TimeStampTrustCondition" type="TimeStampTrustConditionType"/>

    <xs:complexType name="TimeStampTrustConditionType">
        <xs:sequence>
            <xs:element name="TstCertificateTrustTrees" type="CertificateTrustTreesType"
minOccurs="0"/>
            <xs:element name="TstRevocationTrust" type="CertificateRevTrustType" minOccurs="0"/>
            <xs:element name="TstNameConstraints" type="NameConstraintsType" minOccurs="0"/>
            <xs:element name="SignatureTimeStampDelay" type="DeltaTimeType" minOccurs="0"/>
        </xs:sequence>
```

```
    </xs:complexType>

    <xs:complexType name="DeltaTimeType">
        <xs:sequence>
            <xs:element name="DeltaSeconds" type="xs:integer"/>
            <xs:element name="DeltaMinutes" type="xs:integer"/>
            <xs:element name="DeltaHours" type="xs:integer"/>
            <xs:element name="DeltaDays" type="xs:integer"/>
        </xs:sequence>
    </xs:complexType>
```

The element `TstRevocationTrust`, of type `CertificateRevTrustType`, shall be as specified in clause 4.23.3.2.

The element `TstNameConstraints`, of type `NameConstraintsType`, shall be as specified in clause 4.22.3.2.

4.26 The `SignerAttributesConstraints` element

4.26.1 Semantics

This component defines constrains on the signer attributes signed qualifying property, if the signature incorporate such property.

If this component is not present, then any signer attribute enclosed within an attribute certificate or within a signed assertion shall be considered valid under the validation policy.

This component shall be a choice between:

1) an indication that signatures according to the policy shall not include signer attributes; and

2) a list of components defining constraints for signer attributes.

If the first component is present, signatures conformant to the signature policy shall not incorporate any signer attributes.

Each component in the aforementioned list:

1) Shall identify the level of assurance of the signer attribute, namely: claimed, certified, a signed assertion, or any of them.

2) May include, in the case the signer attribute is certified or a signed assertion, an instance of `CertificateTrustTreesType` type as specified in clause 4.22.5.

3) May include, in the case the signer attribute is certified or a signed assertion, an instance of `CertificateRevTrustType` type as specified in clause 4.23.2.

4) May include one or more components identifying one signer attribute and providing more specific constraints for it. More specifically each component:

   - Shall include a component identifying the mandated signer attribute ("signer attribute identifier" hereinafter).

   - May include a component defining constraints on the value of the signer attribute. If the "signer attribute identifier" component is not present, then this component shall not be present.

If the list of components defining constraints for the signer attributes is present, then signatures compliant with the signature policy shall incorporate the signer attributes identified in the "signer attribute identifier" components, and every signer attribute in the signature shall meet the requirements specified by the components in the numbered bulleted list above.

If this component does not have any "signer attribute identifier" component, then the requirements specified by the rest of the components in the numbered bulleted list above shall apply to any signer attribute incorporated to the signature.

4.26.2 Syntax

The `SignerAttributesConstraints` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
    <xs:element name="SignerAttributesConstraints" type="SignerAttributesConstraintsType"/>

    <xs:complexType name="SignerAttributesConstraintsType">
        <xs:choice>
            <xs:element name="NoSignerAttributesAllowed"/>
            <xs:element name="ConstraintsOnOneSetOfAttributes" type="AttributesSetConstraintsType"
maxOccurs="unbounded"/>
        </xs:choice>
    </xs:complexType>

    <xs:complexType name="AttributesSetConstraintsType">
        <xs:sequence>
            <xs:element name="HowCertAttribute" type="HowCertAttributeType"/>
            <xs:element name="AttrCertTrustTrees" type="CertificateTrustTreesType" minOccurs="0"/>
            <xs:element name="AttributeRevocationTrust" type="CertificateRevTrustType"
minOccurs="0"/>
            <xs:element name="AttributeConstraints" type="AttributeConstraintsType" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:simpleType name="HowCertAttributeType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="ClaimedAttribute"/>
            <xs:enumeration value="CertifiedAttribute"/>
            <xs:enumeration value="SignedAssertion"/>
            <xs:enumeration value="Any"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:complexType name="AttributeConstraintsType">
        <xs:sequence >
            <xs:element name="AttributeIdMustBePresent" type="xs:anyURI"/>
            <xs:element name="AttributeValueConstraint" type="AnyType" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
```

If the signer attribute type is identified by an OID, the value of the `AttributeIdeMustBePresent` child element shall be the OID value encoded as an URN as specified by the IETF RFC 3061 [2].

