BUREAU OF INDIAN STANDARDS

DRAFT FOR COMMENTS ONLY

Draft Indian Standard

ELECTRONIC SIGNATURES AND INFRASTRUCTURES (ESI) — XAdES DIGITAL SIGNATURES — PART 1 BUILDING BLOCKS AND XAdES BASELINE SIGNATURES

ICS 35.020

Information Technology and Information Technology enabled Services Sectional Committee, SSD 10	Last date of comments : 10 May 2025

FOREWORD

(*Formal Clauses will be added later*)

This standard is developed for XAdES digital signatures and in three parts. Other parts in the series are:

Part 2 Extended XAdES signatures *(under development)*

Part 3 Incorporation of Evidence Record Syntax (ERS) mechanisms in XAdES *(under development)*

The Indian Standard is the technical adoption of the European Standard ETSI EN 319 132-1 v 1.2.1 ‘Electronic Signatures and Infrastructures (ESI) — XAdES digital signatures Part 1: Building blocks and XAdES baseline signatures’ developed by the European Telecommunication Standards Institute (ETSI). Modifications have been made to adapt it to India and are limited to referencing the Indian relevant regulatory context (*Information Technology Act,* 2000). The technical coverage is otherwise identical.

BUREAU OF INDIAN STANDARDS

DRAFT FOR COMMENTS ONLY
(Not to be reproduced without the permission of BIS or used as an Indian Standard)

Draft Indian Standard

ELECTRONIC SIGNATURES AND INFRASTRUCTURES (ESI) — XAdES DIGITAL SIGNATURES — PART 1 BUILDING BLOCKS AND XAdES BASELINE SIGNATURES

1 SCOPE

This standard specifies XAdES digital signatures. XAdES signatures build on XML digital signatures, by incorporation of signed and unsigned qualifying properties, which fulfil certain common requirements (such as the long term validity of digital signatures, for instance) in a number of use cases.

The standard specifies XML Schema definitions for the aforementioned qualifying properties as well as mechanisms for incorporating them into XAdES signatures.

The standard specifies formats for XAdES baseline signatures, which provide the basic features necessary for a wide range of business and governmental use cases for electronic procedures and communications to be applicable to a wide range of communities when there is a clear need for interoperability of digital signatures used in electronic documents.

The standard document defines four levels of XAdES baseline signatures addressing incremental requirements to maintain the validity of the signatures over the long term, in a way that a certain level always addresses all the requirements addressed at levels that are below it. Each level requires the presence of certain XAdES qualifying properties, suitably profiled for reducing the optionality as much as possible.

The standard aims at supporting electronic signatures in different regulatory frameworks.

2 REFERENCES

The standards listed in **Annex A** contain provisions, which through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of these standards.

3 DEFINITION OF TERMS, SYMBOLS AND ABBREVIATIONS

3.1 Terms and Definitions

For the purposes of this standard, the terms and definitions given in SSD 10 (27047) and the following apply.

3.1.1 *Attribute Certificate* — Data structure, digitally signed by an attribute authority, that binds some attribute values with identification information about its holder.

3.1.2 *Certificate Revocation List* **—** Signed list indicating a set of certificates that are no longer considered valid by the certificate issuer.

3.1.3 *Data Object* — Actual binary/octet data being operated on (transformed, digested, or signed) by an application.

NOTE **—** This definition of term is part of the definition of this term within XMLDSIG.

3.1.4 *Digital Signature* **—** Data appended to, or a cryptographic transformation of a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery for example by the recipient.

3.1.5 *Digital Signature Value* — Result of cryptographic transformation of a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery for example by the recipient.

3.1.6 *Electronic Time-Stamp* — Data in electronic form which binds other electronic data to a particular time establishing evidence that these data existed at that time.

NOTE **—** In the case of IETF RFC 3161 protocol, updated by IETF RFC 5816, the electronic time-stamp is referring to the timeStampToken field within the TimeStampResp element (the TSA's response returned to the requesting client).

3.1.7 *Void*

3.1.8 *Void*

3.1.9 *Void*

3.1.10 *Message Imprint* **—** Digest value of the data that is going to be time-stamped.

NOTE **—**In the case of electronic time-stamps compliant with IETF RFC 3161, as updated by IETF RFC 5816, it corresponds to the digest value incorporated into the hashedMessage field of MessageImprint type.

3.1.11 *Signature Augmentation Policy* **—** Set of rules, applicable to one or more digital signatures, that defines the technical and procedural requirements for their augmentation, in order to meet a particular business need, and under which the digital signature(s) can be determined to be conformant.

NOTE **—** This covers collection of information and creation of new structures that allows performing, on the long term, validations of a signature.

3.1.12 *Signature Creation Policy* **—** Set of rules, applicable to one or more digital signatures, that defines the technical and procedural requirements for their creation, in order to meet a particular business need, and under which the digital signature(s) can be determined to be conformant.

3.1.13 *Signature Policy* **—** Signature creation policy, signature augmentation policy, signature validation policy or any combination thereof, applicable to the same signature or set of signatures.

3.1.14 *Signature Validation Policy* **—** Set of rules, applicable to one or more digital signatures, that defines the technical and procedural requirements for their validation, in order to meet a particular business need, and under which the digital signature(s) can be determined to be valid.

3.1.15 *Trust Anchor* **—** Entity that is trusted by a relying party and used for validating certificates in certification paths.

3.1.16 *Validation Data* **—** Data that is used to validate a digital signature.

3.1.17 *XAdES Signature* **—** Digital signature that satisfies the requirements specified within the standard.

3.2 Symbols
Void.

3.3 Abbreviations

For the purposes of the standard, the abbreviations given in SSD 10 (27047) and the following apply.

Abbreviation	Description
ASN.1	Abstract Syntax Notation 1
BER	Basic Encoding Rules
CA	Certification Authority
CER	Canonical Encoding Rules
CRL	Certificate Revocation List
DER	Distinguished Encoding Rules
ERS	Evidence Record Syntax
HTTP	Hyper Text Transfer Protocol
MD5	Message-Digest Algorithm 5
MIME	Multipurpose Internet Mail Extensions
OCSP	Online Certificate Status Protocol
OID	Object Identifier
PER	Packed Encoding Rules
PI	Processing Instruction
PKI	Public Key Infrastructure
SAML	Security Assertion Markup Language
SIM	Subscriber Identity Module
SPO	Service Provision Option
TSA	Time Stamping Authorities
TSL	Trust-service Status List
TSP	Trusted Service Providers
TSU	Time-Stamping Unit
URI	Uniform Resource Identifier
UPL	Uniform Resource Locator
URN	Uniform Resource Name
UTC	Coordinated Universal Time
XER	XML Encoding Rules
XML	eXtensible Markup Language
XMLDSIG	eXtensible Markup Language Digital SIGnature
XSLT	eXtensible Stylesheet Language Transformations

3.4 Terminology

The standard uses the term ‘qualifying property’ for denoting an XML element that qualifies the signature, the signed data objects, or the signer.

The standard uses the term ‘element’ exclusively for denoting XML elements.

The standard defines new XML elements that are containers of qualifying properties (for instance `QualifyingProperties`, `SignedProperties`, or `UnsignedProperties`). The standard uses the terms ‘element’ or ‘container’ when refers to them.

The standard uses the term ‘attribute’ for denoting either XML attributes of XML elements or for denoting attributes owned by the signer (as in Error! Reference source not found.for instance). Consequently, a qualifying property, being an XML element, can have (XML) attributes.

The standard uses the term ‘child element’ exclusively in the context of XML content, for denoting an XML element that is a child element of another XML element.

The standard uses the term ‘XAdES components’ for denoting any XAdES signature's element, and any XAdES qualifying property incorporated into the XAdES signature.

4 GENERAL SYNTAX

4.1 General Requirements

XAdES signatures shall build on XMLDSIG as specified in W3C recommendation — XML signature syntax and processing version 1.1 by incorporation of XML W3C recommendation — extensible markup language (XML) 1.0 signed and unsigned qualifying properties. These qualifying properties shall be instances of XML types using the XML Schema syntax and structures specified in W3C recommendation Part 1 XML Schema and W3C Recommendation Part 2 XML Schema Part 2 Datatypes Second Edition.

The present clause defines the namespaces used in the aforementioned XML schema definitions.

The present clause also defines the types for the containers of the qualifying properties, and specifies the mechanisms for incorporating them into the XAdES signature.

4.2 XML Namespaces

The standard uses the URI namespaces listed below:

a) http://uri.etsi.org/01903/v1.3.2#

b) http://uri.etsi.org/01903/v1.4.1#

c) http://www.w3.org/2000/09/xmldsig#

d) http://www.w3.org/2001/XMLSchema

ETSI defines two XML Schema files for the standard, namely: XAdES01903v132-201601.xsd and XAdES01903v141-202107.xsd . See Annex D for details on their locations.

Table 1 shows two prefixes that refer to the same namespaces in the two XML Schema files. These prefixes are used throughout the standard to refer to specific elements in the XAdES signature.

Table 1 Namespaces with Constant Prefixes

(Clause 4.2)

XML Namespace URI (1)	Prefix (2)
http://www.w3.org/2000/09/xmldsig#	ds
http://www.w3.org/2001/XMLSchema	xsd

NOTE - The standard uses other prefixes in the excerpts of the XML Schema files for referencing XML elements. The preambles of the corresponding XML Schema files clearly identify the namespace corresponding to each prefix.

Below follows a copy of the `xsd:schema` element of the XML Schema file **Error! Reference source not found.**, whose location is detailed in D-1, and that defines the namespace whose URI is http://uri.etsi.org/01903/v1.3.2#.

```
<xsd:schema targetNamespace="http://uri.etsi.org/01903/v1.3.2#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns="http://uri.etsi.org/01903/v1.3.2#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
```

Below follows a copy of the xsd:schema element of the XML Schema file "XAdES01903v141-202107.xsd", whose location is detailed in D-2 of Annex D, and that defines the namespace whose URI is http://uri.etsi.org/01903/v1.4.1#.

```
<xsd:schema targetNamespace="http://uri.etsi.org/01903/v1.4.1#"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns="http://uri.etsi.org/01903/v1.4.1#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xades="http://uri.etsi.org/01903/v1.3.2#"
elementFormDefault="qualified">
```

NOTES

1 The http://uri.etsi.org/01903/v1.3.2# URI was defined by ETSI TS 101 903 (V1.3.2). Most of the XML elements and types used by XAdES signatures were defined in this namespace. The standard adds new types and elements to this namespace. Additionally, ETSI TS 101 903 (V1.4.1) defined http://uri.etsi.org/01903/v1.4.1# URI, where new types and elements were defined. The standard also adds new types and elements to this namespace.

2 The content of the XML Schema file "XAdES01903v141-202107.xsd", whose location is detailed in **The** file at https://uri.etsi.org/01903/v1.3.2/XAdES01903v132-201601.xsd (XAdES01903v132-201601.xsd) contains the definitions of qualifying properties defined within the namespace whose URI value is http://uri.etsi.org/01903/v1.3.2#.

3 **D-2** is different from the content of the XML Schema file defining types and elements in the namespace whose URI is http://uri.etsi.org/01903/v1.4.1#, in SSD 10 (27046).

In case of discrepancies between the xml schema excerpts provided in the standard and the XML Schema files, the XML Schema files shall take precedence.

4.3 The QualifyingProperties Container

4.3.1 *Semantics And Syntax*

a) Semantics

The `QualifyingProperties` element shall act as a container element for all the qualifying information that is added to an XML signature.

The qualifying properties shall be split into qualifying properties that are cryptographically bound to signed by the XML signature, and qualifying properties that are not cryptographically bound to not signed by the XML signature.

b) Syntax

The `QualifyingProperties` element shall be defined as in XML Schema file **Error! Reference source not found.**, whose location is detailed in D-1 and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="QualifyingProperties"
type="QualifyingPropertiesType"/>

<xsd:complexType name="QualifyingPropertiesType">
      <xsd:sequence>
            <xsd:element ref="SignedProperties" minOccurs="0"/>
            <xsd:element ref="UnsignedProperties" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="Target" type="xsd:anyURI" use="required"/>
      <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

The `Target` attribute shall refer to the Id attribute of the corresponding `ds:Signature`.

The value of `Target` attribute shall be a URI with a bare-name XPointer fragment. If the XAdES signature envelops the `QualifyingProperties` element, its not-fragment part shall be empty. Otherwise, its not-fragment part needs not be empty.

The `Id` attribute shall be used to reference the `QualifyingProperties` container.

A XAdES signature shall not incorporate empty `QualifyingProperties` elements.

4.3.2 *The SignedProperties Container*

a) Semantics

The `SignedProperties` element shall contain qualifying properties that are collectively signed by the XML signature. In consequence one of the `ds:Reference` children of `ds:SignedInfo`

element in the XAdES signature shall be generated in a way that ensures that the `SignedProperties` element contributes to the digital signature value computation.

The `SignedProperties` element may contain qualifying properties that qualify the XML signature itself, the signer, or some of the signed data objects.

b) Syntax

The `SignedProperties` element shall be defined as in XML Schema file **Error! Reference source not found.**, whose location is detailed in D-1 and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="SignedProperties" type="SignedPropertiesType" />

<xsd:complexType name="SignedPropertiesType">
      <xsd:sequence>
            <xsd:element ref="SignedSignatureProperties"  minOccurs="0"/>
            <xsd:element ref="SignedDataObjectProperties"  minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

The `SignedSignatureProperties` element shall contain qualifying properties that qualify the XML signature itself or the signer. This element is specified in **4.3.4**.

The `SignedDataObjectProperties` element shall contain qualifying properties that qualify some of the signed data objects. This element is specified in **4.3.5**.

The Id attribute shall be used to reference the `SignedProperties` element.

A XAdES signature shall not incorporate empty `SignedProperties` element.

4.3.3 *The UnsignedProperties Container*

a) Semantics

The `UnsignedProperties` element shall contain qualifying properties that are not signed by the XML signature.

The `UnsignedProperties` element may contain qualifying properties that qualify the XML signature itself, the signer, or some of the signed data objects.

b) Syntax

The `UnsignedProperties` element shall be defined as in XML Schema file **Error! Reference source not found.**, whose location is detailed in D-1 of Annex D, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="UnsignedProperties" type="UnsignedPropertiesType" />
```

```
<xsd:complexType name="UnsignedPropertiesType">
      <xsd:sequence>
            <xsd:element ref="UnsignedSignatureProperties"
minOccurs="0"/>
            <xsd:element ref="UnsignedDataObjectProperties"
minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

The `UnsignedSignatureProperties` element shall contain qualifying properties that qualify the XML signature itself or the signer. This element is specified in **4.3.6**.

The `UnsignedDataObjectProperties` element shall contain qualifying properties that qualify some of the signed data objects. This element is specified in **4.3.7**.

The `Id` attribute shall be used to reference the `UnsignedProperties` element.

A XAdES signature shall not incorporate empty `UnsignedProperties` elements.

4.3.4 *The SignedSignatureProperties Container*

a) Semantics

This element shall contain signed qualifying properties that qualify the XML signature.

b) Syntax

The `SignedSignatureProperties` element shall be defined as in XML Schema file XAdES01903v132-201601.xsd whose location is detailed in D-1, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#"

The preamble of the XML Schema file also includes the following namespace
declaration:
  xmlns:xadestsv132="http://uri.etsi.org/01903/v1.3.2#",
which assigns the prefix "xadestsv132" to the namespace whose URI is shown
in the declaration.
-->

<xsd:element name="SignedSignatureProperties"
  type="SignedSignaturePropertiesType" />

<xsd:element                                name="SignedSignatureProperties"
type="SignedSignaturePropertiesType"/>

<xsd:complexType name="SignedSignaturePropertiesType">
      <xsd:sequence>
            <xsd:element ref="SigningTime" minOccurs="0"/>
            <xsd:element ref="SigningCertificate" minOccurs="0"/>
            <xsd:element ref="SigningCertificateV2" minOccurs="0"/>
            <xsd:element ref="SignaturePolicyIdentifier" minOccurs="0"/>
```

```
            <xsd:element ref="SignatureProductionPlace"  minOccurs="0"/>
            <xsd:element ref="SignatureProductionPlaceV2" minOccurs="0"/>
            <xsd:element ref="SignerRole"  minOccurs="0"/>
            <xsd:element ref="SignerRoleV2" minOccurs="0"/>
            <xsd:any           namespace="##other"           minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

Future versions of this multi-part deliverable may use the any element for allowing the incorporation of additional signed signature qualifying properties. These additional signed signature qualifying properties shall be defined in a namespace whose URI is different from http://uri.etsi.org/01903/v1.3.2#.

The `SignedSignatureProperties` element shall not incorporate any elements as an instantiation of `xsd:any` that are not specified within any version of this multi-part deliverable.

The `Id` attribute shall be used to reference the `SignedSignatureProperties` element.

Qualifying properties `SigningCertificate`, `SignatureProductionPlace`, and `SignerRole` are obsoleted by `SigningCertificateV2`, `SignatureProductionPlaceV2`, and `SignerRoleV2` respectively (see **5.2.2, 5.2.5** and **5.2.6** respectively).

The aforementioned obsoleted qualifying properties shall not be incorporated into the signature.

A XAdES signature shall not incorporate an empty `SignedSignatureProperties` element.

4.3.5 *The SignedDataObjectProperties Container*

a) Semantics

This element shall contain signed qualifying properties that qualify some of the signed data objects.

b) Syntax

The `SignedDataObjectProperties` element shall be defined as in XML Schema file **Error! Reference source not found.** , whose location is detailed in D-1 of Annex D, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:complexType name="SignedDataObjectPropertiesType">
      <xsd:sequence>
            <xsd:element ref="DataObjectFormat" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element ref="CommitmentTypeIndication" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element ref="AllDataObjectsTimeStamp" minOccurs="0"
maxOccurs="unbounded"/>
            <xsd:element ref="IndividualDataObjectsTimeStamp"
minOccurs="0" maxOccurs="unbounded"/>
```

```
            <xsd:any namespace="##other" minOccurs="0"
maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

Future versions of this multi-part deliverable may use the any element for allowing the incorporation of additional signed data objects qualifying properties. These additional signed data objects qualifying properties shall be defined in a namespace whose URI is different from http://uri.etsi.org/01903/v1.3.2#.

The SignedDataObjectProperties element shall not incorporate any element as an instantiation of `xsd:any` that are not specified within any version of this multi-part deliverable.

The `Id` attribute shall be used to reference the `SignedDataObjectProperties` element.

A XAdES signature shall not incorporate an empty `SignedDataObjectProperties` element.

4.3.6 *The UnsignedSignatureProperties Container*

Semantics

This element shall contain unsigned qualifying properties that qualify the XML.

The XML signature shall not cover the content of this element.

Syntax

The `UnsignedSignatureProperties` element shall be defined as in XML Schema file **Error! Reference source not found.**, whose location is detailed in D-1 of Annex D and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="UnsignedSignatureProperties"
  type="UnsignedSignaturePropertiesType"/>

<xsd:complexType name="UnsignedSignaturePropertiesType">
      <xsd:choice maxOccurs="unbounded">
            <xsd:element ref="CounterSignature" />
            <xsd:element ref="SignatureTimeStamp" />
            <xsd:element ref="CompleteCertificateRefs"/>
            <xsd:element ref="CompleteRevocationRefs"/>
            <xsd:element ref="AttributeCertificateRefs"/>
            <xsd:element ref="AttributeRevocationRefs" />
            <xsd:element ref="SigAndRefsTimeStamp" />
            <xsd:element ref="RefsOnlyTimeStamp" />
            <xsd:element ref="CertificateValues" />
            <xsd:element ref="RevocationValues"/>
            <xsd:element ref="AttrAuthoritiesCertValues" />
            <xsd:element ref="AttributeRevocationValues"/>
            <xsd:element ref="ArchiveTimeStamp" />
            <xsd:any namespace="##other"/>
```

```
      </xsd:choice>
      <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

Future versions of this multi-part deliverable may use the any element for allowing the incorporation of additional signed data objects qualifying properties. These additional signed data objects qualifying properties shall be defined in a namespace whose URI is different from http://uri.etsi.org/01903/v1.3.2#.

The `UnsignedSignatureProperties` element shall not incorporate any element as an instantiation of `xsd:any` that are not specified within any version of this multi-part deliverable.

The `Id` attribute shall be used to reference the `UnsignedSignatureProperties` element.

Qualifying properties `CompleteCertificateRefs`, `AttributeCertificateRefs`, `SigAndRefsTimeStamp`, and `RefsOnlyTimeStamp`, defined in the namespace whose URI value http://uri.etsi.org/01903/v1.3.2# (XML Schema file **Error! Reference source not found.**) are obsoleted by `CompleteCertificateRefsV2`, `AttributeCertificateRefsV2`, `SigAndRefsTimeStampV2`, `RefsOnlyTimeStampV2` (defined in the namespace whose URI value is http://uri.etsi.org/01903/v1.4.1#) respectively (see B-1.1, B-1.3, B-1.5.1 and B-1.5.2 respectively).

Additionally the qualifying property `ArchiveTimeStamp` defined in the namespace whose URI value http://uri.etsi.org/01903/v1.3.2# (XML Schema file "**Error! Reference source not found.**") is obsoleted by `ArchiveTimeStamp` defined in the namespace whose URI value is http://uri.etsi.org/01903/v1.4.1# (XML Schema file "XAdES01903v141-202107.xsd", see **5.5.2**).

The aforementioned obsoleted qualifying properties shall not be incorporated into the XAdES signature.

A XAdES signature shall not incorporate an empty `UnsignedSignatureProperties` element.

4.3.7 *The UnsignedDataObjectProperties Container*

Semantics

This element shall contain qualifying properties that qualify some of the signed data objects.

The XML signature shall not cover the content of this element.

Syntax

The `UnsignedDataObjectProperties` element shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in D-1 of Annex D, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="UnsignedDataObjectProperties"
type="UnsignedDataObjectPropertiesType"/>
```

```
<xsd:complexType name="UnsignedDataObjectPropertiesType">
      <xsd:sequence>
            <xsd:element name="UnsignedDataObjectProperty" type="AnyType"
maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

The `Id` attribute shall be used to reference the `UnsignedDataObjectProperties` element.

A XAdES signature shall not incorporate empty `UnsignedDataObjectProperties` element.

> NOTE — The standard does not specify the usage of any unsigned qualifying property qualifying the signed data objects. It, however, defines this element for the sake of completeness and to cope with potential future needs for inclusion of such kind of qualifying properties. The schema definition leaves open the definition of the contents of this type. The type AnyType is defined in **5.1.1**Error! Reference source not found..

4.4 Incorporating Qualifying Properties into XAdES Signatures

4.4.1 *General Requirements*

The `ds:Object` auxiliary element from XMLDSIG shall be used for incorporating the qualifying properties into the XAdES signature.

The standard specifies two different means for incorporating qualifying properties:

a) Direct incorporation means that a `QualifyingProperties` element shall be a child of the `ds:Object`;
b) Indirect incorporation means that one or more `QualifyingPropertiesReference` elements shall appear as children of the `ds:Object`. Each one shall contain information about one `QualifyingProperties` element that shall not be a descendant element of the `ds:Signature` XAdES signature root element (see **4.4.3**).

The following restrictions apply for using `ds:Object`, `QualifyingProperties` and `QualifyingPropertiesReference`:

a) All instances of `QualifyingProperties` directly incorporated into the XAdES signature, and all the instances of `QualifyingPropertiesReference`, shall occur within a single `ds:Object` element;
b) At most one instance of the QualifyingProperties element may occur within this `ds:Object` element;
c) All signed qualifying properties shall occur within a single `QualifyingProperties` element. This element shall either be a child of this `ds:Object` element (direct incorporation), or referenced by a `QualifyingPropertiesReference` element (see **4.4.2** for information how to sign qualifying properties); and
d) Zero or more instances of the `QualifyingPropertiesReference` element may occur within this `ds:Object` element.

XAdES signatures may contain `ds:Object` elements different from the `ds:Object` elements containing the `QualifyingProperties` or `QualifyingPropertiesReference` elements.

No restrictions apply to the relative position of the `ds:Object` containing the `QualifyingProperties` or `QualifyingPropertiesReference` with respect to other `ds:Object` elements present within `ds:Signature`.

> NOTE — It is out of the scope of the standard to specify the mechanisms required to guarantee the correct storage of the distributed `QualifyingProperties` elements (i.e. that the qualifying properties are stored by the entity that has to store them and that they are not undetectably modified).

4.4.2 *Signing Properties*

All the signed qualifying properties shall be children of the `SignedProperties` child of the `QualifyingProperties` element.

In order to protect the qualifying properties with the signature, a `ds:Reference` element shall be added to the XML signature.

This `ds:Reference` element shall be composed in such a way that it uses the `SignedProperties` element mentioned above as the input for computing its corresponding digest.

This `ds:Reference` element shall include the Type attribute with its value set to:

http://uri.etsi.org/01903#SignedProperties.

> NOTE — This value indicates that the data used for digest computation is a `SignedProperties` element and therefore helps to detect the signed qualifying properties of a XAdES signature conforming to the standard.

4.4.3 *The QualifyingPropertiesReference Element*

Semantics

This element shall contain information about one `QualifyingProperties` element that is not descendant of the `ds:Signature` XAdES signature root element (if for instance, it is stored in another XML document).

Syntax

The `QualifyingPropertiesReference` element shall be defined as in XML Schema file **Error! Reference source not found.**, whose location is detailed in D-1 of Annex D and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="QualifyingPropertiesReference"
  type="QualifyingPropertiesReferenceType"/>

<xsd:complexType name="QualifyingPropertiesReferenceType">
  <xsd:attribute name="URI" type="xsd:anyURI" use="required"/>
```

```
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

The URI attribute shall contain a bare-name XPointer fragment and shall reference an external `QualifyingProperties` element. Its not-fragment part shall identify the enclosing document and its bare-name XPointer fragment shall identify the aforementioned element.

The `Id` attribute shall be used to reference the `QualifyingPropertiesReference` element.

4.5 Managing Canonicalization of XML Nodesets

A number of qualifying properties specified in the standard incorporate optional means for identifying a canonicalization algorithm for computing the canonical form of a certain XML node set.

When generating new XAdES signatures, all the XAdES qualifying properties that provide optional means for indicating the canonicalization algorithm shall include the canonicalization algorithm identifier.

When augmenting a legacy XAdES signature by the generation and incorporation of a certain XAdES qualifying property specified in the standard, and whose XML Schema definition includes an optional identifier of a canonicalization algorithm, this qualifying property shall include the canonicalization algorithm identifier.

NOTES

1 Canonical XML 1.0 does not properly process the inheritance of attributes in the XML namespace (`xml:id` and `xml:base`) when canonicalizing document sub-trees. Canonical XML version 1.1 (whose version omitting comments is identified by the URI http://www.w3.org/2006/12/xml-c14n11), specifies a variant of the former canonicalization algorithm that properly addresses these issues.

2 Canonical XML 1.0 when applied to a XML sub-tree, includes the sub-tree's ancestor context including all of the namespace declarations and attributes in the "xml:" namespace. The exclusive XML Canonicalization algorithm (whose version omitting comments is identified by the URI http://www.w3.org/2001/10/xml-exc-c14n#) completely excludes this ancestor context from the canonicalized sub-tree.

5 QUALIFYING PROPERTIES SEMANTICS AND SYSTAX

5.1 Auxiliary Syntax

5.1.1 *The AnyType Data Type*

Semantics

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

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

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

Syntax

The `AnyType` type shall be defined as in XML Schema file **Error! Reference source not found.**, whose location is detailed in D-1 of Annex D and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="Any" type="AnyType"/>

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

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

5.1.2 *The ObjectIdentifierType Data Type*

Semantics

Instances of O`bjectIdentifierType` data type shall contain a unique and permanent identifier of one data object.

Instances of `ObjectIdentifierTyp`e data type may contain a textual description of the nature of the data object qualified by the instance of the ObjectIdentifierType data type.

Instances of `ObjectIdentifierType` data type may contain a number of references to documents where additional information about the nature of the data object qualified by the instance of the `ObjectIdentifierType` data type, can be found.

Syntax

The `ObjectIdentifierType` shall be defined as in XML Schema file **Error! Reference source not found.**, whose location is detailed in D-1 of Annex D and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="ObjectIdentifier" type="ObjectIdentifierType"/>

<xsd:complexType name="ObjectIdentifierType">
  <xsd:sequence>
    <xsd:element name="Identifier" type="IdentifierType"/>
    <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DocumentationReferences"
     type="DocumentationReferencesType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="IdentifierType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:anyURI">
      <xsd:attribute name="Qualifier" type="QualifierType"
        use="optional"/>
    </xsd:extension>
```

```
  </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="QualifierType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="OIDAsURI"/>
    <xsd:enumeration value="OIDAsURN"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="DocumentationReferencesType">
  <xsd:sequence maxOccurs="unbounded">
    <xsd:element name="DocumentationReference" type="xsd:anyURI"/>
  </xsd:sequence>
</xsd:complexType>
```

The `Identifier` element shall contain a permanent identifier. Once the identifier is assigned, it shall not be re-assigned again.

The `Identifier` element supports two mechanisms for identifying objects:

a) If a URI identifies the object, then the value of the `Identifier` element shall be this URI and the `Qualifier` attribute shall not be present; or
b) If an Object Identifier (OID) identifies the object, then the value of the `Identifier` element shall be the OID value encoded either as a Uniform Resource Name (URN), or as URI that is not a URN. The following rules shall apply in this case:

    1) If the OID is encoded as a URN, then:
        i) The `Qualifier` attribute shall be present and shall have the value "`OIDASURN`"; and
        ii) The OID shall be encoded as an URN as specified by the IETF RFC 3061; or

    2) If the OID is encoded as a URI that is not a URN, then the `Qualifier` attribute shall be present and shall have the value "`OIDAsURI`".

If both an OID and a URI exist identifying one object, the URI value should be used in the `Identifier` element.

The `Description` element shall contain an informal text describing the object.

The `DocumentationReferences` element shall contain an arbitrary number of references pointing to further explanatory documentation of the data object.

5.1.3 *The EncapsulatedPKIDataType Data Type*

Semantics

The `EncapsulatedPKIDataType` shall be used to incorporate PKI objects, which can be non-XML encoded, into the XAdES signature.

> NOTE ~~1~~— Examples of such PKI objects, include X.509 certificates and revocation lists, OCSP responses, attribute certificates, and electronic time-stamps.

Syntax

The `EncapsulatedPKIDataType` type shall be defined as in XML Schema file "**Error! Reference source not found.** whose location is detailed in Annex D, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="EncapsulatedPKIData" type="EncapsulatedPKIDataType"/>


<xsd:complexType name="EncapsulatedPKIDataType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:base64Binary">
      <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
      <xsd:attribute name="Encoding" type="xsd:anyURI"
        use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

The content of this data type shall be the PKI object, base-64 encoded.

The `Encoding` attribute value shall be a URI identifying the encoding used in the original PKI object. The following URIs shall be used:

a) http://uri.etsi.org/01903/v1.2.2#DER for denoting that the original PKI data were ASN.1 data encoded in DER;
b) http://uri.etsi.org/01903/v1.2.2#BER for denoting that the original PKI data were ASN.1 data encoded in BER;
c) http://uri.etsi.org/01903/v1.2.2#CER for denoting that the original PKI data were ASN.1 data encoded in CER;
d) http://uri.etsi.org/01903/v1.2.2#PER for denoting that the original PKI data were ASN.1 data encoded in PER; or
e) http://uri.etsi.org/01903/v1.2.2#XER for denoting that the original PKI data were ASN.1 data encoded in XER.

If the `Encoding` attribute is not present, then the PKI data shall be ASN.1 data encoded in DER.

NOTE 2 — In some clauses of the standard, specific XAdES qualifying properties related to these data restrict the encoding options to only one certain type of the aforementioned PKI data.

The `Id` attribute shall be used to reference an element of this data type.

5.1.4 *Types For Electronic Time-Stamps Management*

5.1.4.1 *Semantics*

The standard specifies qualifying properties that act as electronic time-stamps containers.

Electronic time-stamps within the aforementioned containers may time-stamp elements defined in XMLDSIG and/or qualifying properties specified in the standard, and/or detached signed data objects.

NOTE — The standard specifies:

a) An XML schema definition of an abstract base type and two concrete derived types used as containers for electronic time-stamps; and

b) A number of qualifying properties of one of the aforementioned concrete types.

5.1.4.2 *Containers for electronic time-stamps*

Below follows the list of the electronic time-stamps container qualifying properties that are defined by the standard:

a) Containers for electronic time-stamps proving that some or all the signed data objects have been created before certain time instant: `AllDataObjectsTimeStamp` and `IndividualDataObjectsTimeStamp`;
b) Container for electronic time-stamps proving that the `SignatureValue` element has been created before a certain time instant (to protect against repudiation in case of a key compromise): `SignatureTimeStamp;`
c) Container for electronic time-stamps time-stamping the signature and validation data values, for providing long term XAdES signatures: ArchiveTimeStamp qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.4.1#; and
d) Annex B specifies two qualifying properties that contain electronic time-stamps on qualifying properties that contain references to validation data, namely: `SigAndRefsTimeStampV2` and `RefsOnlyTimeStampV2.`

5.1.4.3 *The GenericTimeStampType data type*

Semantics

The `GenericTimeStampType` type shall:

a) Allow encapsulating IETF RFC 3161 updated by IETF RFC 5816 electronic time-stamps, which shall be instances of TimeStampToken type specified in **2.4.2** of IETF RFC 3161;
b) Allow encapsulating XML electronic time-stamps;
c) Allow encapsulating other formats of electronic time-stamps;
d) Allow encapsulating more than one electronic time-stamp generated for the same set of data objects (each one issued by different TSAs, for instance);
e) Provide means for managing electronic time-stamps computed on XAdES components, electronic time-stamps computed on XAdES components and detached signed data objects, or electronic time-stamps computed on external data; and
f) Specify mechanisms for explicitly identifying what is time-stamped and how to generate the input data for the computation of the message imprint to be sent to the TSA.

Syntax

The `GenericTimeStampType` type shall be the abstract base type defined as in XML Schema file "**Error! Reference source not found.** whose location is detailed in D-1 of Annex D, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="Include" type="IncludeType"/>

<xsd:complexType name="IncludeType">
<xsd:attribute name="URI" type="xsd:anyURI" use="required"/>
```

```
<xsd:attribute name="referencedData" type="xsd:boolean"
use="optional"/>
</xsd:complexType>

<xsd:element name="ReferenceInfo" type="ReferenceInfoType"/>

<xsd:complexType name="ReferenceInfoType">
  <xsd:sequence>
    <xsd:element ref="ds:DigestMethod"/>
    <xsd:element ref="ds:DigestValue"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
  <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
</xsd:complexType>

<xsd:complexType name="GenericTimeStampType" abstract="true">
  <xsd:sequence>
    <xsd:choice minOccurs="0">
      <xsd:element ref="Include" minOccurs="0"
maxOccurs="unbounded"/>
      <xsd:element ref="ReferenceInfo" maxOccurs="unbounded"/>
    </xsd:choice>
    <xsd:element ref="ds:CanonicalizationMethod" minOccurs="0"/>
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="EncapsulatedTimeStamp"
        type="EncapsulatedPKIDataType"/>
      <xsd:element name="XMLTimeStamp" type="AnyType"/>
    </xsd:choice>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

The `ds:CanonicalizationMethod` element shall indicate the canonicalization algorithm used for canonicalizing XML node sets resulting after retrieving (and processing when required) the data objects time-stamped by the electronic time-stamp(s).

4.5 shall apply when dealing with the `ds:CanonicalizationMethod` element.

If the electronic time-stamp generated by the TSA is conformant to IETF RFC 3161 as updated by IETF RFC 5816, it shall be included within the `EncapsulatedTimeStamp` element. If the electronic time-stamp generated by the TSA is encoded as XML then it shall be included within XMLTimeStamp element.

Details on the different elements and supporting types are given in the clauses that define the two concrete types: `XAdESTimeStampType and` for `OtherTimeStampType.`

5.1.4.4 *The XAdESTimeStampType data type*

5.1.4.4.1*Semantics and syntax*

Semantics

Instances of `XAdESTimeStampType` type shall be used for incorporating electronic time-stamps on XAdES components, or electronic time-stamps on XAdES components and detached signed data objects, into XAdES signatures.

Syntax

The `XAdESTimeStampType` type shall be defined as in XML Schema file **Error! Reference source not found.**", whose location is detailed in D-1 of Annex D, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="XAdESTimeStamp" type="XAdESTimeStampType"/>

<xsd:complexType name="XAdESTimeStampType">
  <xsd:complexContent>
    <xsd:restriction base="GenericTimeStampType">
      <xsd:sequence>
        <xsd:element          ref="Include"          minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element               ref="ds:CanonicalizationMethod"
minOccurs="0"/>
        <xsd:choice maxOccurs="unbounded">
          <xsd:element                  name="EncapsulatedTimeStamp"
type="EncapsulatedPKIDataType"/>
          <xsd:element name="XMLTimeStamp" type="AnyType"/>
        </xsd:choice>
      </xsd:sequence>
      <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
```

This type provides two mechanisms for identifying data objects that are time-stamped by the electronic time-stamp present in the container, and for specifying how to compute the electronic time-stamp's message imprint:

a) Explicit. This mechanism shall use the Include element for referencing specific data objects and for indicating their contribution to the input of the message imprint's computation; or
b) Implicit. For certain time-stamp container qualifying properties under certain circumstances, no explicit indications are required for knowing what data objects are time-stamped by the electronic time-stamps and how they contribute to the input of the message imprint's computation. The standard specifies, in the clauses defining such qualifying properties (**5.2.8.1**, **5.3**, **5.5.2.2**, **A-1.5.1.2** and **A-1.5.2.2**), what data objects are time-stamped by the electronic time-stamps and how they contribute to the input of the message imprint's computation.

5.1.4.4.2 *Include mechanism*

5.1.4.4.2.1 *Semantics and syntax*

Semantics

`Include` elements shall explicitly reference data objects that contribute to the input of the electronic time-stamp's message imprint computation, and consequently are time-stamped by the electronic time-stamp.

The order of appearance of the `Include` elements shall indicate the order in which the referenced data objects contribute to the input of the electronic time-stamp's message imprint computation.

Syntax

The `URI` attribute in `Include` element shall reference one data object that contributes to the input of the electronic time-stamp's message imprint computation.

The value of `URI` attribute follows the rules indicated below:

a) It shall have an empty non-fragment part and a bare-name XPointer fragment when the `Include` element and the time-stamped data object are in the same document.
b) It shall have a not empty non-fragment part and a bare-name XPointer fragment when the `Include` element and the time-stamped data object are not in the same document.
c) If not empty, its non-fragment part shall be equal to:
    1) the non-fragment part of the Target attribute of the `QualifyingProperties` enclosing the `Include` element if the time-stamped data object is enveloped by the XAdES signature; or
    2) the non-fragment part of the URI attribute of the `QualifyingPropertiesReference` element referencing the `QualifyingProperties` element enveloping the time-stamped data object if this `QualifyingProperties` element is not enveloped by the XAdES signature.

If the object referenced by the `URI` attribute is not a `ds:Reference` element, the `referencedData` attribute shall not be present.

If the object referenced by the URI attribute is a `ds:Reference` element, the `referencedData` attribute may be present.

NOTE — The presence and value of `referencedData` attribute impacts the computation of the octets that will contribute to the input of the electronic time-stamp's message imprint computation as specified below in **5.1.4.4.2.3**.

5.1.4.4.2.2 *Processing model for URI attribute*

The retrieved resource shall be parsed, and then the bare-name XPointer shall be processed.

The bare-name XPointer shall be processed as follows:

a) Use as XPointer evaluation context the root node of the XML document that contains the element referenced by the not-fragment part of URI attribute's value; and
b) Derive a XPath node-set from the resultant location-set as indicated below:
    1) Replace the element node E retrieved by the bare-name XPointer with E plus all descendants of E (text, comments, PIs, elements) and all namespace and attribute nodes of E and its descendant elements; and

2) Delete all the comment nodes.