The element `AttrCertTrustTrees`, of type `CertificateTrustTreesType`, shall be as specified in clause 4.22.5.2.

The element `AttributeRevocationTrust`, of type `CertificateRevTrustType`, shall be as specified in clause 4.23.3.2.

4.27 The `QualifyingPropertiesRules` element

4.27.1 Semantics

This component shall include a sequence of tuple. Each tuple:

1) May include an identifier of a signature level, as specified in clause 4.20.2.

2) Shall include one component defining requirements for signed properties.

3) Shall include one component defining requirements for unsigned properties.

The components defining requirements for signed and unsigned properties:

1) Shall provide means for individually and completely identifying the qualifying property affected by the constraints, using an URI.

2) Shall provide means for indicating a choice between several properties.

3) Shall provide means for indicating the level of the presence requirement, namely: "mandatory" or "optional". Shall provide means for applying the level of presence requirement to both individual properties and choices among several properties.

Absence of the present component indicates that no requirements are imposed to qualifying properties, and signatures compliant with this signature policy may have signed/unsigned qualifying properties or not, and if they have all of them shall be considered valid against the policy.

EXAMPLE: ETSI TS 119 192 [i.4] defines strategies for building URIs identifying the signed and unsigned qualifying properties.

4.27.2 Syntax

The `QualifyingPropertiesRules` element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
    <xs:element name="QualifyingPropertiesRules" type="QualifyingPropertiesRulesType"/>

    <xs:complexType name="QualifyingPropertiesRulesType">
        <xs:sequence>
            <xs:element name="LevelRules" type="LevelQualifyingPropertiesRulesType"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="LevelQualifyingPropertiesRulesType">
        <xs:sequence>
            <xs:element name="LevelIdentifier" type="xs:anyURI" minOccurs="0"/>
            <xs:element name="SignedQualifyingProperties" type="QualifyingPropertiesListType"
minOccurs="0"/>
            <xs:element name="UnsignedQualifyingProperties" type="QualifyingPropertiesListType"
minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="QualifyingPropertiesListType">
        <xs:sequence maxOccurs="unbounded">
            <xs:choice>
                <xs:element name="Choice">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="PropertyId" type="xs:anyURI" maxOccurs="unbounded"/>
                        </xs:sequence>
                        <xs:attribute name="presenceLevel" type="PresenceLevelType" use="required"/>
                    </xs:complexType>
                </xs:element>
                <xs:sequence>
                    <xs:element name="PropertyId" maxOccurs="unbounded">
                        <xs:complexType>
                            <xs:simpleContent>
                                <xs:extension base="xs:anyURI">
                                    <xs:attribute name="presenceLevel" type="PresenceLevelType"
use="required"/>
                                </xs:extension>
                            </xs:simpleContent>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:choice>
        </xs:sequence>
    </xs:complexType>

    <xs:simpleType name="PresenceLevelType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Mandatory"/>
            <xs:enumeration value="Optional"/>
        </xs:restriction>
    </xs:simpleType>
```

4.28 The SCDLoARules element

4.28.1 Semantics

This component shall indicate the Level of Assurance of the signature creation device where resides the private key corresponding to the public key within the certificates validated during the certificate validation process.

4.28.2 Syntax

The SCDLoARules element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="SCDLoARules" type="xs:anyURI"/>
```

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

4.29 The CryptoSuitesRules element

4.29.1 Semantics

This component shall contain a non-empty sequence of components specifying algorithm constraints.

Each component:

1) Shall include an identifier of one algorithm.