5.1.4.4.2.3 *Processing model for include element*

Each `Include` element within a time-stamp container qualifying property shall be processed as detailed below:

a) Retrieve the data object referenced in the URI attribute as specified in **5.1.4.4.2.2**;

b) If the retrieved data object is a `ds:Reference` element and the `referencedData` attribute is set to the value "true", take the result of processing the retrieved ds:Reference element according to the reference processing model of XMLDSIG, **4.4.3.2**; otherwise keep the retrieved data object retrieved in a);

c) If the data object obtained after step b) is an XML node set, canonicalize it as specified in **4.5**; and

d) Concatenate the resulting octets to the input of the electronic time-stamp's message imprint computation resulting from previous processing as indicated in the corresponding time-stamp container qualifying property.

5.1.4.5 *The OtherTimeStampType data type*

Semantics

This concrete derived type shall contain electronic time-stamps computed on a collection of data objects that are not incorporated into the XAdES signature.

Syntax

The `OtherTimeStampType` type shall be defined as in XML Schema file **Error! Reference source not found.**, whose location is detailed in D-1 of Annex D and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="OtherTimeStamp" type="OtherTimeStampType"/>

<xsd:complexType name="OtherTimeStampType">
      <xsd:complexContent>
            <xsd:restriction base="GenericTimeStampType">
                  <xsd:sequence>
                        <xsd:element ref="ReferenceInfo"
maxOccurs="unbounded"/>
                        <xsd:element ref="ds:CanonicalizationMethod"
minOccurs="0"/>
                        <xsd:choice>
                              <xsd:element name="EncapsulatedTimeStamp"
                        type="EncapsulatedPKIDataType"/>
                              <xsd:element name="XMLTimeStamp"
type="AnyType"/>
                  </xsd:choice>
                  </xsd:sequence>
                  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
            </xsd:restriction>
      </xsd:complexContent>
```

```
</xsd:complexType>
```

For this type the actual input to the computation of the message imprint shall be the concatenation (in the order of appearance) of the present ReferenceInfo elements, canonicalized as specified in **4.5.**

Each `ReferenceInfo` element shall contain the digest of one external data object.

The `URI` attribute shall reference the data object contributing to the input of the electronic time-stamp's message imprint. As in XMLDSIG, if it is omitted, the context where the XAdES signature is used shall allow to know the identity of the referenced object.

Element `ds:DigestMethod` shall identify the digest algorithm applied to the external data object.

Element `ds:DigestValue` shall contain the base-64 encoded value of the digest of the referenced data object.

Attribute Id shall be used for referencing this element from elsewhere.

Attribute Id and elements `ds:CanonicalizationMethod`, `EncapsulatedTimeStamp` and `XMLTimeStamp` shall be used exactly as in `XAdESTimeStampType`.

5.2 Basic Qualifying Properties For XAdES Signatures

5.2.1 *The SigningTime Qualifying Property*

Semantics

The `SigningTime` qualifying property shall be a signed qualifying property that qualifies the signature.

The `SigningTime` qualifying property's value shall specify the time at which the signer claims to having performed the signing process.

Syntax

The `SigningTime` qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in D-1 of Annex D and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="SigningTime" type="xsd:dateTime"/>
```

5.2.2 *The SigningCertificateV2 Qualifying Property*

Semantics

The `SigningCertificateV2` qualifying property shall be a signed qualifying property that qualifies the signature.

The `SigningCertificateV2` qualifying property shall contain one reference to the signing certificate.

The `SigningCertificateV2` qualifying property may contain references to some of or all the certificates within the signing certificate path, including one reference to the trust anchor when this is a certificate.

NOTES

1 For instance, the signature validation policy can mandate other certificates to be present which can include all the certificates up to the trust anchor.

For each certificate, the `SigningCertificateV2` qualifying property shall contain a digest value together with a unique identifier of the algorithm that has been used to calculate it.

The first reference in `SigningCertificateV2` qualifying property shall be the reference of the signing certificate.

Syntax

The `SigningCertificateV2` qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in D-1 and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="SigningCertificateV2" type="CertIDListV2Type"/>

<xsd:complexType name="CertIDListV2Type">
     <xsd:sequence>
          <xsd:element name="Cert" type="CertIDTypeV2"
maxOccurs="unbounded"/>
     </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CertIDTypeV2">
     <xsd:sequence>
          <xsd:element name="CertDigest"
type="DigestAlgAndValueType"/>
          <xsd:element name="IssuerSerialV2"
type="xsd:base64Binary" minOccurs="0"/>
     </xsd:sequence>
     <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
</xsd:complexType>

<xsd:complexType name="DigestAlgAndValueType">
     <xsd:sequence>
          <xsd:element ref="ds:DigestMethod"/>
          <xsd:element ref="ds:DigestValue"/>
     </xsd:sequence>
</xsd:complexType>
```

The element `CertDigest` shall contain the digest of the referenced certificate.

`CertDigest's` children elements satisfy the following requirements:

a) `ds:DigestMethod` element shall identify the digest algorithm; and
b) `ds:DigestValue` element shall contain the base-64 encoded value of the digest computed on the DER-encoded certificate.

The content of `IssuerSerialV2` element shall be the base-64 encoding of one DER-encoded instance of type `IssuerSerial` type defined in IETF RFC 5035.

2 The information in the `IssuerSerialV2` element is only a hint, that can help to identify the certificate whose digest matches the value present in the reference. But the binding information is the digest of the certificate.

The `URI` attribute shall provide an indication of where the referenced certificate can be found.

3 It is intended that this attribute be used as a hint, as implementations can have alternative ways for retrieving the referenced certificate if it is not found at the referenced place.

5.2.3 *The CommitmentTypeIndication Qualifying Property*

Semantics

The `CommitmentTypeIndication` qualifying property shall be a signed qualifying property that qualifies signed data object(s).

The `CommitmentTypeIndication` qualifying property shall indicate one commitment made by the signer when signing.
The `CommitmentTypeIndication` qualifying property may indicate one commitment made by the signer for a subset of the set of signed data objects.

The `CommitmentTypeIndication` qualifying property may also indicate one commitment made by the signer for the complete set of signed data objects.

The `CommitmentTypeIndication` qualifying property shall express the commitment type with a URI.

The `CommitmentTypeIndication` qualifying property may contain a sequence of qualifiers providing more information about the commitment.

NOTES

1 The commitment type can be:
   i. defined as part of the signature policy, in which case, the commitment type has precise semantics that are defined as part of the signature policy; or
   ii. be a registered type, in which case, the commitment type has precise semantics defined by registration, under the rules of the registration authority. Such a registration authority can be a trading association or a legislative authority.

2 The specification of commitment type identifiers is outside the scope of the standard. For a list of predefined commitment type identifiers, see ETSI TS 119 172-1.

Syntax

The `CommitmentTypeIndication` qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**"", whose location is detailed in D-1 of Annex D and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="CommitmentTypeIndication"
type="CommitmentTypeIndicationType"/>

<xsd:complexType name="CommitmentTypeIndicationType">
     <xsd:sequence>
          <xsd:element name="CommitmentTypeId"
       type="ObjectIdentifierType"/>
          <xsd:choice>
               <xsd:element name="ObjectReference"
type="xsd:anyURI"
         maxOccurs="unbounded"/>
               <xsd:element name="AllSignedDataObjects"/>
          </xsd:choice>
          <xsd:element name="CommitmentTypeQualifiers"
      type="CommitmentTypeQualifiersListType" minOccurs="0"/>
     </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CommitmentTypeQualifiersListType">
     <xsd:sequence>
          <xsd:element name="CommitmentTypeQualifier"
      type="AnyType" minOccurs="0" maxOccurs="unbounded"/>
     </xsd:sequence>
</xsd:complexType>
```

The CommitmentTypeId element is an element of ObjectIdentifierType type, which fulfils the following requirements:

a) Its Identifier child shall have a URI as value, uniquely identifying one commitment made by the signer; and
b) Its Identifier child shall not have a Qualifier attribute (i.e. the aforementioned URI shall not represent an OID value).

Each ObjectReference shall reference one ds:Reference element within the ds:SignedInfo element or within a signed ds:Manifest element.

If a commitment is made only for a subset (different from the full set) of signed data objects, the CommitmentTypeIndication element shall incorporate one ObjectReference element for each one of signed data objects in the aforementioned subset.

If a certain commitment is made for all the signed data objects, the CommitmentTypeIndication qualifying property shall contain:

a) one AllSignedDataObjects empty element; or
b) one ObjectReference element for each one of the signed data objects except the SignedProperties element (XAdES signed qualifying properties).

The CommitmentTypeQualifiers element provides means to include additional qualifying information on the commitment made by the signer.

5.2.4 *The DataObjectFormat Qualifying Property*

Semantics

The DataObjectFormat qualifying property shall be a signed qualifying property that qualifies one specific signed data object.

The DataObjectFormat qualifying property shall contain information that describes the format of the signed data object.

Syntax

The DataObjectFormat qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in Clause D-1 of Annex D and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="DataObjectFormat" type="DataObjectFormatType"/>

<xsd:complexType name="DataObjectFormatType">
        <xsd:sequence>
                <xsd:element name="Description" type="xsd:string" minOccurs="0"/>
                <xsd:element name="ObjectIdentifier" type="ObjectIdentifierType"
    minOccurs="0"/>
                <xsd:element name="MimeType" type="xsd:string" minOccurs="0"/>
                <xsd:element name="Encoding" type="xsd:anyURI" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attribute name="ObjectReference" type="xsd:anyURI"
  use="required"/>
</xsd:complexType>
```

Element Description shall contain textual information related to the signed data object.

Element ObjectIdentifier shall contain an identifier of the type of the signed data object, as assigned by an authority that defines that type.

Element MimeType shall contain a MIME type value indicating the format of the signed data object. Its content shall be a string containing values defined by IETF RFC 2045.

Element Encoding shall contain an indication of the encoding of the signed data object.

This qualifying property shall contain at least one of the following elements: Description, ObjectIdentifier and MimeType.

The ObjectReference attribute shall reference the ds:Reference child of the ds:SignedInfo or a signed ds:Manifest element referencing the signed data object qualified by this qualifying property.

If the DataObjectFormat qualifying property references a ds:Reference that in turn references a ds:Object within the XAdES signature, and if this ds:Object element has the MimeType or (and) the

Encoding attribute(s), then DataObjectFormat's children MimeType and Encoding shall have exactly the same values, if they are present.

5.2.5 *The SignatureProductionPlaceV2 Qualifying Property*

Semantics

The SignatureProductionPlaceV2 qualifying property shall be a signed qualifying property that qualifies the signer.

The SignatureProductionPlaceV2 qualifying property shall specify an address associated with the signer at a particular geographical (for example city) location.

Syntax

The SignatureProductionPlaceV2 qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in

D-1 XML SCHEMA FILE **LOCATION FOR NAMESPACE** and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="SignatureProductionPlaceV2" type="SignatureProductionPlaceV2Type"/>

<xsd:complexType name="SignatureProductionPlaceV2Type">
        <xsd:sequence>
                <xsd:element name="City" type="xsd:string" minOccurs="0"/>
                <xsd:element name="StreetAddress" type="xsd:string" minOccurs="0"/>
                <xsd:element name="StateOrProvince" type="xsd:string" minOccurs="0"/>
                <xsd:element name="PostalCode" type="xsd:string" minOccurs="0"/>
                <xsd:element name="CountryName" type="xsd:string" minOccurs="0"/>
        </xsd:sequence>
</xsd:complexType>
```

Empty SignatureProductionPlaceV2 qualifying properties shall not be generated.

5.2.6 *The SignerRoleV2 Qualifying Property*

Semantics

The SignerRoleV2 qualifying property shall be a signed qualifying property that qualifies the signer.

The SignerRoleV2 qualifying property shall encapsulate signer attributes (for example ~~e.g.~~ role). This qualifying property may encapsulate the following types of attributes:

a) Attributes claimed by the signer;
b) Attributes certified in attribute certificates issued by an Attribute Authority; or/and
c) Assertions signed by a third party.

Syntax

The SignerRoleV2 qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in Clause D-1 of Annex D, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="SignerRoleV2" type="SignerRoleV2Type"/>

<xsd:complexType name="SignerRoleV2Type">
	<xsd:sequence>
		<xsd:element ref="ClaimedRoles" minOccurs="0"/>
		<xsd:element ref="CertifiedRolesV2" minOccurs="0"/>
		<xsd:element ref="SignedAssertions" minOccurs="0"/>
	</xsd:sequence>
</xsd:complexType>

<xsd:element name="ClaimedRoles" type="ClaimedRolesListType"/>
<xsd:element name="CertifiedRolesV2" type="CertifiedRolesListTypeV2"/>
<xsd:element name="SignedAssertions" type="SignedAssertionsListType"/>

<xsd:complexType name="ClaimedRolesListType">
	<xsd:sequence>
		<xsd:element name="ClaimedRole" type="AnyType" maxOccurs="unbounded"/>
	</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CertifiedRolesListTypeV2">
	<xsd:sequence>
		<xsd:element name="CertifiedRole" type="CertifiedRoleTypeV2"
maxOccurs="unbounded"/>
	</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CertifiedRoleTypeV2">
	<xsd:choice>
		<xsd:element ref="X509AttributeCertificate"/>
		<xsd:element ref="OtherAttributeCertificate"/>
	</xsd:choice>
</xsd:complexType>

<xsd:element name="X509AttributeCertificate" type="EncapsulatedPKIDataType"/>
<xsd:element name="OtherAttributeCertificate" type="AnyType"/>

<xsd:complexType name="SignedAssertionsListType">
	<xsd:sequence>
		<xsd:element ref="SignedAssertion" maxOccurs="unbounded"/>
	</xsd:sequence>
</xsd:complexType>

<xsd:element name="SignedAssertion" type="AnyType"/>
```

The ClaimedRoles element shall contain a non-empty sequence of roles claimed by the signer but which are not certified.

Additional content types may be defined on a domain application basis and be part of this element.

NOTES

1 The namespaces given to the corresponding XML schemas allow their unambiguous identification in the case these attributes are expressed in XML syntax (~~e.g.~~ for example SAML assertions of different versions).

The CertifiedRolesV2 element shall contain a non-empty sequence of certified attributes, which shall be one of the following:

a) The base-64 encoding of DER-encoded X509 attribute certificates conformant to Recommendation ITU-T X.509 issued to the signer, within the X509AttributeCertificate element; or
b) Attribute certificates (issued, in consequence, by Attribute Authorities) in different syntax than the one specified in Recommendation ITU-T X.509, within the OtherAttributeCertificate element. The definition of specific OtherAttributeCertificate is outside of the scope of the standard.

The SignedAssertions element shall contain a non-empty sequence of assertions signed by a third party.

2 A signed assertion is stronger than a claimed attribute, since a third party asserts with a signature that the attribute of the signer is valid. However, it is less restrictive than an attribute certificate.

The definition of specific content types for SignedAssertions is outside of the scope of the standard.

3 A possible content can be a signed SAML assertion.

Empty SignerRoleV2 qualifying properties shall not be generated.

5.2.7 *Countersignatures*

5.2.7.1 *Countersignature identifier in Type attribute of ds:Reference*

The standard defines the following URI value:

- http://uri.etsi.org/01903#CountersignedSignature.

A XAdES signature containing a ds:Reference element whose Type attribute has this value shall indicate that it is a countersignature of the signature referenced by this element.

The ds:Reference element shall be built so that the countersignature signs the ds:SignatureValue element of the countersigned signature.

All the XMLDSIG rules shall apply in the processing of the aforementioned ds:Reference element.

NOTE — The only purpose of this definition is to serve as an easy identification of a signature as being a countersignature.

5.2.7.2 *Enveloped countersignatures: the CounterSignature qualifying property*

Semantics

The CounterSignature qualifying property shall be an unsigned qualifying property that qualifies the signature.

The CounterSignature qualifying property shall contain one countersignature of the XAdES signature where CounterSignature is incorporated. This countersignature may also be a XAdES signature.

Syntax

The CounterSignature qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in Clause D-1 of Annex D and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="CounterSignature" type="CounterSignatureType" />

<xsd:complexType name="CounterSignatureType">
      <xsd:sequence>
            <xsd:element ref="ds:Signature"/>
      </xsd:sequence>
      <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

NOTES

1 The Id attribute has been incorporated in the definition of CounterSignatureType for allowing this unsigned property to be referenced by a URI in case the signature uses indirect incorporation of properties and an electronic time-stamp container needs to refer to the counter-signature through an Include element.

The content of this qualifying property shall be a XMLDSIG or XAdES signature whose ds:SignedInfo shall contain one ds:Reference element referencing the ds:SignatureValue element of the embedding and countersigned XAdES signature.

The content of the ds:DigestValue in the aforementioned ds:Reference element of the countersignature shall be the base-64 encoded digest of the complete (and canonicalized) ds:SignatureValue element (i.e. including the starting and closing tags) of the embedding and countersigned XAdES signature.

Other ds:Reference elements referencing other data objects may be added to the countersignature.

2 This includes, for instance, ds:Reference elements referencing the ds:SignatureValue elements of previously existent CounterSignature elements. This allows for building arbitrarily long chains of explicit countersignatures.

A countersignature may itself be signed using a CounterSignature qualifying property, which shall have a ds:Reference element referencing the ds:SignatureValue of the first countersignature, built as described above.

3 This is an alternative way of constructing arbitrarily long series of countersignatures, each one signing the ds:SignatureValue element of the one where it is directly embedded.

FIGURE 1 illustrates this qualifying property and its relationship with the countersigned XAdES signature.

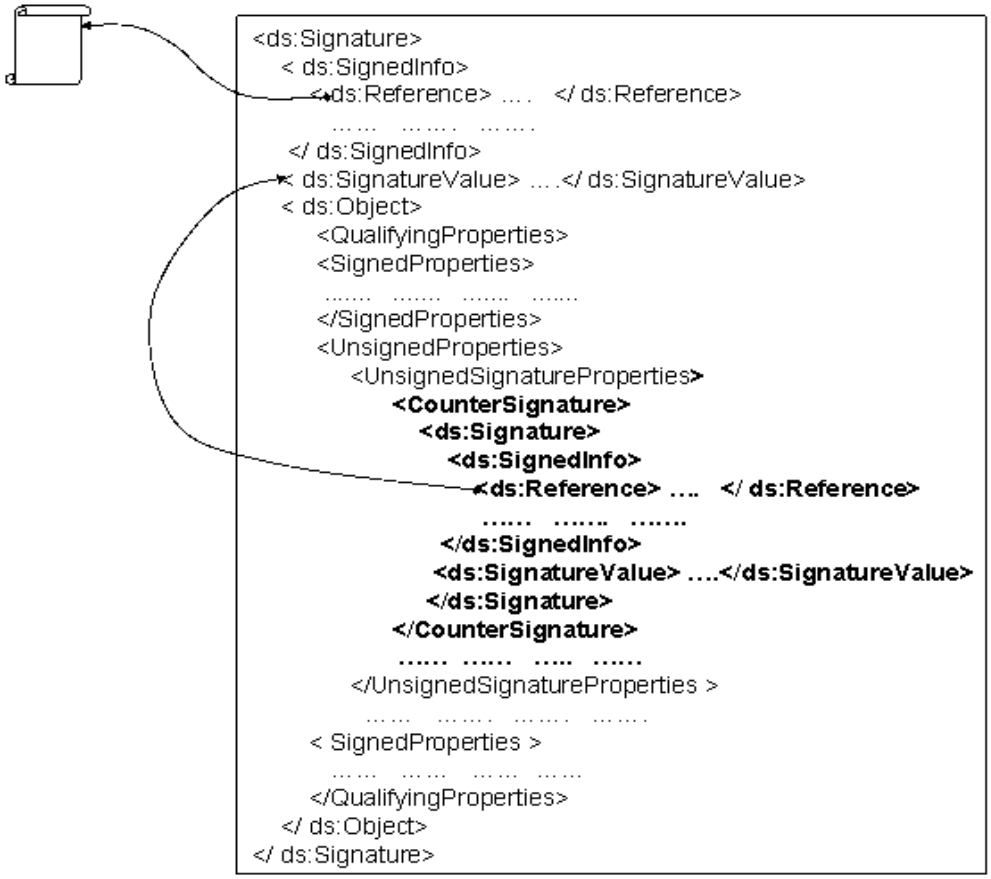

FIGURE 1 USE OF COUNTERSIGNATURE ELEMENT

5.2.8 *Time-Stamps On Signed Data Objects*

5.2.8.1 *The AllDataObjectsTimeStamp qualifying property*

Semantics

The AllDataObjectsTimeStamp qualifying property shall be a signed qualifying property that qualifies signed data objects.

The AllDataObjectsTimeStamp qualifying property shall encapsulate one or more electronic time-stamps, generated before the signature production, whose message imprint computation input is the concatenation of all the objects obtained after processing as specified in XMLDSIG, **4.4.3.2**; all the ds:Reference elements within the ds:SignedInfo except the one referencing the SignedProperties element, in their order of appearance.

Syntax

The AllDataObjectsTimeStamp qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in Clause D-1 of Annex D and is copied below for information.

<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="AllDataObjectsTimeStamp" type="XAdESTimeStampType"/>

The Implicit mechanism (see **1**) shall be used for generating this qualifying property.

The input to the computation of the message imprint shall be computed as follows:

a) Initialize the final octet stream as an empty octet stream.

b) Take all the ds:Reference elements in their order of appearance within ds:SignedInfo, except the one referencing the SignedProperties element. Process each one as indicated below:

1) Retrieve the data object referenced by the URI attribute of the ds:Reference element, as specified in **4.4.3.2** of XMLDSIG.

NOTES

1 **4.4.3.2** of XMLDSIG, specifies rules for URI dereferencing. For instance, it mandates that the dereferencing of a non 'same-document' reference (which XMLDSIG defines as "a URI-Reference that consists of a hash sign ('#') followed by a fragment or alternatively consists of an empty URI") is always an octet-stream.

2) If the ds:Reference element does not contain the ds:Transforms element, then:

i) if the retrieved data object is an XML node-set, then canonicalize it as specified in **4.5** of the standard;

ii) else proceed to step 4).

3) If the ds:Reference element contains the ds:Transforms element, then apply all the transforms indicated within the ds:Transform children elements. After that:

i) if the output of the last transform is a XML node-set according to XMLDSIG, canonicalize it as specified in **4.5** of the standard;

ii) else proceed to step 4).

4) Concatenate the resulting octets to the final octet stream.

2 Values of data objects referenced by ds:Reference elements present within signed ds:Manifest do not contribute to the message imprint computation input. However, the digest values of data objects that are referenced by ds:Reference elements within ds:Manifest elements that are referenced by ds:Reference elements within ds:SignedInfo, do contribute to the message imprint computation input.

5.2.8.2 *The IndividualDataObjectsTimeStamp qualifying property*

Semantics

The IndividualDataObjectsTimeStamp qualifying property shall be a signed qualifying property that qualifies signed data objects.

The IndividualDataObjectsTimeStamp qualifying property shall encapsulate one or more electronic time-stamps, generated before the signature production, whose message imprint computation input is the concatenation of the objects obtained after processing as specified in XMLDSIG, **4.4.3.2**; some of the ds:Reference elements within the ds:SignedInfo or also signed ds:Manifest.

The set of ds:Reference elements processed shall not include the one referencing the SignedProperties element.

Syntax

The IndividualDataObjectsTimeStamp qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in Clause D-1 of Annex D and is copied below for information.

<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="IndividualDataObjectsTimeStamp" type="XAdESTimeStampType"/>

The explicit (Include) mechanism shall be used for generating this qualifying property.

The Include elements shall be composed to refer to those ds:Reference elements referencing the data objects that have to be time-stamped.

The referencedData attribute shall be present in each and every Include element, and set to "true".

The message imprint computation input shall be computed as follows:

a) Initialize the final octet stream as an empty octet stream.

b) Take all the ds:Reference elements within ds:SignedInfo or within a signed ds:Manifest which are referenced within the Include element. Process each one in their order of appearance within the Include element as indicated below:

    1) Retrieve the data object referenced by the URI attribute of the ds:Reference element, as specified in **4.4.3.2** of XMLDSIG.

NOTE —**4.4.3.2** of XMLDSIG, specifies rules for URI dereferencing. For instance, it mandates that the dereferencing of a non 'same-document' reference (which XMLDSIG defines as "a URI-Reference that consists of a hash sign ('#') followed by a fragment or alternatively consists of an empty URI") is always an octet-stream.

    2) If the ds:Reference element does not contain the ds:Transforms element, then:

        i) if the retrieved data object is an XML node-set, then canonicalize it as specified in **4.5** of the standard;

        ii) else proceed to step 4).

    3) If the ds:Reference element contains the ds:Transforms element, then apply all the transforms indicated within the ds:Transform children elements. After that:

        i) if the output of the last transform is a XML node-set according to XMLDSIG, canonicalize it as specified in **4.5** of the standard;

        ii) else proceed to step 4).

    4) Concatenate the resulting octets to the final octet stream.

5.2.9 *The SignaturePolicyIdentifier Qualifying Property*

5.2.9.1 *Semantics and syntax*

Semantics

The SignaturePolicyIdentifier qualifying property shall be a signed qualifying property qualifying the signature.

The SignaturePolicyIdentifier qualifying property shall contain either an explicit identifier of a signature policy or an indication that there is an implied signature policy that the relying party should be aware of.

NOTES

1 ETSI TS 119 172-1 specifies a framework for signature policies.

Syntax

The SignaturePolicyIdentifier qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in Clause D-1 of Annex D and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="SignaturePolicyIdentifier"
type="SignaturePolicyIdentifierType"/>

<xsd:complexType name="SignaturePolicyIdentifierType">
  <xsd:choice>
    <xsd:element name="SignaturePolicyId" type="SignaturePolicyIdType"/>
    <xsd:element name="SignaturePolicyImplied"/>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="SignaturePolicyIdType">
  <xsd:sequence>
    <xsd:element name="SigPolicyId" type="ObjectIdentifierType"/>
    <xsd:element ref="ds:Transforms" minOccurs="0"/>
    <xsd:element name="SigPolicyHash" type="DigestAlgAndValueType"/>
    <xsd:element name="SigPolicyQualifiers"
      type="SigPolicyQualifiersListType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="SigPolicyQualifiersListType">
  <xsd:sequence>
    <xsd:element name="SigPolicyQualifier" type="AnyType"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

The SignaturePolicyId element shall be used for referencing the signature policy explicitly.

The SigPolicyId element shall uniquely identify a specific version of the signature policy.

The ds:Transforms element shall contain the transformations performed on the signature policy document before computing its hash. The processing model for these transformations shall be as described in XMLDSIG.

The SigPolicyHash element shall contain the identifier of the hash algorithm and the hash value of the object obtained after processing SigPolicyId and ds:Transforms if present.

The standard defines a new transform, which shall be identified by setting the value of the Algorithm attribute of the ds:Transform element to:

- http://uri.etsi.org/01903/v1.3.2/SignaturePolicy/SPDocDigestAsInSpecification.

This transform shall indicate that the hash value of the signature policy document has been computed as specified in a certain technical specification.
If this transform is used, then the SignaturePolicyIdentifier shall be qualified at least by the SPDocSpecification qualifier, specified in Error! Reference source not found., which identifies the aforementioned technical specification.

2 This transform can be used when the technical specification defines a mechanism for computing the hash value of the signature policy document that is not easily implementable using widely used XML technologies (for example XPath), as can occur, for instance, when the signature policy document is DER-encoded ASN.1.

This transform shall not be used in elements different than SignaturePolicyId element.

The SigPolicyQualifier element may contain additional information qualifying the signature policy identifier.

The SigPolicyQualifiers element shall contain one or more qualifiers of the signature policy.

The SigPolicyQualifiers element may contain one or more qualifiers of the same type.

The SignaturePolicyImplied empty element shall indicate that the data object(s) being signed and other external data imply the signature policy.

3 The SignaturePolicyImplied element can be used when the signature policy can be unambiguously derived from the semantics of the type of data object(s) being signed, and some other information.

5.2.9.2 *Signature policy qualifiers*

Semantics

Three qualifiers for the signature policy have been identified so far:

a) A URL where a copy of the signature policy document can be obtained (SPURI element, defined in the namespace whose URI is http://uri.etsi.org/01903/v1.3.2#).

b) A user notice that should be displayed when the signature is validated (SPUserNotice element, defined in the namespace whose URI is http://uri.etsi.org/01903/v1.3.2#).

c) An identifier of the technical specification that defines the syntax used for producing the signature policy document (SPDocSpecification element, defined in the namespace whose URI is http://uri.etsi.org/01903/v1.4.1#).

Syntax

The SPURI and SPUserNotice elements shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in Clause D-1 of Annex D, and are copied below for information:

NOTES

1 These elements are defined within the namespace whose URI is http://uri.etsi.org/01903/v1.3.2#.

<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

```
<xsd:element name="SPURI" type="xsd:anyURI"/>
<xsd:element name="SPUserNotice" type="SPUserNoticeType"/>

<xsd:complexType name="SPUserNoticeType">
  <xsd:sequence>
    <xsd:element name="NoticeRef" type="NoticeReferenceType"
      minOccurs="0"/>
    <xsd:element name="ExplicitText" type="xsd:string"
      minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="NoticeReferenceType">
  <xsd:sequence>
    <xsd:element name="Organization" type="xsd:string"/>
    <xsd:element name="NoticeNumbers" type="IntegerListType"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="IntegerListType">
  <xsd:sequence>
    <xsd:element name="int" type="xsd:integer" minOccurs="0"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

The SPURI element shall contain a URL value where a copy of the signature policy document can be obtained.

2 This URL can reference, for instance, a remote site (which can be managed by an entity entitled for this purpose) from where (signing/validating) applications can retrieve the signature policy document.

The SPUserNotice element shall contain information that is intended for being displayed whenever the signature is validated.

The ExplicitText element shall contain the text of the notice to be displayed.

3 Other notices can come from the organization issuing the signature policy.

The NoticeRef element shall name an organization and shall identify by numbers (NoticeNumbers element) a group of textual statements prepared by that organization, so that the application could get the explicit notices from a notices file.

The SPDocSpecification shall identify the technical specification that defines the syntax used for producing the signature policy document.

The SPDocSpecification shall be defined as in XML Schema file "XAdES01903v141-202107.xsd", whose location is detailed in Clause D-2 of Annex D, and is copied below for information.

4 This element is defined within the namespace whose URI is http://uri.etsi.org/01903/v1.4.1#.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.4.1#"
```

The preamble of the XML Schema file also includes the following namespace declaration:
xmlns:xades="http://uri.etsi.org/01903/v1.3.2#",
which assigns the prefix "xades" to the namespace whose URI is shown in the declaration.

```
-->

<xsd:element name="SPDocSpecification" type="xades:ObjectIdentifierType"/>
```

If the technical specification is identified using an OID, then the Identifier child shall contain a URN encoding this OID as specified in IETF RFC 3061, and its QualifierType attribute shall be present with its value set to "OIDAsURN".

If the technical specification is identified using a URI, then the Identifier child shall contain this URI and its QualifierType attribute shall not be present.

5 This qualifier allows identifying whether the signature policy document is human readable, XML encoded, or ASN.1 encoded, by identifying the specific technical specifications where these formats will be defined.

5.2.10 *The SignaturePolicyStore Qualifying Property*

Semantics

The SignaturePolicyStore qualifying property shall be an unsigned qualifying property qualifying the signature.

The SignaturePolicyStore qualifying property shall contain either:

a) the signature policy document which is referenced in the SignaturePolicyIdentifier qualifying property so that the signature policy document can be used for offline and long-term validation; or
b) a URI referencing a local store where the signature policy document can be retrieved.

Syntax

The SignaturePolicyStore shall be defined as in XML Schema file "XAdES01903v141-202107.xsd", whose location is detailed in Clause D-2 of Annex D, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.4.1#" -->

<xsd:element name="SignaturePolicyStore" type="SignaturePolicyStoreType"/>

<xsd:complexType name="SignaturePolicyStoreType">
        <xsd:sequence>
                <xsd:element ref="SPDocSpecification"/>
                <xsd:choice>
                        <xsd:element name="SignaturePolicyDocument" type="xsd:base64Binary"/>
                        <xsd:element name="SigPolDocLocalURI" type="xsd:anyURI"/>
                </xsd:choice>
        </xsd:sequence>
        <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

The SignaturePolicyDocument element shall contain the base-64 encoded signature policy.

The SigPolDocLocalURI element shall have as value the URI referencing a local store where the standard can be retrieved.
NOTES

1 Contrary to the SPURI, the SigPolDocLocalURI points to a local file.

The SPDocSpecification element shall identify the technical specification that defines the syntax used for producing the signature policy document.

2 It is the responsibility of the entity incorporating the signature policy to the signature-policy-store to make sure that the correct document is securely stored.

3 Being an unsigned qualifying property, it is not protected by the digital signature. If the SignaturePolicyIdentifier qualifying property is incorporated into the signature and contains the SigPolicyHash element with the digest value of the signature policy document, any alteration of the signature policy document present within SignaturePolicyStore or within a local store, would be detected by the failure of the digests comparison.

5.3 The SignatureTimeStamp Qualifying Property

Semantics

The SignatureTimeStamp qualifying property shall be an unsigned qualifying property qualifying the signature.

The SignatureTimeStamp qualifying property shall encapsulate one or more electronic time-stamps time-stamping the ds:SignatureValue element.

Syntax

The SignatureTimeStamp qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in Clause D-1 of Annex D and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="SignatureTimeStamp" type="XAdESTimeStampType"/>
```

The Implicit mechanism shall be used for generating this qualifying property.

The input to the electronic time-stamp's message imprint computation shall be built as indicated below:

a) take the ds:SignatureValue element and its contents; and

b) canonicalize it as specified in **4.5**.

5.4 Qualifying Properties For Validation Data Values

5.4.1 *The CertificateValues Qualifying Property*

Semantics

The CertificateValues qualifying property shall be an unsigned qualifying property qualifying the signature.

The CertificateValues qualifying property:

a) Shall contain the certificate of the trust anchor, if such certificate does exist and if it is not present within the ds:KeyInfo. If this certificate is present within the ds:KeyInfo, it should not be included.
b) Shall contain the CA certificates within the signing certificate path that are not present within the ds:KeyInfo. The certificates present within ds:KeyInfo element should not be included.
c) Shall contain the signing certificate if it is not present within the ds:KeyInfo. If this certificate is present within the ds:KeyInfo, it should not be included.
d) Shall contain certificates used to sign revocation status information (for example CRLs or OCSP responses) of certificates in a~~1~~), b~~2~~), and c~~3~~), and certificates within their respective certificate paths that are not present in the signature. Certificate values present within the signature, including certificate values within the revocation status information themselves should not be included.
e) Shall not contain CA certificates that pertain exclusively to the certificate paths of certificates used to sign attribute certificates or signed assertions within SignerRoleV2, or electronic time-stamps. And
f) May contain a set of certificates used to validate any countersignature incorporated into the XAdES signature that are not present in other elements of the XAdES signature or its countersignatures. This set may include any of the certificates listed in a), b), c) and d) referred to signing certificates of countersignatures instead of the signing certificate of the XAdES signature. The certificates present elsewhere in the XAdES signature or its countersignatures should not be included.

Syntax

The CertificateValues qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in D-1 of Annex D, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="CertificateValues" type="CertificateValuesType"/>

<xsd:complexType name="CertificateValuesType">
   <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="EncapsulatedX509Certificate"
       type="EncapsulatedPKIDataType"/>
      <xsd:element name="OtherCertificate" type="AnyType"/>
   </xsd:choice>
   <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

The EncapsulatedX509Certificate element shall contain the base-64 encoding of a DER-encoded X.509 certificate.
The OtherCertificate element is a placeholder for potential future new formats of certificates.

NOTE — If XML electronic time-stamps based in XMLDSIG are standardized and spread, this type can also be used to contain the certification chain for any TSUs providing such electronic time-stamps, if these certificates are not already present in the electronic time-stamps themselves as part of the TSUs' signatures. In this case, an element of

this type can be added as an unsigned property to the XML electronic time-stamp using the incorporation mechanisms defined in the standard.

5.4.2 *The RevocationValues Qualifying Property*

Semantics

The RevocationValues qualifying property shall be an unsigned qualifying property that qualifies the signature.

The RevocationValues qualifying property:

a) Shall contain revocation values corresponding to CA certificates within the signing certificate path if they are not present within the ds:KeyInfo. It shall not contain a revocation value for the trust anchor. The revocation values present within ds:KeyInfo element should not be included.

b) Shall contain a revocation value for the signing certificate if it is not present within the ds:KeyInfo. If it is present within ds:KeyInfo element, it should not be included.

c) May contain revocation values corresponding to certificates used to sign CRLs or OCSP responses of a) and b), and certificates within their respective certificate paths. The revocation values present within ds:KeyInfo element should not be included.

d) Shall not contain revocation values corresponding to CA certificates that pertain exclusively to the certificate paths of certificates used to sign attribute certificates or signed assertions within SignerRoleV2, or electronic time-stamps. And

e) May contain revocation values corresponding to the signing certificate of any countersignature incorporated into the XAdES signature as well as to the CA certificates in its certificate path. This set may include any of the revocation values listed in a), b), and c) referred to signing certificates of countersignatures instead of the signing certificate of the XAdES signature. However, those revocation values among the aforementioned ones that are already present in other elements of the XAdES signature should not be included.

Syntax

The RevocationValues qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in D-1 of Annex D, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="RevocationValues" type="RevocationValuesType"/>

<xsd:complexType name="RevocationValuesType">
  <xsd:sequence>
    <xsd:element name="CRLValues" type="CRLValuesType"
      minOccurs="0"/>
    <xsd:element name="OCSPValues" type="OCSPValuesType"
      minOccurs="0"/>
    <xsd:element name="OtherValues" type="OtherCertStatusValuesType"
      minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID"  use="optional"/>
</xsd:complexType>
```

```
<xsd:complexType name="CRLValuesType">
  <xsd:sequence>
     <xsd:element name="EncapsulatedCRLValue"
       type="EncapsulatedPKIDataType"
       maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="OCSPValuesType">
  <xsd:sequence>
      <xsd:element name="EncapsulatedOCSPValue"
        type="EncapsulatedPKIDataType" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="OtherCertStatusValuesType">
  <xsd:sequence>
    <xsd:element name="OtherValue" type="AnyType"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

CRLValues element shall contain a sequence of encoded X.509 CRLs.

Each EncapsulatedCRLValue child of CRLValues element shall contain the base-64 encoding of a DER-encoded X.509 CRL.

If the validation data contain one or more Delta CRLs, the CRLValues element shall contain the set of CRLs required to provide complete revocation lists.

OCSPValues element shall contain a sequence of encoded OCSP responses.

Each EncapsulatedOCSPValue child of OCSPValues element shall contain the base-64 encoding of a DER-encoded OCSPResponse defined in IETF RFC 6960.

The OtherValues element provides a placeholder for other revocation information that can be used in the future. Their semantics and syntax are outside the scope of the standard.

NOTE — If XML electronic time-stamps based in XMLDSIG are standardized and spread, this type can also serve to contain the values of revocation data including CRLs and OCSP responses for any TSUs providing such electronic time-stamps, if they are not already present in the electronic time-stamps themselves as part of the TSUs' signatures. In this case, an element of this type can be added as an unsigned property to the XML electronic time-stamp using the incorporation mechanisms defined in the standard.

5.4.3 *The AttrAuthoritiesCertValues Qualifying Property*

Semantics

The AttrAuthoritiesCertValues qualifying property shall be an unsigned qualifying property that qualifies the signature.

The AttrAuthoritiesCertValues qualifying property:

a) shall contain the value(s) of the signing certificate(s) of the attribute certificate(s) and signed assertion(s) incorporated into the XAdES signature;

b) shall contain, if not present within the signature, the value(s) of the certificate(s) for the trust anchor(s) if such certificates exist, and the CA certificate values within path of the signing certificate(s) of the attribute

certificate(s) and signed assertion(s) incorporated into the XAdES signature. Certificate values present within the signature should not be included; and

c) may contain the certificate values used to sign CRLs or OCSP responses and the certificates values within their respective certificate paths, used for validating the signing certificate(s) of the attribute certificate(s) and signed assertion(s) incorporated into the XAdES signature. Certificate values present within the signature, including certificate values within the revocation status information themselves should not be included.