2) May contain a list of usages where it shall be allowed to use the algorithm identified by the former component. Absence of this component shall imply that it shall be allowed to use the algorithm identified by the former component in all the possible usages. Below follow the list of possible usages:

   - Algorithms used for generating the digital signature (Signer).
   - Algorithms used for generating the signing certificate (SigningCert).
   - Algorithms used for generating certificates in certification paths.
   - Algorithms used for generating revocation status data.
   - Algorithms used for generating time-stamp tokens.
   - Algorithms used for generating attribute certificates.
   - Algorithms used for generating signed assertions.

3) Shall contain an indication of the expiration date.

4) May contain an indication of the minimum length of a key.

5) May contain an indication of the minimum length of a hash value.

6) May contain other data whose specification is out of the scope of the present document.

4.29.2 Syntax

The CryptoSuitesRules element shall be as defined in XML Schema file "19172xmlSchema.xsd", whose location is detailed in clause B.1, and is copied below for information.

```
<xs:element name="CryptoSuitesRules" type="CryptoSuiteRulesType"/>

<xs:complexType name="CryptoSuiteRulesType">
    <xs:sequence>
        <xs:element name="AlgConstraints" type="AlgConstraintsType" maxOccurs="unbounded"/>
```

```
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="AlgConstraintsType">
        <xs:sequence>
            <xs:element name="AlgId" type="xs:anyURI"/>
            <xs:element name="Usages" type="UsagesType" minOccurs="0"/>
            <xs:element name="ExpirationDate" type="xs:dateTime" />
            <xs:element name="MinKeyLength" type="xs:integer" minOccurs="0"/>
            <xs:element name="MinHashLength" type="xs:integer" minOccurs="0"/>
            <xs:element name="Other" type="AnyType" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <xs:simpleType name="UsagesType">
        <xs:list itemType="UsageType"/>
    </xs:simpleType>

    <xs:simpleType name="UsageType">
        <xs:restriction base="xs:anyURI">
            <xs:enumeration value="http://uri.etsi.org/19172/v1.1.1/Usage#Signature">
                <xs:annotation>
                    <xs:documentation>Constraints for algorithms used for generating the digital
signature</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="http://uri.etsi.org/19172/v1.1.1/Usage/#SigningCert">
                <xs:annotation>
                    <xs:documentation>Constraints for algorithms used for generating the signing
certificate</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="http://uri.etsi.org/19172/v1.1.1/Usage#PathCert">
                <xs:annotation>
                    <xs:documentation>Constraints for algorithms used for generating certificates in
certification paths</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="http://uri.etsi.org/19172/v1.1.1/Usage#RevStatData">
                <xs:annotation>
                    <xs:documentation>Constraints for algorithms used for generating revocation
status data</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="http://uri.etsi.org/19172/v1.1.1/Usage#Tstk">
                <xs:annotation>
                    <xs:documentation>Constraints for algorithms used for generating time-stamp
tokens</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="http://uri.etsi.org/19172/v1.1.1/Usage#AaCert">
                <xs:annotation>
                    <xs:documentation>Constraints for algorithms used for generating attribute
certificates</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="http://uri.etsi.org/19172/v1.1.1/Usage#SigAss">
                <xs:annotation>
                    <xs:documentation>Constraints for algorithms used for generating signed
assertions</xs:documentation>
                </xs:annotation>
            </xs:enumeration>
        </xs:restriction>
    </xs:simpleType>
```

Annex A (normative): URIs for identifying signature formats

A.1 URIs to signature formats mapping

Table A.1 lists the URIs used for identifying the digital signature formats.

Table A.1: Namespaces for identifying digital signature formats

URI	Format identified
http://uri.etsi.org/ades/format/cades	CAdES
http://uri.etsi.org/ades/format/pades	PAdES
http://uri.etsi.org/ades/format/xades	XAdES
http://uri.etsi.org/ades/format/asic	ASiC
http://uri.etsi.org/ades/format/any	Any format

Annex B (normative): XML Schema file

B.1 XML Schema file location for namespace http://uri.etsi.org/19172/v1.1.1#

The file at https://forge.etsi.org/rep/esi/x19_17202_XML_sign-policies/raw/v1.1.1/19172xmlSchema.xsd contains the definitions of qualifying properties defined within the namespace whose URI value is http://uri.etsi.org/19172/v1.1.1#.

History

Document history		
V1.1.1	December 2019	Publication