Syntax

The AttrAuthoritiesCertValues qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in D-1 of Annex D and is copied below for information.

<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="AttrAuthoritiesCertValues" type="CertificateValuesType"/>

5.4.4 *The AttributeRevocationValues Qualifying Property*

Semantics

The AttributeRevocationValues qualifying property shall be an unsigned qualifying property that qualifies the signature.

The AttributeRevocationValues qualifying property:

a) shall contain the revocation value(s) of the certificate(s) that sign the attribute certificate(s) and signed assertion(s) incorporated into the XAdES signature;

b) shall contain, if not incorporated into the signature, the revocation values corresponding to CA certificates within the path(s) of the signing certificate(s) of the attribute certificate(s) and signed assertion(s) incorporated into the XAdES signature. It shall not contain revocation values for the trust anchors. Values already incorporated into the signature should not be included; and

c) may contain the revocation values on certificates used to sign CRLs or OCSP responses and certificates within their respective certificate paths, which are used for validating the signing certificate(s) of the attribute certificate(s) and signed assertion(s) incorporated into the XAdES signature. Revocation values already incorporated into the signature should not be included.

Syntax

The AttributeRevocationValues qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in D-1 of Annex D and is copied below for information.

<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="AttributeRevocationValues" type="RevocationValuesType"/>

If the validation data contain one or more Delta CRLs, this qualifying property shall include the set of CRLs required to provide complete revocation lists.

5.5 Qualifying Properties For Long Term Availability And Integrity Of Validation Material

5.5.1 *The TimeStampValidationData Qualifying Property*

5.5.1.1 *Semantics and syntax*

Semantics

The TimeStampValidationData qualifying property shall be an unsigned qualifying property qualifying the signature.

The TimeStampValidationData qualifying property shall be a container for validation data required for carrying a full verification of the electronic time-stamps embedded within any of the different electronic time-stamp container qualifying properties defined in the standard.

The TimeStampValidationData qualifying property shall allow incorporating certificate values.

The TimeStampValidationData qualifying property shall allow incorporating revocation values.

Syntax

The `TimeStampValidationData` qualifying property shall be defined as in XML Schema file "XAdES01903v141-202107.xsd", whose location is detailed in D-2 and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.4.1#"

The preamble of the XML Schema file also includes the following
namespace declaration:
  xmlns:xades="http://uri.etsi.org/01903/v1.3.2#",
which assigns the prefix "xades" to the namespace whose URI is shown
in the declaration.
-->

<xsd:element                              name="TimeStampValidationData"
type="ValidationDataType"/>

<xsd:complexType name="ValidationDataType">
   <xsd:sequence>
      <xsd:element ref="xades:CertificateValues" minOccurs="0"/>
      <xsd:element ref="xades:RevocationValues" minOccurs="0"/>
   </xsd:sequence>
   <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
   <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
</xsd:complexType>
```

The CertificateValues child element shall contain certificates used in the full verification of electronic time-stamps embedded in one XAdES time-stamp container.

The CertificateValues child element may contain all the certificates required for a full verification of the electronic time-stamps.

The CertificateValues child element may also contain only some of the certificate values if the rest are present elsewhere in the XAdES signature (for instance within the electronic time-stamp itself, or in other TimeStampValidationData created for other electronic time-stamps).

The RevocationValues child element shall contain revocation values used in the full verification of electronic time-stamps embedded in one XAdES time-stamp container.

The RevocationValues child element may contain all the revocation values required for a full verification of the electronic time-stamps.

The RevocationValues child element may also contain only some of the revocation values if the rest are present elsewhere in the XAdES signature (for instance within the electronic time-stamp itself, or in other TimeStampValidationData created for other electronic time-stamps).

The Id attribute shall be used for referencing this element from elsewhere.

The TimeStampValidationData qualifying property may have the URI attribute. This attribute shall be used for referencing the time-stamp container of the electronic time-stamps whose validation data is contained within this element.

5.5.1.2 *Use of URI attribute*

If a XAdES signature requires including all the validation data required for a full verification of an electronic time-stamp embedded in a `SignatureTimeStamp`, `a RefsOnlyTimeStampV2`, `a SigAndRefsTimeStampV2`, or an `ArchiveTimeStamp` qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.4.1#, and part of that validation data is not present in other parts of the signature, then:

a) A new `TimeStampValidationData` qualifying property shall be created containing the missing validation data;

b) If direct incorporation as specified in **Error! Reference source not found.** of the standard is used, the `TimeStampValidationData` qualifying property shall be added as a child of `UnsignedSignatureProperties` element immediately after the respective electronic time-stamp container element;

c) If the `TimeStampValidationData` qualifying property is not incorporated immediately after the respective electronic time-stamp container element (this may only happen when indirect incorporation of properties as specified in **4.4** is used), then the URI attribute shall be present and shall reference the specific container encapsulating the electronic time-stamps whose validation data the new `TimeStampValidationData` qualifying property will contain; otherwise the URI attribute should not be present.

NOTE - In the case that the TimeStampValidationData qualifying property is incorporated immediately after the respective electronic time-stamp container element, the URI attribute is not needed because the identification of the related electronic time-stamp container is implicit in the relative position of both elements, the qualifying property containing the electronic time-stamp and the TimeStampValidationData qualifying property.

If a XAdES signature requires including all the validation data required for a full verification of an electronic time-stamp embedded in any of the following qualifying properties containing electronic

time-stamps: IndividualDataObjectsTimeStamp or AllDataObjectTimeStamp, and part of that validation data is not present in other parts of the signature, then:

a) A new TimeStampValidationData qualifying property shall be created containing the missing validation data;

b) It shall be added as the first child of the UnsignedSignatureProperties element; and

c) The URI attribute element shall be present and shall reference the specific signed container encapsulating the electronic time-stamps whose validation data the new TimeStampValidationData qualifying property will contain.

1 The treatment is different than for the other electronic time-stamp containers because first, there can be more than one signed electronic time- stamp container, and second IndividualDataObjectsTimeStamp and AllDataObjectTimeStamp are signed qualifying properties whereas the corresponding TimeStampValidationData qualifying properties are unsigned and they appear as children of different parents.

When validating a XAdES signature, in the case that the TimeStampValidationData qualifying property appears immediately after one of the following qualifying properties containing electronic time-stamps: SignatureTimeStamp, RefsOnlyTimeStampV2, SigAndRefsTimeStampV2, and ArchiveTimeStamp qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.4.1#, regardless of whether direct or indirect incorporation is used, if the value in URI attribute does not reference the electronic time-stamps container that precedes the TimeStampValidationData qualifying property, this URI attribute's value shall be ignored.

When validating a XAdES signature, in the case that the TimeStampValidationData qualifying property does not appear immediately after one qualifying property containing electronic time-stamps, if the value of URI attribute does not reference a qualifying property containing electronic time-stamps, then the application shall ignore this value and may try to use the validation material within the TimeStampValidationData in the validation of the different electronic time-stamps incorporated to the signature.

2 This behaviour is coherent with the fact of considering the value of the URI attribute as a hint that may help signature validation applications to establish a correspondence between the signing certificate of a certain electronic time-stamp and part of its validation data. However, applications can also establish this correspondence without the help of such an URI, and consequently, if the value of the URI attribute is wrong, still the material may correspond to some of the electronic time-stamps incorporated into the signature, and signature validation applications would be able to successfully use that material in the validation of the aforementioned electronic time-stamps.

5.5.2 *The ArchiveTimeStamp Qualifying Property Defined in Namespace with URI "http://uri.etsi.org/01903/v1.4.1#"*

5.5.2.1 *Semantics and syntax*
Semantics

The `ArchiveTimeStamp` qualifying property shall be an unsigned qualifying property qualifying the signature.

The `ArchiveTimeStamp` qualifying property shall encapsulate electronic time-stamps computed on all the data objects incorporated into the XAdES signature at the time of generating each electronic time-stamp.

NOTE - The purpose of this element is to tackle the long term availability and integrity of the validation material.

Syntax

The ArchiveTimeStamp qualifying property shall be defined as in XML Schema file XAdES01903v141-202107.xsd, whose location is detailed in The file at https://uri.etsi.org/01903/v1.3.2/XAdES01903v132-201601.xsd (XAdES01903v132-201601.xsd) contains the definitions of qualifying properties defined within the namespace whose URI value is http://uri.etsi.org/01903/v1.3.2#.

D-2, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.4.1#"

The preamble of the XML Schema file also includes the following namespace
declaration:
  xmlns:xades="http://uri.etsi.org/01903/v1.3.2#",
which assigns the prefix "xades" to the namespace whose URI is shown in
the declaration.
-->

<xsd:element name="ArchiveTimeStamp" type="xades:XAdESTimeStampType"/>
```

If the XAdES signature incorporates a CounterSignature unsigned qualifying property, all the required material for conducting the validation of the counter-signature shall be incorporated into the XAdES signature before generating the first ArchiveTimeStamp qualifying property. This may be done within the counter-signature itself or within the containers available within the counter-signed XAdES signature.

The contents of the CounterSignature qualifying property should not be changed, once it has been time-stamped by the ArchiveTimeStamp.

1 If a CounterSignature qualifying unsigned property is time-stamped by the ArchiveTimeStamp, any ulterior change of their contents (by addition of unsigned qualifying properties if the counter-signature is a XAdES signature, for instance) would make the validation of the ArchiveTimeStamp and, in consequence, the validation of the countersigned XAdES signature, fail.

2 Under these circumstances, the detached counter-signature mechanism specified in **5.2.7.1** can be used.

3 The standard permits counter-signing a previously time-stamped countersignature with another CounterSignature qualifying property added to the embedding XAdES signature after the time-stamp container.

4 Once an ArchiveTimeStamp qualifying property is added to the signature, any ulterior addition of a ds:Object to the signature would make the verification of such time-stamp fail.

Depending whether all the unsigned qualifying properties time-stamped by the electronic time-stamp and the ArchiveTimeStamp qualifying property itself have the same parent or not, its contents may be different. Details are given in **5.5.2.2** and **5.5.2.3**.

5.5.2.2 *Not distributed case*

If `ArchiveTimeStamp` and all the unsigned qualifying properties time-stamped by its electronic time-stamp have the same parent, this qualifying property shall use the implicit mechanism for identifying all the time-stamped data objects.

The input to the electronic time-stamp's message imprint computation shall be built following the algorithm specified below:

a) If the XAdES signature contains one or more signed `ds:Manifest` referencing data objects that are detached from the signature, and some of the digest algorithms indicated within `ds:Reference` children are known to be near the end of its recommended period of use, incorporate a new `RenewedDigestsV2` qualifying property as specified in Error! Reference source not found. of the standard. The value of its `ds:DigestMethod` child element shall be the identifier of a stronger digest algorithm, and it shall contain as many `RecomputedDigestValue` children as `ds:Reference` elements within the aforementioned signed `ds:Manifest` elements with digest algorithms known to be near the end of its recommended period of use.

b) Initialize the final octet stream as an empty octet stream.

c) Take all the `ds:Reference` elements in their order of appearance within `ds:SignedInfo` referencing whatever the signer wants to sign including the `SignedProperties` element. Process each one as indicated below:

1) Retrieve the data object referenced by the URI attribute of the `ds:Reference` element, as specified in **4.4.3.2** of XMLDSIGF.

NOTE - **4.4.3.2** of XMLDSIG, specifies rules for URI dereferencing. For instance, it mandates that the dereferencing of a non 'same-document' reference (which XMLDSIG [**Error! Reference source not found.**] defines as "a URI-Reference that consists of a hash sign ('#') followed by a fragment or alternatively consists of an empty URI") is always an octet-stream.

2) If the `ds:Reference` element does not contain the `ds:Transforms` element, then:

i) if the retrieved data object is an XML node-set, then canonicalize it as specified in clause 4.5 of the present document;

ii) else proceed to step d).

3) If the ds:Reference element contains the ds:Transforms element, then apply all the transforms indicated within the ds:Transform children elements. After that:

i) if the output of the last transform is a XML node-set according to XMLDSIG, canonicalize it as specified in Error! Reference source not found. of the present document;

ii) else proceed to step d).

4) Concatenate the resulting octets to the final octet stream.

d) Take the following XMLDSIG elements in the order they are listed below, canonicalize each one as specified in Error! Reference source not found., and concatenate each resulting octet stream to the final octet stream:

1) The `ds:SignedInfo` element;
2) The `ds:SignatureValue` element; and
3) The `ds:KeyInfo` element, if present.

e) Take the unsigned signature qualifying properties that appear before the current `ArchiveTimeStamp` in the order they appear within the `UnsignedSignatureProperties`, canonicalize each one as specified in **4.5**, and concatenate each resulting octet stream to the final octet stream. While concatenating, the following rules apply:

1) the `CertificateValues` qualifying property shall be incorporated into the signature if it is not already present and the signature misses some of the certificates listed in **5.4.1** that are required to validate the XAdES signature;

2) the `RevocationValues` qualifying property shall be incorporated into the signature if it is not already present and the signature misses some of the revocation data listed in **5.4.2** that are required to validate the XAdES signature;

3) the `AttrAuthoritiesCertValue`s qualifying property shall be incorporated into the signature if not already present and the following conditions are true: attribute certificate(s) or signed assertions have been incorporated into the signature, and the signature misses some certificates required for their validation; and

4) the `AttributeRevocationValues` qualifying property shall be incorporated into the signature if not already present and the following conditions are true: attribute certificates or signed assertions have been incorporated into the signature, and the signature misses some revocation values required for their validation.

f) Take all the ds:Object elements except the one containing QualifyingProperties element, in their order of appearance. Canonicalize each one as specified in **4.5**, and concatenate each resulting octet stream to the final octet stream.

5.5.2.3 *Distributed case*

If `ArchiveTimeStamp` and some of the unsigned qualifying properties covered by some of its electronic time-stamps do not have the same parent, the explicit (based on `Include` elements) mechanism shall be used only for referencing the unsigned qualifying properties.

One `Include` element shall be generated for each unsigned qualifying property already incorporated into the XAdES signature, which will consequently be time-stamped by the electronic time-stamp.

These `Include` elements shall be added in the same order as the unsigned qualifying properties are processed for contributing to the input of the electronic time-stamp's message imprint computation.

No `Include` elements shall be generated for any other XMLDSIG element present in the signature.

The input to the electronic time-stamp's message imprint computation shall be built as for the not distributed case (Error! Reference source not found.) substituting its step 5 by the one specified below:

a) Take the unsigned signature qualifying properties present in the signature, extract comment nodes, canonicalize each unsigned signature qualifying property as specified in Error! Reference source not found., and concatenate each resulting octet stream to the final octet stream. While concatenating, the following rules apply:

1) the `CertificateValues` qualifying property shall be incorporated into the signature if it is not already present and the signature misses some of the certificates listed in Error! Reference source not found. that are required to validate the XAdES signature;

2) the `RevocationValues` qualifying property shall be incorporated into the signature if it is not already present and the signature misses some of the revocation data listed in Error! Reference source not found. that are required to validate the XAdES signature;

3) the `AttrAuthoritiesCertValues` qualifying property shall be incorporated into the signature if not already present and the following conditions are true: attribute certificate(s) or signed assertions have been incorporated into the signature, and the signature misses some certificates required for their validation; and

4) If `ArchiveTimeStamp` and some of the unsigned qualifying properties covered by some of its electronic time-stamps do not have the same parent, the explicit (based on `Include` elements) mechanism shall be used only for referencing the unsigned qualifying properties.

   One `Include` element shall be generated for each unsigned qualifying property already incorporated into the XAdES signature, which will consequently be time-stamped by the electronic time-stamp.

   These `Include` elements shall be added in the same order as the unsigned qualifying properties are processed for contributing to the input of the electronic time-stamp's message imprint computation.

   No `Include` elements shall be generated for any other XMLDSIG element present in the signature.

   The input to the electronic time-stamp's message imprint computation shall be built as for the not distributed case (Error! Reference source not found.) substituting its step 5 by the one specified below:

b) Take the unsigned signature qualifying properties present in the signature, extract comment nodes, canonicalize each unsigned signature qualifying property as specified in Error! Reference source not found., and concatenate each resulting octet stream to the final octet stream. While concatenating, the following rules apply:

   1) the `CertificateValues` qualifying property shall be incorporated into the signature if it is not already present and the signature misses some of the certificates listed in Error! Reference source not found. that are required to validate the XAdES signature;

   2) the `RevocationValues` qualifying property shall be incorporated into the signature if it is not already present and the signature misses some of the revocation data listed in Error! Reference source not found. that are required to validate the XAdES signature;

   3) the `AttrAuthoritiesCertValues` qualifying property shall be incorporated into the signature if not already present and the following conditions are true: attribute certificate(s) or signed assertions have been incorporated into the signature, and the signature misses some certificates required for their validation; and

5.5.3 *The RenewedDigestsV2 Qualifying Property*

Semantics

The RenewedDigestsV2 qualifying property is an unsigned qualifying property qualifying the signature.

The RenewedDigestsV2 qualifying property may be used when the XAdES signature contains one or more signed ds:Manifest referencing data objects that are detached from the signature.

The RenewedDigestsV2 qualifying property shall not be used if the XAdES signature does not contain any signed ds:Manifest referencing data objects that are detached from the signature.

The RenewedDigestsV2 qualifying property shall contain the digest values of the aforementioned indirectly signed detached data objects computed with a different digest algorithm than the one used for computing the digest values present within the ds:Manifest's ds:Reference children.

When a certain digest algorithm is becoming weak, one or more detached data objects have been indirectly signed using that algorithm with a signed ds:Manifest, when long term signatures are managed using ArchiveTimeStamp qualifying property, and when suspected that some of the aforementioned data objects might be substituted by others with the same digest value due to the weakness of the digest algorithm, the RenewedDigestsV2 element shall be incorporated into the XAdES signature including the digest values of the aforementioned data objects computed with a different algorithm.

NOTES

1 This will ensure that any substitution of a detached document indirectly signed through a signed ds:Manifest, will be detected even if the digest algorithm used for computing the references within the aforementioned ds:Manifest, has been broken.

Whenever it is detected that the digest algorithm used for building a RenewedDigestsV2 element is becoming weak, a new RenewedDigestsV2 element shall be generated incorporating digest values of the detached signed data objects computed with different algorithms.

Syntax

The RenewedDigestsV2 qualifying property shall be defined as in XML Schema file "XAdES01903v141-202107.xsd", whose location is detailed in Clause D-2 of Annex D, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.4.1#" -->

<xsd:element name="RenewedDigestsV2" type="RenewedDigestsV2Type"/>

<xsd:complexType name="RenewedDigestsV2Type">
        <xsd:sequence>
                <xsd:element ref="ds:CanonicalizationMethod"/>
                <xsd:element ref="ds:DigestMethod"/>
                <xsd:element ref="RecomputedDigestValue" maxOccurs="unbounded"/>
        </xsd:sequence>
        <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

```
<xsd:element name="RecomputedDigestValue" type="RecomputedDigestValueType"/>

<xsd:complexType name="RecomputedDigestValueType">
        <xsd:sequence>
                <xsd:element name="NewSDODigestValue" type="ds:DigestValueType"/>
                <xsd:element name="OriginalRefDigest" type="ds:DigestValueType"/>
        </xsd:sequence>
</xsd:complexType>
```

The ds:CanonicalizationMethod child shall identify a canonicalization algorithm.

The ds:DigestMethod child shall identify the digest algorithm used for recomputing the digest values of the detached signed data objects referenced by ds:Reference elements children of signed ds:Manifest elements.

RecomputedDigestValue's children elements satisfy the following requirements:

a) The NewSDODigestValue child shall contain the base-64 encoded digest value, computed using the algorithm identified in the RenewedDigestsV2's ds:DigestMethod child element, of one signed detached object. This digest value shall be computed on the result of processing, as specified by the reference processing model defined in **4.4.3.2** of XMLDSIG, the ds:Reference element, child of one of the signed ds:Manifest elements, referencing the aforementioned detached signed data object.

b) The OriginalRefDigest child shall contain the base-64 encoded digest value of the canonicalized ds:Reference element, child of one of the signed ds:Manifest elements, referencing the detached signed data object whose recomputed value is placed in the NewSDODigestValue element. The ds:Reference shall be canonicalized using the canonicalization algorithm identified in the RenewedDigestsV2's ds:CanonicalizationMehtod child element. The digest value shall be computed using the algorithm identified in the RenewedDigestsV2's ds:DigestMethod child element.

The content of the RecomputedDigestValue 's OriginalRefDigest shall be computed as indicated below:

a) Take the ds:Reference child of the ds:Manifest element referencing to the detached signed data object whose new digest value has been placed within RecomputedDigestValue's NewSDODigestValue child, and canonicalize it using the canonicalization algorithm present in the RenewedDigestsV2's ds:CanonicalizationMethod child.

b) Compute the digest of the canonicalized ds:Reference using the digest algorithm identified in the RenewedDigestsV2's ds:DigestMethod child.

c) Base-64 encode the resulting digest value and place it within the RecomputedDigestValue 's OriginalRefDigest child.

When validating the signature, each RenewedDigestsV2 qualifying property incorporated to the XAdES signature shall be processed as follows:

d) Take the first RecomputedDigestValue child element.

e) Retrieve the ds:Reference element referenced by the OriginalRefDigest child of the taken RecomputedDigestValue element.

f) Retrieve the data object referenced by the aforementioned ds:Reference element and process its ds:Transforms child according to the reference-processing model of XMLDSIG, **4.4.3.2**. If it is not possible to retrieve the referenced octets, then notify that it has not been possible to successfully retrieve the detached signed data object referenced by the ds:Reference element identified in the previous step, and continue with step f).

2 **4.4.3.2** of XMLDSIG specifies that the last step of the ds:Reference element is the computation of the digest of the retrieved and transformed signed data object with an algorithm identified in the ds:Reference's ds:DigestMethod child element. This computation is not performed in step c).

g) Compute the digest value of the resulting octets using the digest algorithm identified in the RenewedDigestsV2's ds:DigestMethod child element.

3 In step d), the retrieved and transformed signed data object is digested using the digest algorithm identified within RenewedDigestsV2's ds:DigestMethod child element, which is different than the algorithm identified within the ds:DigestMethod child of the ds:Reference element.

h) If the digest value computed in step d) is different from the digest value present within the NewSDODigestValue element, then notify that there is a mismatch in the digest value of the detached signed data object referenced by the ds:Reference element identified in step b).

4 For comparing the aforementioned digest values, it needs to be considered that the content of NewSDODigestValue element is a base-64 encoded digest value.

i) If there are other RecomputedDigestValue elements that have not been yet processed, take the next RecomputedDigestValue element and go to step b); else finalize the process.

6 XAdES BASELINE SIGNATURES

6.1 Signature Levels

Clause 6 defines four levels of XAdES baseline signatures, intended to facilitate interoperability and to encompass the life cycle of XAdES signature, namely:

a) B-B level provides requirements for the incorporation of signed and some unsigned qualifying properties when the signature is generated.

b) B-T level provides requirements for the generation and inclusion, for an existing signature, of a trusted token proving that the signature itself actually existed at a certain date and time.

c) B-LT level provides requirements for the incorporation of all the material required for validating the signature in the signature document. This level aims to tackle the long term availability of the validation material.

d) B-LTA level provides requirements for the incorporation of electronic time-stamps that allow validation of the signature long time after its generation. This level aims to tackle the long term availability and integrity of the validation material.

NOTES

1 ETSI TR 119 100 provides a description on the life-cycle of a signature and the rationales on which level is suitable in which situation.

2 The levels c) to d) are appropriate where the technical validity of signature needs to be preserved for a period of time after signature creation where certificate expiration, revocation and/or algorithm obsolescence is of concern. The specific level applicable depends on the context and use case.

3 B-LTA level targets long term availability and integrity of the validation material of digital signatures over long term. The B-LTA level can help to validate the signature beyond many events that limit its validity (for instance, the weakness of used cryptographic algorithms, or expiration of validation data). The use of B-LTA level is considered an appropriate preservation and transmission technique for signed data.

4 Conformance to B-LT level, when combined with appropriate additional preservation techniques tackling the long term availability and integrity of the validation material is sufficient to allow validation of the signature long time after its generation. The assessment of the effectiveness of preservation techniques for signed data other than implementing the B-LTA level are out of the scope of the standard. The reader is advised to consider legal instruments in force and/or other standards (for example ETSI TS 119 511, or IETF RFC 4998, or IETF RFC 6283) or that can indicate other preservation techniques. Annex D defines what needs to be taken into account when using other techniques for long term availability and integrity of validation data and incorporating a new unsigned property derived from these techniques into the signature.

6.2 General Requirements

6.2.1 *Algorithm Requirement*

The algorithms and key lengths used to generate and augment digital signatures should be as specified in ETSI TS 119 312.

NOTES

1 Cryptographic suites recommendations defined in ETSI TS 119 312 can be superseded by national recommendations.
2 IETF RFC 6931 defines a set of additional XML security URIs, which complement those ones defined in XMLDSIG.

In addition, MD5 algorithm shall not be used as digest algorithm.

6.2.2 *Notation For Requirements*

The present clause describes the notation used for defining the requirements of the different XAdES signature levels.

The requirements on the qualifying properties and certain other signature's elements for each XAdES signature level are expressed in Table 2Error! Reference source not found.**.** A row in the table either specifies requirements for a qualifying property, other signature's element, or a service.

A service can be provided by different qualifying properties, by other signature's elements, or by other mechanisms (service provision options hereinafter). In these cases, the specification of the requirements for a service is provided by three or more rows. The first row contains the requirements of the service. The requirements for the qualifying properties, other signature's elements, and/or mechanisms used to provide the service are stated in the following rows.

Table 2 contains 8 columns. Below follows a detailed explanation of their meanings and contents:

a) Column "Elements/Qualifying properties/Services":

   1) In the case where the cell identifies a Service, the cell content starts with the keyword "Service" followed by the name of the service.

2) In the case where the qualifying property or other signature's element provides a service, this cell contains "SPO" (for Service Provision Option), followed by the name of the qualifying property or the other signature's element.

3) Otherwise, this cell contains the name of the qualifying property or the other signature's element.

b) Column "Presence in B-B-Level": This cell contains the specification of the presence of the qualifying property or other signature's element, or the provision of a service, for XAdES-B-B signatures.

c) Column "Presence in B-T level": This cell contains the specification of the presence of the qualifying property or other signature's element, or the provision of a service, for XAdES-B-T signatures.

d) Column "Presence in B-LT level": This cell contains the specification of the presence of the qualifying property or other signature's element, or the provision of a service, for XAdES-B-LT signatures.

e) Column "Presence in B-LTA level": This cell contains the specification of the presence of the qualifying property or other signature's element, or the provision of a service, for XAdES-B-LTA signatures. Below follow the values that can appear in columns "Presence in B-B", "Presence in B-T", "Presence in B-LT", and "Presence in B-LTA":

- "shall be present": means that the qualifying property or signature's element shall be incorporated to the signature, and shall be as specified in the document referenced in column "References", further profiled with the additional requirements referenced in column "Requirements", and with the cardinality indicated in column "Cardinality".

- "shall not be present": means that the qualifying property or signature's element shall not be incorporated to the signature.

- "may be present": means that the qualifying property or signature's element may be incorporated to the signature, and shall be as specified in the document referenced in column "References", further profiled with the additional requirements referenced in column "Requirements", and with the cardinality indicated in column "Cardinality".

- "shall be provided": means that the service identified in the first column of the row shall be provided as further specified in the SPO-related rows. This value only appears in rows that contain requirements for services. It does not appear in rows that contain requirements for qualifying properties or signature's elements.

- "conditioned presence": means that the incorporation to the signature of the item identified in the first column is conditioned as per the requirements referenced in column "Requirements" and requirements in specifications and clauses referenced by column "References", with the cardinality indicated in column "Cardinality".

- "*": means that the qualifying property or signature's element (service) identified in the first column should not be incorporated to the signature (provided) in the corresponding level. Upper signature levels may specify other requirements.

NOTE — Incorporating an unsigned property that is marked with a "*" into a signature can lead to cases where a higher level cannot be achieved, except by removing the corresponding unsigned property.

f) Column "Cardinality": This cell indicates the cardinality of the qualifying property or other signature's element. If the cardinality is the same for all the levels, only the values listed below appear. Otherwise the content specifies the cardinality for each level. See the example at the end of the present clause showing this situation. Below follow the values indicating the cardinality:

- **0:** The signature shall not incorporate any instance of the qualifying property or the signature's element.
- **1:** The signature shall incorporate exactly one instance of the qualifying property or the signature's element.
- **0 or 1:** The signature shall incorporate zero or one instance of the qualifying property or the signature's element.
- **≥ 0:** The signature shall incorporate zero or more instances of the qualifying property or the signature's element.
- **≥ 1:** The signature shall incorporate one or more instances of the qualifying property or the signature's element.

g) Column "References": This shall contain either the number of the clause specifying the qualifying property in the standard, or a reference to the document and clause that specifies the other signature's element.

h) Column "Additional notes and requirements": This cell contains numbers referencing notes and/or letters referencing additional requirements on the qualifying property or the other signature's element. Both notes and additional requirements are listed below the Table.

Names of XML elements in the namespace whose URI is http://www.w3.org/2000/09/xmldsig# are preceded by prefix ds.

EXAMPLE:

In Table 2, the row corresponding to CompleteCertificateRefsV2 qualifying property has a value "*" in the cells in columns "Presence in B-B level" and "Presence in B-T level", and "shall not be present" in cells in columns "Presence in B-LT level" and "Presence in B-LTA level". The cell in column "Cardinality" indicates the cardinality for each level as follows: "B-B, B-T: 0 or 1" indicates that XAdES-B-B and XAdES-B-T signatures can incorporate one instance of CompleteCertificateRefsV2 qualifying property; "B-LT, B-LTA: 0" indicates that XAdES-B-LT and XAdES-B-LTA do not incorporate the CompleteCertificateRefsV2 qualifying property.

6.3 Requirements on XAdES Signature's Elements, Qualifying Properties and Services

The four XAdES signature levels specified in the present clause shall be built as specified in **4** The XAdES qualifying properties specified in **5** shall be incorporated into the signature using only the direct incorporation mechanism specified in **4.4**.

NOTES

1 This means that all the XAdES qualifying properties remain within one single QualifyingProperties element, which in turn is the child of one ds:Object element within the signature, and that in consequence, no QualifyingPropertiesReference element is present.

Table 2 shows the presence and cardinality requirements on the signature elements, qualifying properties, and services indicated in the first column for the four XAdES baseline signature levels, namely: XAdES-B-B, XAdES-B-T, XAdES-B-LT, and XAdES-B-LTA). Additional requirements are detailed below the Table suitably labelled with the letter indicated in the last column.

2 XAdES-B-B signatures that incorporate only the elements/qualifying properties that are mandatory in Table 2, and that implement the mandatory requirements, contain the lowest number of elements/qualifying properties, with the consequent benefits for interoperability.

In XAdES baseline signatures the qualifying properties that act as electronic time-stamps containers shall encapsulate only IETF RFC 3161 updated by IETF RFC 5816 electronic time-stamps.

Table 1 Requirements For XAdES-B-B, XAdES-B-T, XAdES-B-LT, And XAdES-B-LTA Signatures

(Clause 6.3*)*

Sl. No. (1)	Elements/Qualifying properties/Services (2)	Presence in B-B level (3)	Presence in B-T level (4)	Presence in B-LT level (5)	Presence in B-LTA level (6)	Cardinality (7)	References (8)	Additional requirements and notes (9)
i)	ds:KeyInfo/X509Data	shall be present	shall be present	shall be present	shall be present	1	XMLDSIG , **4.5.4**	a, b, c 3, 4, 5

Sl. No. (1)	Elements/Qualifying properties/Services (2)	Presence in B-B level (3)	Presence in B-T level (4)	Presence in B-LT level (5)	Presence in B-LTA level (6)	Cardinality (7)	References (8)	Additional requirements and notes (9)
ii)	ds:SignedInfo/ds:CanonicalizationMethod	shall be present	shall be present	shall be present	shall be present	1	XMLDSIG , **4.4.1**	d, e 6
iii)	ds:Reference	shall be present	shall be present	shall be present	shall be present	≥ 2	XMLDSIG , **4.4.3**	
iv)	ds:Reference/ds:Transforms	may be present	may be present	may be present	may be present	0 or 1	XMLDSIG , **4.4.3.4**	f, g
v)	SigningTime	shall be present	shall be present	shall be present	shall be present	1	**5.2.1**	h
vi)	SigningCertificateV2	shall be present	shall be present	shall be present	shall be present	1	**5.2.2**	, i, k 7
vii)	SigningCertificate	shall not be present	shall not be present	shall not be present	shall not be present	0	-	
viii)	DataObjectFormat	conditioned presence	conditioned presence	conditioned presence	conditioned presence	≥ 0	**5.2.4**	m
ix)	DataObjectFormat/Description	may be present	may be present	may be present	may be present	0 or 1	**5.2.4**	n 8
x)	DataObjectFormat/ObjectIdentifier	may be present	may be present	may be present	may be present	0 or 1	**5.2.4**	n
xi)	DataObjectFormat/MimeType	shall be present	shall be present	shall be present	shall be present	1	**5.2.4**	n
xii)	DataObjectFormat/Encoding	may be present	may be present	may be present	may be present	0 or 1	**5.2.4**	n

Sl. No. (1)	Elements/Qualifying properties/Services (2)	Presence in B-B level (3)	Presence in B-T level (4)	Presence in B-LT level (5)	Presence in B-LTA level (6)	Cardinality (7)	References (8)	Additional requirements and notes (9)
xiii)	DataObjectFormat's ObjectReference attribute	shall be present	shall be present	shall be present	shall be present	1	**5.2.4**	n
xiv)	SignerRole	shall not be present	shall not be present	shall not be present	shall not be present	0	-	
xv)	SignerRoleV2	may be present	may be present	may be present	may be present	0 or 1	**5.2.6**	
xvi)	CommitmentTypeIndication	may be present	may be present	may be present	may be present	≥ 0	**5.2.3**	
xvii)	SignatureProductionPlaceV2	may be present	may be present	may be present	may be present	0 or 1	**5.2.5**	
xviii)	SignatureProductionPlace	shall not be present	shall not be present	shall not be present	shall not be present	0	-	
xix)	CounterSignature	may be present	may be present	may be present	may be present	≥ 0	**5.2.7.2**	
xx)	AllDataObjectsTimeStamp	may be present	may be present	may be present	may be present	≥ 0	Error! Reference source not found.	10
xxi)	IndividualDataObjectsTimeStamp	may be present	may be present	may be present	may be present	≥ 0	Error! Reference source not found.	10
xxii)	SignaturePolicyIdentifier	may be present	may be present	may be present	may be present	0 or 1	**5.2.9**	
xxiii)	SignaturePolicyStore	conditioned presence	conditioned presence	conditioned presence	conditioned presence	0 or 1	**5.2.10**	p
xxiv)	SignatureTimeStamp	*	shall be present	shall be present	shall be present	B-B: ≥ 0	**5.3**	q, r

Sl. No. (1)	Elements/Qualifying properties/Services (2)	Presence in B-B level (3)	Presence in B-T level (4)	Presence in B-LT level (5)	Presence in B-LTA level (6)	Cardinality (7)	References (8)	Additional requirements and notes (9)
						B-T, B-LT, B-LTA: ≥ 1		10
xxv)	CertificateValues	*	*	conditioned presence	conditioned presence	0 or 1	Error! Reference source not found.	s, t
xxvi)	AnyValidationData	*	*	conditioned presence	conditioned presence	≥ 0	Error! Reference source not found.	t, y, z, ac
xxvii)	CompleteCertificateRefsV2	*	*	shall not be present	shall not be present	B-B, B-T: 0 or 1 B-LT, B-LTA: 0	**B.1.1**	k
xxviii	CompleteCertificateRefs	shall not be present	shall not be present	shall not be present	shall not be present	0	-	
xxix)	AttrAuthoritiesCertValues	*	*	conditioned presence	conditioned presence	0 or 1	**5.4.3**	t, u
xxx)	AttributeCertificateRefsV2	*	*	shall not be present	shall not be present	B-B, B-T: 0 or 1 B-LT, B-LTA: 0	**B.1.3**	k, v
xxxi)	AttributeCertificateRefs	shall not be present	shall not be present	shall not be present	shall not be present	0	-	
xxxii)	RevocationValues	*	*	conditioned presence	conditioned presence	0 or 1	**5.4.2**	w, y, z
xxxiii	CompleteRevocationRefs	*	*			B-B, B-T: 0 or 1	**B.1.2**	

Sl. No. (1)	Elements/Qualifying properties/Services (2)	Presence in B-B level (3)	Presence in B-T level (4)	Presence in B-LT level (5)	Presence in B-LTA level (6)	Cardinality (7)	References (8)	Additional requirements and notes (9)
				shall not be present	shall not be present	B-LT, B-LTA: 0		
xxxiv	AttributeRevocationValues	*	*	conditioned presence	conditioned presence	0 or 1	**5.4.4**Error! Reference source not found.	v,w
xxxv)	AttributeRevocationRefs	*	*	shall not be present	shall not be present	B-B, B-T: 0 or 1 B-LT, B-LTA: 0	**B.1.4**	s
xxxvi	SigAndRefsTimeStampV2	*	*	shall not be present	shall not be present	B-B, B-T: ≥ 0 B-LT, B-LTA: 0	**B.1.5.1**	
xxxvi	SigAndRefsTimeStamp	shall not be present	shall not be present	shall not be present	shall not be present	0	-	
xxxvi	RefsOnlyTimeStampV2	*	*	shall not be present	shall not be present	B-B, B-T: ≥ 0 B-LT, B-LTA: 0	**B.1.5.2**	
xxxix	RefsOnlyTimeStamp	shall not be present	shall not be present	shall not be present	shall not be present	0	-	
xl)	Service: Incorporation of validation data for electronic time-stamps	*	*	shall be provided	shall be provided	-	-	ab, ac

Sl. No. (1)	Elements/Qualifying properties/Services (2)	Presence in B-B level (3)	Presence in B-T level (4)	Presence in B-LT level (5)	Presence in B-LTA level (6)	Cardinality (7)	References (8)	Additional requirements and notes (9)
xli)	SPO: TimeStampValidationData	*	*	conditioned presence	conditioned presence	≥ 0	**5.5.1**	ac
xlii)	SPO: certificate and revocation values embedded in the electronic time-stamp itself	*	*	conditioned presence	conditioned presence	≥ 0	-	ac
xliii)	SPO: AnyValidationData	*	*	conditioned presence	conditioned presence	≥ 0		ac
xliv)	ArchiveTimeStamp (defined in namespace whose URI is "http://uri.etsi.org/01903/v1.4.1#")	*	*	*	shall be present	≥ 1	**5.5.2**	ad, ae
	ArchiveTimeStamp (defined in namespace whose URI is "http://uri.etsi.org/01903/v1.3.2#")	shall not be present	shall not be present	shall not be present	shall not be present	0	-	
	RenewedDigestsV2	*	*	*	conditioned presence	≥ 0	**5.5.3**	af

Additional requirements:

a) Requirement for ds:KeyInfo/X509Data. The generator shall include the signing certificate as content of ds:KeyInfo/X509Data/X509Certificate element.

b) Requirement for ds:KeyInfo/X509Data. In order to facilitate path-building, generators should include in the same ds:KeyInfo/X509Data element as in requirement a) all certificates not available to verifiers that can be used during path building.

c) Requirement for ds:KeyInfo/X509Data. If the signature is to be validated through Root Certifying Authority of India (RCAI) established by Controller of Certifying Authorities (CCA) as per provisions under Indian Information Technology (IT) Act, 2000 , then the generator should include all intermediary certificates forming a chain between the signing certificate and a CA present in the Trusted List, which are not available to verifiers.

d) Requirement for ds:SignedInfo/ds:CanonicalizationMethod element. The Algorithm attribute of ds:SignedInfo's ds:CanonicalizationMethod child element shall have one of the following values:

- "http://www.w3.org/2006/12/xml-c14n11". The corresponding canonicalization algorithm Canonical XML v1.1 (omits comments) shall be supported.

- "http://www.w3.org/2001/10/xml-exc-c14n#". The corresponding canonicalization algorithm Exclusive Canonicalization (omits comments) shall be supported.

- "http://www.w3.org/TR/2001/REC-xml-c14n-20010315". The corresponding canonicalization algorithm Canonical XML v1.0 (omits comments) shall be supported.

- "http://www.w3.org/2006/12/xml-c14n11#WithComments". The corresponding canonicalization algorithm Canonical XML v1.1 (with comments) shall be supported.

- "http://www.w3.org/2001/10/xml-exc-c14n#WithComments". The corresponding canonicalization algorithm Exclusive Canonicalization (with comments) shall be supported. Or

- "http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments". The corresponding canonicalization algorithm Canonical XML v1.0 (with comments) shall be supported.

e) Requirement for ds:SignedInfo/CanonicalizationMethod element. The generator should not use canonicalization algorithms "with comments". See note 6.

f) Requirement for ds:Reference/ds:Transforms element. If the transform indicated by a ds:Reference/ds:Transforms's ds:Transform child element is a canonicalization, its Algorithm attribute shall have one of the values listed in the present clause, additional requirement d) and the generator should not use canonicalization algorithms "with comments".

g) Requirement for ds:Reference/ds:Transforms element. If the transform indicated by a ds:Reference/ds:Transforms's ds:Transform child element is not a canonicalization, its Algorithm attribute should be one of the following values:

- "http://www.w3.org/2000/09/xmldsig#base64". The corresponding Base 64 transform, whose usage within XML signatures shall be supported.

- "http://www.w3.org/TR/1999/REC-xpath-19991116". The corresponding XPath transform, whose usage within XML signatures shall be supported.

- "http://www.w3.org/2000/09/xmldsig#enveloped-signature". The corresponding Enveloped Signature transform, whose usage within XML signatures shall be supported.

- "http://www.w3.org/TR/1999/REC-xslt-19991116". The corresponding XSLT transform, whose usage within XML signatures shall be supported.

- "http://www.w3.org/2002/06/xmldsig-filter2". The corresponding XML-Signature XPath Filter 2.0, shall be supported. or

- "http://schemas.openxmlformats.org/package/2006/RelationshipTransform". The corresponding Relationships transform, shall be supported.

h) Requirement for SigningTime. The generator shall include the claimed UTC time when the signature was generated as content of the SigningTime qualifying property.

j) Requirement for SigningCertificateV2/Cert. The generator shall not generate Cert children's URI optional attribute.

k) Requirement for SigningCertificateV2, CompleteCertificateRefsV2, and AttributeCertificateRefsV2. The references to certificates should not include the IssuerSerialV2 element.

m) Requirement for DataObjectFormat. One DataObjectFormat shall be generated for each signed data object, except the SignedProperties element, and except if the signature is a baseline signature countersigning a signature. If the signature is a baseline signature countersigning another signature, and if it only signs its own signed properties and the countersigned signature, then it shall not include any DataObjectFormat signed property. If the signature is a baseline signature countersigning another signature and if it signs its own signed properties, the countersigned signature, and other data object(s), then it shall include one DataObjectFormat signed property for each of these other signed data object(s) aforementioned.

n) Requirement for XML components within DataObjectFormat. The number of occurrences allowed of the concerned XML component within one DataObjectFormat element, shall be as indicated in column "Cardinality".

p) Requirement for SignaturePolicyStore. This qualifying property may be incorporated into the XAdES signature only if the SignaturePolicyIdentifier is also incorporated and it contains the SigPolicyHash element with the digest value of the signature policy document. Otherwise the SignaturePolicyStore shall not be incorporated into the XAdES signature.

q) Requirement for SignatureTimeStamp. Each SignatureTimeStamp element shall contain only one electronic time-stamp.

r) Requirement for SignatureTimeStamp. The electronic time-stamps encapsulated within the signature-time-stamp attributes shall be created before the signing certificate has been revoked or has expired

s) Requirement for incorporation of CertificateValues. If a XAdES-B-LT or a XAdES-B-LTA signature is generated, the incorporation of CertificateValues shall be determined by requirements specified in Error! Reference source not found., specified for computing the input to the electronic time-stamp's message imprint.

t) Requirement for CertificateValues, AttrAuthoritiesCertValues, and AnyValidationData. Duplication of certificate values within the signature should be avoided.

u) Requirement for incorporation of AttrAuthoritiesCertValues. The AttrAuthoritiesCertValues qualifying property may be used when a at least an attribute certificate or a signed assertion is incorporated into a XAdES-B-LT or a XAdES-B-LTA signature for incorporating the certificates enumerated in **5.5.2.2.** Otherwise, AttrAuthoritiesCertValues qualifying property shall not be used.

v) The AttributeCertificateRefsV2 and AttributeRevocationRefs qualifying properties may be used when a at least an attribute certificate or a signed assertion is incorporated into the XAdES signature. Otherwise, AttributeCertificateRefsV2 and AttributeRevocationRefs qualifying properties shall not be used.

w) Requirement for incorporation of RevocationValues. If a XAdES-B-LT or a XAdES-B-LTA signature is generated, the incorporation of RevocationValues may be used for incorporating the certificate status values data enumerated in **5.5.2.2**.

y) Requirement for RevocationValues and AnyValidationData. Certificate status values should be included within RevocationValues or AnyValidationData qualifying properties instead within ds:KeyInfo element.

z) Requirement for RevocationValues, AttributeRevocationValues, and AnyValidationData. Duplication of certificate status values within the signature should be avoided.

aa) Requirement for incorporation of AttributeRevocationValues. The AttributeRevocationValues qualifying property may be used when a at least an attribute certificate or a signed assertion is incorporated into a XAdES-B-LT or a XAdES-B-LTA signature for incorporating the certificate status values enumerated in **5.5.2.2**. Otherwise, AttributeRevocationValues qualifying property shall not be used.

bb) Requirement for service "incorporation of validation data for electronic time-stamps". The validation data for electronic time-stamps shall be present within the TimeStampValidationData qualifying property, or AnyValidationData qualifying property, or embedded in the electronic time-stamp itself.

cc) Requirement for service "incorporation of validation data for electronic time-stamps" and its three options. The validation data for electronic time-stamps should be included either in the TimeStampValidationData qualifying property, or the AnyValidationData qualifying property.

dd) Requirement for ArchiveTimeStamp defined in the namespace whose URI is "http://uri.etsi.org/01903/v1.4.1#". Each ArchiveTimeStamp qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.4.1# may contain more than one electronic time-stamp issued by different TSAs.

ee) Requirement for ArchiveTimeStamp defined in the namespace whose URI is "http://uri.etsi.org/01903/v1.4.1#". Before generating and incorporating a new ArchiveTimeStamp qualifying property defined in the namespace whose URI is "http://uri.etsi.org/01903/v1.4.1#", all the validation material required for validating the signed data in the XAdES signature shall be included. This validation material shall include all the certificates and all certificate status information (like CRLs or OCSP responses) required for:

1) validating the signing certificate;
2) validating the signing certificate of any countersignature incorporated into the signature;

3) validating any attribute certificate or signed assertion present in the signature; and
4) validating the signing certificate of any previous electronic time-stamp already incorporated into the signature within any XAdES electronic time-stamp container qualifying property (including any ArchiveTimeStamp defined in the namespace whose URI is "http://uri.etsi.org/01903/v1.4.1#").

ff) Requirement for RenewedDigestsV2. If the XAdES signature signs external data objects through a signed ds:Manifest, and some of the digest algorithms used for computing some of the digest values within the aforementioned ds:Manifest is suspected to become weak enough as to represent a threat, the RenewedDigest element should be used to counter this threat.

gg) Requirement for AnyValidationData. If a XAdES-B-LT or a XAdES-B-LTA signature is generated, the incorporation of AnyValidationData may be used for incorporating any missing certificate and any missing certificate status values required for validating the XAdES signature.

NOTES

1 On ds:KeyInfo/X509Data/X509Certificate. A certificate is considered available to the verifier if reliable information about its location is known and allows automated retrieval of the certificate (for instance through an Authority Info Access Extension or equivalent information present in a TSL).

2 Void.

3 On ds:KeyInfo/X509Data/X509Certificate. In the general case, different verifiers can have different trust parameters and can validate the signing certificate through different chains. Therefore, generators may not know which certificates will be relevant for path building. However, in practice, generators can often clearly identify such certificates. In this case, including them in the signature is a good practice, unless verifiers can automatically retrieve them.

4 On ds:SignedInfo/CanonicalizationMethod. Support of canonicalization algorithms "with comments" is for residual interoperability in the signature validation process.

5 On SigningCertificateV2. The presence of the signing certificate within ds:KeyInfo ensures a way to locate it (on the basis of digest equality with the value within SigningCertificateV2/CertDigest) within the signature.

6 On DataObjectFormat. **5.2.4** of the standard establishes that this signed property "qualifies one specific signed data object". This is done by forcing that ObjectReference attribute refers to a ds:Reference. However, the aforementioned clause does not mandate this ds:Reference to be a child of ds:SignedInfo; it actually could be a ds:Reference within a signed ds:Manifest, as the object referenced in this way is also a signed object.

7 On service "incorporation of validation data for electronic time-stamps": the incorporation of the validation material of the electronic time-stamps ensures that the XAdES signature actually contains all the validation material needed.

8 On SignatureTimeStamp, IndividualDataObjectsTimeStamp, AllDataObjectsTimeStamp: Several instances of these qualifying properties can be incorporated into the XAdES signature, coming from different TSAs.

ANNEX A
(*Clause 2*)

LIST OF REFERRED STANDARDS

IS No./Other Standards	*Title*
W3C® Recommendation (11 April 2013)	XML Signature Syntax and Processing Version 1.1
W3C® Recommendation Part 1 (28 October 2004)	XML Schema Part 1: Structures Second Edition.
W3C® Recommendation Part 2 (28 October 2004)	XML Schema Part 2: Datatypes Second Edition.
Recommendation ITU-T X.509	Information technology — Open Systems Interconnection — The Directory: Public-key and attribute certificate frameworks
W3C® Recommendation (26 November 2008)	Extensible Markup Language (XML) 1.0
IETF RFC 6960	X.509 Internet Public Key Infrastructure Online Certificate Status Protocol — OCSP
IETF RFC 3161	Internet X.509 Public Key Infrastructure Time Stamp Protocol (TSP)
IETF RFC 3061	A URN Namespace of Object Identifiers
W3C® Recommendation (15 March 2001)	Canonical XML Version 1.0
W3C® Recommendation (18 July 2002)	Exclusive XML Canonicalization Version 1.0
W3C® Recommendation (2 May 2008)	Canonical XML Version 1.1
IETF RFC 3986	Uniform Resource Identifier (URI): Generic Syntax
W3C® Recommendation (8 November 2002)	XML-Signature XPath Filter 2.0
ISO/IEC 29500-2:2021	Information technology — Document description and processing languages — Office Open XML File Formats — Part 2: Open Packaging Conventions
IETF RFC 5280	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
IETF RFC 5816	ESSCertIDv2 Update for IETF RFC 3161
IETF RFC 5035	Enhanced Security Services (ESS) Update: Adding CertID Algorithm Agility
IETF RFC 2045	Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies

ANNEX B
(Normative)

ADDITIONAL QUALIFYING PROPERTIES SPECIFICATION

B-1 QUALIFYING PROPERTIES FOR VALIDATION DATA

B-1.1 The CompleteCertificateRefs Qualifying Property

Semantics

The CompleteCertificateRefsV2 qualifying property shall be an unsigned qualifying property qualifying the signature.

The CompleteCertificateRefsV2 qualifying property:

a) Shall contain the reference to the certificate of the trust anchor if such certificate does exist, and the references to CA certificates within the signing certificate path.

b) Shall not contain the reference to the signing certificate.

c) May contain references to certificates in the path of the certificates used for signing the electronic time-stamps already incorporated into the signature when the CompleteCertificateRefsV2 unsigned property is incorporated, including references to the electronic time-stamps' signing certificates and references to certificates of trust anchors if such certificates do exist.

d) May contain references to the certificates used to sign CRLs or OCSP responses for certificates referenced by references in 1) and 3), and references to certificates within their respective certificate paths. And

e) Shall not contain references to CA certificates that pertain exclusively to the certificate paths of certificates used to sign attribute certificates or signed assertions within SignerRoleV2.

NOTES

1 The references to certificates exclusively used in the validation of attribute certificate or signed assertions are stored in the AttributeCertificateRefsV2 qualifying property (see B-1.3).

Syntax

The CompleteCertificateRefsV2 element shall be defined as in XML Schema file "XAdES01903v141-202107.xsd", whose location is detailed in D-2 of Annex D, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.4.1#"

The preamble of the XML Schema file also includes the following namespace declaration:
  xmlns:xades="http://uri.etsi.org/01903/v1.3.2#",
which assigns the prefix "xades" to the namespace whose URI is shown in the declaration.
-->

<xsd:element name="CompleteCertificateRefsV2" type="CompleteCertificateRefsTypeV2"/>
<xsd:complexType name="CompleteCertificateRefsTypeV2">
    <xsd:sequence>
        <xsd:element name="CertRefs" type="xades:CertIDListV2Type"/>
```

```
    </xsd:sequence>
    <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

The CertRefs element is of type xades:CertIDListV2Type , already defined in **5.2.2**.

If at least one of the following unsigned properties: CertificateValues, AttrAuthoritiesCertValues, AnyValidationData with a non empty CertificateValues child element, or the ArchiveTimeStamp defined in the namespace whose URI is http://uri.etsi.org/01903/v1.4.1#, is incorporated into the signature, all the certificates referenced in CompleteCertificateRefsV2 shall be present elsewhere in the signature.

NOTE - If XML electronic time-stamps based in XMLDSIG are standardized and spread, this type can also be used to contain references to the certification chain for any TSUs providing such electronic time-stamps. In this case, an element of this type can be added as an unsigned qualifying property to the XML electronic time-stamp using the incorporation mechanisms defined in the standard.

B-1.2 The CompleteRevocationRefs Qualifying Property

Semantics

The CompleteRevocationRefs qualifying property shall be an unsigned qualifying property that qualifies the signature.

The CompleteRevocationRefs qualifying property:

a) Shall contain a reference to a revocation value for the signing certificate.

b) Shall contain the references to the revocation values (for example CRLs or OCSP values) corresponding to CA certificates within the signing certificate path. It shall not contain references to revocation values for the trust anchor.

   NOTE - A trust anchor is by definition trusted, thus no revocation information for the trust anchor is used during the validation.

c) May contain references to revocation values (for exampleCRLs or OCSP values) corresponding to certificates in the path of signing certificates of electronic time-stamps already incorporated into the signature when the CompleteRevocationRefs unsigned property is incorporated. It shall not contain references to revocation values for the trust anchors of these certificates.

d) May contain references to the revocation values corresponding to certificates used to sign CRLs or OCSP responses referenced in references from a), b) and c), and to certificates within their respective certificate paths. And

e) Shall not contain references to the revocation values corresponding to CA certificates that pertain exclusively to the certificate paths of certificates used to sign attribute certificates or signed assertions within SignerRoleV2 qualifying properties.

   NOTE - The references to revocation values exclusively used in the validation of attribute certificate or signed assertions are stored in the AttributeRevocationRefs qualifying property (see **Error! Reference source not found.**).

References within CompleteRevocationRefs qualifying property may be references to CRLs, OCSP responses and other type of revocation data.

Syntax

The CompleteRevocationRefs qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in

D-1 XML SCHEMA FILE **LOCATION FOR NAMESPACE**, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="CompleteRevocationRefs"
 type="CompleteRevocationRefsType"/>

<xsd:complexType name="CompleteRevocationRefsType">
 <xsd:sequence>
  <xsd:element name="CRLRefs" type="CRLRefsType" minOccurs="0"/>
  <xsd:element name="OCSPRefs" type="OCSPRefsType" minOccurs="0"/>
  <xsd:element name="OtherRefs" type="OtherCertStatusRefsType"
   minOccurs="0"/>
 </xsd:sequence>
 <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>

<xsd:complexType name="CRLRefsType">
 <xsd:sequence>
  <xsd:element name="CRLRef" type="CRLRefType"
   maxOccurs="unbounded"/>
 </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CRLRefType">
 <xsd:sequence>
  <xsd:element name="DigestAlgAndValue"
   type="DigestAlgAndValueType"/>
  <xsd:element name="CRLIdentifier" type="CRLIdentifierType"
   minOccurs="0"/>
 </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CRLIdentifierType">
 <xsd:sequence>
  <xsd:element name="Issuer" type="xsd:string"/>
  <xsd:element name="IssueTime" type="xsd:dateTime" />
  <xsd:element name="Number" type="xsd:integer"  minOccurs="0"/>
 </xsd:sequence>
 <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
</xsd:complexType>

<xsd:complexType name="OCSPRefsType">
 <xsd:sequence>
  <xsd:element name="OCSPRef" type="OCSPRefType"
   maxOccurs="unbounded"/>
 </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="OCSPRefType">
```

```
  <xsd:sequence>
    <xsd:element name="OCSPIdentifier" type="OCSPIdentifierType"/>
    <xsd:element name="DigestAlgAndValue"
      type="DigestAlgAndValueType"
      minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="ResponderIDType">
  <xsd:choice>
    <xsd:element name="ByName" type="xsd:string"/>
    <xsd:element name="ByKey" type="xsd:base64Binary"/>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="OCSPIdentifierType">
  <xsd:sequence>
    <xsd:element name="ResponderID" type="ResponderIDType"/>
    <xsd:element name="ProducedAt" type="xsd:dateTime"/>
  </xsd:sequence>
  <xsd:attribute name="URI" type="xsd:anyURI" use="optional"/>
</xsd:complexType>

<xsd:complexType name="OtherCertStatusRefsType">
  <xsd:sequence>
    <xsd:element name="OtherRef" type="AnyType"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
```

Empty CompleteRevocationRefs qualifying properties shall not be incorporated.

The CRLRefs element shall contain a sequence of references to CRLs.

Each CRLRef child of CRLRefs shall contain one reference to one CRL.

The DigestAlgAndValue child of CRLRef element shall contain one indication of a digest algorithm, and the base-64 encoding of the digest value of the DER-encoded referenced CRL.

The CRLIdentifier child needs not to be present if the referenced CRL can be inferred from other information.

The CRLIdentifier child of CRLRef element shall include the name issuer in its Issuer element.

The value of Issuer child of CRLIdentifier element shall fulfil the requirements specified in **4.5.4.1** of XMLDSIG for strings representing Distinguished Names.

The CRLIdentifier child of CRLRef element shall include the time when the CRL was issued in its IssueTime element.

The CRLIdentifier child of CRLRef element may include the number of the CRL in its Number element.

NOTE- The Number element is an optional hint helping to get the CRL whose digest matches the value present in the reference.

URI attribute of CRLIdentifier element shall indicate one place where the referenced CRL can be found.

NOTE - It is intended that this attribute be used as a hint, as implementations can have alternative ways for retrieving the referenced CRL if it is not found at the referenced place.

If one or more of the identified CRLs are a Delta CRL, this qualifying property shall include references to the set of CRLs required to provide complete revocation lists.

The OCSPRefs element shall contain a sequence of references to OCSP responses.

Each OcspRef child of OCSPRefs shall contain one reference to one OCSP response.

The OCSPIdentifier child of OCSPRef element shall include an identifier of the responder in its ResponderID child.

If the responder is identified by its name, then this name shall appear within the ByName child of ResponderID element.

The value of ByName element shall fulfil the requirements for strings representing Distinguished Names.

If the responder is identified by the digest of the server's public key computed as mandated in IETF RFC 6960 , then the base-64 encoding of the DER-encoded of byKey field specified in IETF RFC 6960 shall appear within the ByKey child of ResponderID element.

The OCSPIdentifier child of OCSPRef element shall include the generation time of the OCSP response in its ProducedAt child.

The value in ProducedAt child of OCSPIdentifier shall indicate the same time as the time indicated by the ProducedAt field of the referenced OCSP response.

URI attribute of OCSPIdentifier element indicates one place where the referenced OCSP response can be archived.

NOTE - It is intended that this attribute be used as a hint, as implementations can have alternative ways for retrieving the referenced OCSP response if it is not found at the referenced place.

The DigestAlgAndValue child of OCSPRef element shall contain one indication of a digest algorithm, and the base-64 encoding of the DER-encoded OCSPResponse field defined in IETF RFC 6960 .

The DigestAlgAndValue child element should be included within the OCSPRef element.

NOTE - The absence of the DigestAlgAndValue child of OCSPRef element makes OCSP responses substitutions attacks possible, if for instance OCSP responder keys are compromised. In this case, out-of-band mechanisms can be used to ensure that none of the OCSP responder keys have been compromised at the time of validation.

References to alternative forms of validation data may be included in this qualifying property making use of the OtherRefs element, a sequence whose items (OtherRef elements) may contain any kind of information. Their semantics and syntax are outside the scope of the standard.

If at least one of the following unsigned properties: RevocationValues, AttributeRevocationValues, AnyValidationData with a non empty RevocationValues child element, or the ArchiveTimeStamp defined in namespace whose URI is http://uri.etsi.org/01903/v1.4.1#, is incorporated into the

signature, all the revocation data referenced in CompleteRevocationRefs shall be present elsewhere in the signature.

NOTE - If XML electronic time-stamps based in XMLDSIG are standardized and spread, this type can also serve to contain references to the full set of CRL or OCSP responses that have been used to verify the certification chain for any TSUs providing such electronic time-stamps. In this case, an element of this type can be added as an unsigned qualifying property to the XML electronic time-stamp using the incorporation mechanisms defined in the standard.

B-1.3 The AttributeCertificateRefsV2 Qualifying Property

Semantics

The AttributeCertificateRefsV2 qualifying property shall be an unsigned qualifying property that qualifies the signature.

The AttributeCertificateRefsV2 qualifying property:

a) Shall contain, if they are not present within CompleteCertificateRefsV2 or SigningCertificateV2 qualifying properties, the references to the trust anchors if certificates exist for them, and the references to CA certificates within the path of the signing certificate(s) of the attribute certificate(s) and signed assertion(s) incorporated into the XAdES signature. References present within CompleteCertificateRefsV2 or SigningCertificateV2 qualifying properties should not be included.

b) Shall contain, if they are not present within CompleteCertificateRefsV2 or SigningCertificateV2 qualifying properties, the reference(s) to the signing certificate(s) of the attribute certificate(s) and signed assertion(s) incorporated into the XAdES signature. References present within CompleteCertificateRefsV2 or SigningCertificateV2 qualifying properties should not be included. And

c) May contain references to the certificates used to sign CRLs or OCSP responses and certificates within their respective certificate paths, which are used for validating the signing certificate(s) of the attribute certificate(s) and signed assertion(s) incorporated into the XAdES signature. References present within CompleteCertificateRefsV2 or SigningCertificateV2 qualifying properties should not be included.

Syntax

The AttributeCertificateRefsV2 element shall be defined as in XML Schema file "XAdES01903v141-202107.xsd", whose location is detailed in The file at https://uri.etsi.org/01903/v1.3.2/XAdES01903v132-201601.xsd (XAdES01903v132-201601.xsd) contains the definitions of qualifying properties defined within the namespace whose URI value is http://uri.etsi.org/01903/v1.3.2#.

D-2, and is copied below for information.

<!-- targetNamespace="http://uri.etsi.org/01903/v1.4.1#"

The preamble of the XML Schema file also includes the following namespace declaration:
xmlns:xades="http://uri.etsi.org/01903/v1.3.2#",
which assigns the prefix "xades" to the namespace whose URI is shown in the declaration.
-->

<xsd:element name="AttributeCertificateRefsV2" type="CompleteCertificateRefsTypeV2"/>

If at least one of the following unsigned properties: CertificateValues, AttrAuthoritiesCertValues, AnyValidationData with a non empty CertificateValues child element, or the ArchiveTimeStamp defined in namespace whose URI is http://uri.etsi.org/01903/v1.4.1#, is incorporated into the signature, all the certificates referenced in AttributeCertificateRefsV2 shall be present elsewhere in the signature.

B-1.4 The AttributeRevocationRefs Qualifying Property

Semantics

The `AttributeRevocationRefs` qualifying property shall be an unsigned qualifying property that qualifies the signature.

The `AttributeRevocationRefs` qualifying property:

a) Shall contain, if they are not present within the `CompleteRevocationRefs` qualifying property, the references to the revocation values corresponding to CA certificates within the path(s) of the signing certificate(s) of the attribute certificate(s) and signed assertion(s) incorporated into the XAdES signature. It shall not contain a revocation value for the trust anchors. References present within `CompleteRevocationRefs` qualifying property should not be included.

NOTE — A trust anchor is by definition trusted, thus no revocation information for the trust anchor is used during the validation.

b) Shall contain, if they are not present within the `CompleteRevocationRefs` qualifying property, the references to the revocation value(s) for the signing certificate(s) of the attribute certificate(s) and signed assertion(s) incorporated into the XAdES signature. References present within CompleteRevocationRefs property should not be included. And

c) May contain references to the revocation values on certificates used to sign CRLs or OCSP responses and certificates within their respective certificate paths, which are used for validating the signing certificate(s) of the attribute certificate(s) and signed assertion(s) incorporated into the XAdES signature. References present within CompleteRevocationRefs property should not be included.

Syntax

The `AttributeRevocationRefs` qualifying property shall be defined as in XML Schema file "**Error! Reference source not found.**", whose location is detailed in D-1, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.3.2#" -->

<xsd:element name="AttributeRevocationRefs"
type="CompleteRevocationRefsType"/>
```

If one or more of the identified CRLs are a Delta CRL, this property shall include references to the set of CRLs required to provide complete revocation lists.

If at least one of the following unsigned properties: RevocationValues, AttributeRevocationValues, AnyValidationData with a non empty RevocationValues child element, or the ArchiveTimeStamp defined in namespace whose URI is http://uri.etsi.org/01903/v1.4.1#, is incorporated into the

signature, all the revocation data referenced in AttributeRevocationRefs shall be present elsewhere in the signature.

B-1.5 Time-Stamps On References To Validation Data

B-1.5.1 *The SigAndRefsTimeStampV2 Qualifying Property*

B-1.5.1.1 *Semantics and syntax*

Semantics

The `SigAndRefsTimeStampV2` qualifying property shall be an unsigned qualifying property qualifying the signature.

The `SigAndRefsTimeStampV2` qualifying property shall encapsulate electronic time-stamps on the digital signature value, the signature time-stamp, if present, and the XAdES qualifying properties containing references to validation data.

Syntax

The `SigAndRefsTimeStampV2` qualifying property shall be defined as in XML Schema file "XAdES01903v141-202107.xsd", whose location is detailed in **The** file at https://uri.etsi.org/01903/v1.3.2/XAdES01903v132-201601.xsd (XAdES01903v132-201601.xsd) contains the definitions of qualifying properties defined within the namespace whose URI value is http://uri.etsi.org/01903/v1.3.2#.

D-2, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.4.1#"

The preamble of the XML Schema file also includes the following namespace
declaration:
  xmlns:xades="http://uri.etsi.org/01903/v1.3.2#",
which assigns the prefix "xades" to the namespace whose URI is shown in
the declaration.
-->

<xsd:element name="SigAndRefsTimeStampV2"
type="xades:XAdESTimeStampType"/>
```

This qualifying property shall contain an electronic time-stamp that time-stamps the following XAdES components:
`ds:SignatureValue` element, all present `SignatureTimeStamp` qualifying properties, `CompleteCertificateRefsV2`, `CompleteRevocationRefs`, and when present, `AttributeCertificateRefsV2` and `AttributeRevocationRefs`.

Depending whether all the aforementioned time-stamped unsigned qualifying properties and the SigAndRefsTimeStampV2 qualifying property itself have the same parent or not, its contents may be different. Details are given in clauses below.

B-1.5.1.2 *Not distributed case*

If `SigAndRefsTimeStampV2` and all the unsigned qualifying properties covered by its electronic time-stamp have the same parent, this qualifying property shall use the implicit mechanism.

The input to the electronic time-stamp's message imprint computation input shall be the result of taking in order each of the XAdES components listed below, canonicalizing each one as specified in **4.5,** and concatenating the resulting octet streams:

a) The `ds:SignatureValue` element.

b) Those among the following unsigned qualifying properties that appear before `SigAndRefsTimeStampV2`, in their order of appearance within the `UnsignedSignatureProperties` element:
   1) the `SignatureTimeStamp` qualifying properties;
   2) the `CompleteCertificateRefsV2` qualifying property;
   3) the `CompleteRevocationRefs` qualifying property;
   4) the `AttributeCertificateRefsV2` qualifying property if it is present; and
   5) `the AttributeRevocationRefs` qualifying property if it is present.

B-1.5.1.3 *Distributed case*

If `SigAndRefsTimeStampV2` and some of the unsigned qualifying properties covered by its electronic time-stamp do not have the same parent, this qualifying property shall be built as indicated below:

a) No Include element will be added for `ds:SignatureValue`. It shall implicitly be assumed its contribution to the digest input (see below in this clause); and

b) Generate one Include element for each unsigned qualifying property that shall be covered by the electronic time-stamp in the order they appear listed below:

   1) the `SignatureTimeStamp` qualifying properties;
   2) the `CompleteCertificateRefsV2` qualifying property;
   3) the `CompleteRevocationRefs` qualifying property;
   4) the `AttributeCertificateRefsV2` qualifying property if it is present; and
   5) the `AttributeRevocationRefs` qualifying property if it is present.

The URI attributes shall be built following the rules stated in **5.1.4.4.2.1**.

The electronic time-stamp's message imprint computation input shall be built as indicated below:

a) Initialize the final octet stream as an empty octet stream;

b) Take the `ds:SignatureValue` element and its content. Canonicalize it as specified in Error! Reference source not found., and put the result in the final octet stream; and

c) Take each unsigned qualifying property listed above in the order they have been listed above (this order shall be the same as the order the `Include` elements appear in the property). For each one extract comment nodes, canonicalize it as specified in **4.5**, and concatenate the resulting octet string to the final octet stream.

B-1.5.2 *The RefsOnlyTimeStampV2 Qualifying Property*

B-1.5.2.1 *Semantics and syntax*

Semantics

The `RefsOnlyTimeStampV2` qualifying property shall be an unsigned qualifying property qualifying the signature.

The `RefsOnlyTimeStampV2` qualifying property shall encapsulate electronic time-stamps on the XAdES qualifying properties containing references to validation data.

Syntax

The `RefsOnlyTimeStampV2` qualifying property shall be defined as in XML Schema file XAdES01903v141-202107.xsd**,** whose location is detailed in D-2 of Annex D, and is copied below for information.

```
<!-- targetNamespace="http://uri.etsi.org/01903/v1.4.1#"

The preamble of the XML Schema file also includes the following namespace
declaration:
  xmlns:xades="http://uri.etsi.org/01903/v1.3.2#",
which assigns the prefix "xades" to the namespace whose URI is shown in
the declaration.
-->
<xsd:element name="RefsOnlyTimeStampV2" type="xades:XAdESTimeStampType"/>
```

This qualifying property shall contain an electronic time-stamp that time-stamps the following XAdES qualifying properties `CompleteCertificateRefsV2`, `CompleteRevocationRefs`, and when present, `AttributeCertificateRefsV2` and `AttributeRevocationRefs`.

Depending whether all the aforementioned time-stamped unsigned qualifying properties and the `RefsOnlyTimeStampV2` qualifying property itself have the same parent or not, its contents may be different. Details are given in clauses below.

B-1.5.2.2 *Not distributed case*

If `RefsOnlyTimeStampV2` and all the unsigned qualifying properties covered by its electronic time-stamp have the same parent, this qualifying property shall use the implicit mechanism. The electronic time-stamp's message imprint computation input shall be the result of taking those of the qualifying unsigned properties listed below that appear before the `RefsOnlyTimeStampV2` in their order of appearance within the `UnsignedSignatureProperties` element, canonicalizing each one as specified in Error! Reference source not found., and concatenating the resulting octet streams:

a) The `CompleteCertificateRefsV2` qualifying property;
b) The `CompleteRevocationRefs` qualifying property;
c) The `AttributeCertificateRefsV2` qualifying property if it is present; and
d) The `AttributeRevocationRefs` qualifying property if it is present.

B-1.5.2.3 *Distributed case*

If `RefsOnlyTimeStampV2` and some of the unsigned qualifying properties covered by its electronic time-stamp do not have the same parent, one `Include` element shall be generated for each unsigned qualifying property that shall be time-stamped by the electronic time-stamp in the order they appear listed below:

a) The `CompleteCertificateRefsV2` qualifying property;
b) The `CompleteRevocationRef`s qualifying property;
c) The `AttributeCertificateRefsV2` qualifying property if it is present; and
d) The `AttributeRevocationRefs` qualifying property if it is present.

The URI attributes shall be built following the rules stated in **5.1.4.4.2.1**.

The electronic time-stamp's message imprint computation input shall be built as indicated below:

a) Initialize the final octet stream as an empty octet stream; and
b) Take each unsigned qualifying property listed above in the order they have been listed above (this order shall be the same as the order the `Include` elements appear in the qualifying property). For each one extract comment nodes, canonicalize it as specified in **4.5**, and concatenate the resulting octet stream to the final octet stream.

B-2 DEPRECATED QUALIFYING PROPERTIES

B-2.1 Usage of Deprecated Qualifying Properties

B.2 lists deprecated qualified properties. They are kept in the document to facilitate the handling of XAdES signatures that meet the requirements of SSD 10 (27046), but they shall not be added to any new XAdES signature.

B-2.2 The RenewedDigests Qualifying Property

The RenewedDigests qualifying property as defined in **5.5.3** of SSD 10 (27046) is deprecated. Instead, the `RenewedDigestsV2` qualifying property as defined in **5.5.3** of this standard, shall be used.

ANNEX C
(Normative)

ALTERNATIVE MECHANISMS FOR LONG TERM AVAILABILITY AND INTEGRITY OF VALIDATION DATA

There may be mechanisms to achieve long term availability and integrity of validation data different from the ones described in **5.5**Error! Reference source not found..

If such a mechanism is incorporated into the signature using an unsigned property, then for this mechanism shall be specified:

a) The clear specification of the semantics and syntax of the property including its name, and namespace.

b) The strategy of how this mechanism guarantees that all necessary parts of the signature are protected by this property.

c) The strategy of how to handle signatures containing properties defined in the present document. In particular, in case ArchiveTimeStamp unsigned properties defined in the namespace whose URI is http://uri.etsi.org/01903/v1.4.1# are already incorporated into the signature, it shall be ensured that the previous electronic time-stamps within these properties are not invalidated, and that all validation material needed to validate the signature before the incorporation of the new property is incorporated into the signature and protected by the new property.

d) The strategy of how to handle legacy XAdES signatures. In particular it shall be guaranteed that in case of previously incorporated properties for long term availability and integrity of validation data they are not invalidated.

NOTES

1 Such mechanisms, defined outside of the standard, can be used to provide long term availability and integrity of validation data. However, they do not represent XAdES-B-LTA level as defined in **6** or XAdES-E-A levels as defined in ETSI EN 319 132-2.

2 Such mechanisms might be included in future versions of the standard and assigned to a corresponding XAdES level.

ANNEX D
(Normative)

XML SCHEMA FILES

D-1 XML SCHEMA FILE LOCATION FOR NAMESPACE http://uri.etsi.org/01903/v1.3.2#

The file at https://uri.etsi.org/01903/v1.3.2/XAdES01903v132-201601.xsd (XAdES01903v132-201601.xsd) contains the definitions of qualifying properties defined within the namespace whose URI value is http://uri.etsi.org/01903/v1.3.2#.

D-2 XML SCHEMA FILE LOCATION FOR NAMESPACE http://uri.etsi.org/01903/v1.4.1#

The file at https://uri.etsi.org/01903/v1.4.1/XAdES01903v141-202107.xsd (XAdES01903v141-202107.xsd) contains the definitions of qualifying properties defined within the namespace whose URI value is http://uri.etsi.org/01903/v1.4.1#.

ANNEX E
(Normative)

DEPRECATED QUALIFYING PROPERTIES

The qualifying properties, specified in ETSI TS 101 903 (V1.4.2) and listed below, are deprecated. XAdES signatures shall not include any of these qualifying properties.

a) The SigningCertificate qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.3.2#, and specified in ETSI TS 101 903 (V1.4.2). Instead the SigningCertificateV2 qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.3.2#, specified in **5.2.2** of the standard, shall be used.

b) The SignatureProductionPlace qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.3.2#, and specified in ETSI TS 101 903 (V1.4.2). Instead the SignatureProductionPlaceV2 qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.3.2#, specified in **5.2.5** of the standard, shall be used.

c) The SignerRole qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.3.2#, and specified in ETSI TS 101 903 (V1.4.2). Instead the SignerRoleV2 qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.3.2#, specified in **5.2.6** of the standard, shall be used.

d) The CompleteCertificateRefs qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.3.2#, and specified in ETSI TS 101 903 (V1.4.2). Instead the CompleteCertificateRefsV2 qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.4.1#, specified in **B-1.1** of the standard, shall be used.

e) The AttributeCertificateRefs qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.3.2#, and specified in ETSI TS 101 903 (V1.4.2). Instead the AttributeCertificateRefsV2 qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.4.1#, specified in **B-1.3** of the standard, shall be used.

f) The SigAndRefsTimeStamp qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.3.2#, and specified in ETSI TS 101 903 (V1.4.2) . Instead the SigAndRefsTimeStampV2 qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.4.1#, specified in **B-1.3** of the standard, shall be used. And

g) The RefsOnlyTimeStamp qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.3.2#, and specified in ETSI TS 101 903 (V1.4.2) . Instead the RefsOnlyTimeStamp V2 qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.4.1#, specified in **B-1.3** of the standard, shall be used.

NOTE - The ArchiveTimeStamp qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.3.2# had already been deprecated in ETSI TS 101 903 (V1.4.2) by the ArchiveTimeStamp qualifying property defined in the namespace whose URI is http://uri.etsi.org/01903/v1.4.1#, which is the one defined and allowed by the standard.

ANNEX D
(Foreword)

BIBLIOGRAPHY

[1]	IS 19156 Electronic Signatures and Infrastructures (ESI) — Cryptographic Suites
[2]	SSD 10 (24323) Electronic Signatures and Infrastructures (ESI) — Procedures for Creation and Validation of AdES Digital Signatures Part 1: Creation and Validation
[3]	SSD 10 (27046) Electronic Signatures and Infrastructures (ESI) — XAdES digital signatures Part 1: Building blocks and XAdES baseline signatures
[4]	SSD 10 (27047) Electronic Signatures and Infrastructures (ESI) — The framework for standardization of signatures — Definitions and abbreviations
[5]	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
[6]	IETF RFC 6931 Additional XML Security Uniform Resource Identifiers (URIs)
[7]	OASIS Standard Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0
[8]	ETSI TR 119 000 Electronic Signatures and Infrastructures (ESI) — The framework for standardization of digital signatures and trust services — Overview
[9]	ETSI TR 119 100 Electronic Signatures and Infrastructures (ESI) — Guidance on the use of standards for signature creation and validation
[10]	ETSI TS 119 612 Electronic Signatures and Infrastructures (ESI) — Trusted Lists
[11]	ETSI TS 119 511 Electronic Signatures and Infrastructures (ESI) — Policy and security requirements for trust service providers providing long-term preservation of digital signatures or general data using digital signature techniques
[12]	ETSI TS 103 171 (V2.1.1) Electronic Signatures and Infrastructures (ESI) — XAdES Baseline Profile
[13]	IETF RFC 4998 Evidence Record Syntax (ERS)
[14]	IETF RFC 6283 Extensible Markup Language Evidence Record Syntax (XMLERS)
[15]	ETSI EN 319 132-2 Electronic Signatures and Infrastructures (ESI) — XAdES digital signatures Part 2: Extended XAdES signatures
[16]	ETSI TS 119 132-3 Electronic Signatures and Infrastructures (ESI) — XAdES digital signatures Part 3: Incorporation of Evidence Record Syntax (ERS) mechanisms in XAdES
[17]	ETSI TS 101 903 (V1.4.2) Electronic Signatures and Infrastructures (ESI) — XML Advanced Electronic Signatures (XAdES)
[20]	ETSI TS 101 903 (V1.4.1) XML Advanced Electronic Signatures (XAdES)