XAdES Long-Term Signature Format Profile
Version 1.0

March 2006

Next Generation Electronic Commerce Promotion Council of Japan (ECOM)

1. XML long-term signature formats

This document provides a profile of a long-term signature format used for XML signatures. The long-term signature format profile here implements ETSI[1] TS 101 903 V1.3.1 (2005-05), "XML Advanced Electronic Signatures (XAdES)". It applies essentially the same details contained in ETSI TS 101 703 V1.5.1 (2003-12) "Electronic Signature Formats", a CMS long-term signature format, to XML signatures. The CMS long-term signature format profile, which was developed at the same time as this profile, includes the required subset of information from ETSI TS 101 703 V1.5.1 (2003-12) "Electronic Signature Formats", and the content of XAdES is very similar to this document (ETSI TS 101 703 V1.5.1 (2003-12) "Electronic Signature Formats"). The signature format profile shown in this document is therefore also a XAdES subset, and follows the structure of the CMS long-term signature format profile in terms of content.

Three versions of XAdES exist: V1.1.1 (2002-02) given in a W3C Note[2], that described in the ETSI document, ETSI TS 101 903 V1.2.2 (2004-4), and the draft version of ETSI TS 101 903 V1.3.1 (2005-05) currently being formulated. Looking at domestic trends, implementation has still not progressed that far, however the latest version currently being formulated is to be detailed in a W3C Note, as V1.1.1 (2002-02) was, and so future implementation and diffusion of this technology is expected. This document therefore implements the latest draft, V1.3.1 (2005-05).

2. Signature format

Depending on the existence or not of a signature policy, two basic forms of electronic signatures can be used: "Basic Electronic Signatures" (XAdES-BES), and "Explicit Policy based Electronic Signatures" (XAdES-EPES). A signature policy specifies a set of rules relating to the generation and validation of signatures so that a signer and a validator can deem a digital signature to be valid. ETSI TR 102 038 V1.1.1 (2002-04), "XML format for signature policies" specifies an XML format that is computer processable, and the content, as with the CMS long-term signature format, is exactly the same as RFC 3125, "Electronic Signature Policies".

Electronic signature document forms are based on the W3C/IETF Recommendation "XML-Signature Syntax and Processing" (XMLDSIG) (February 2002).

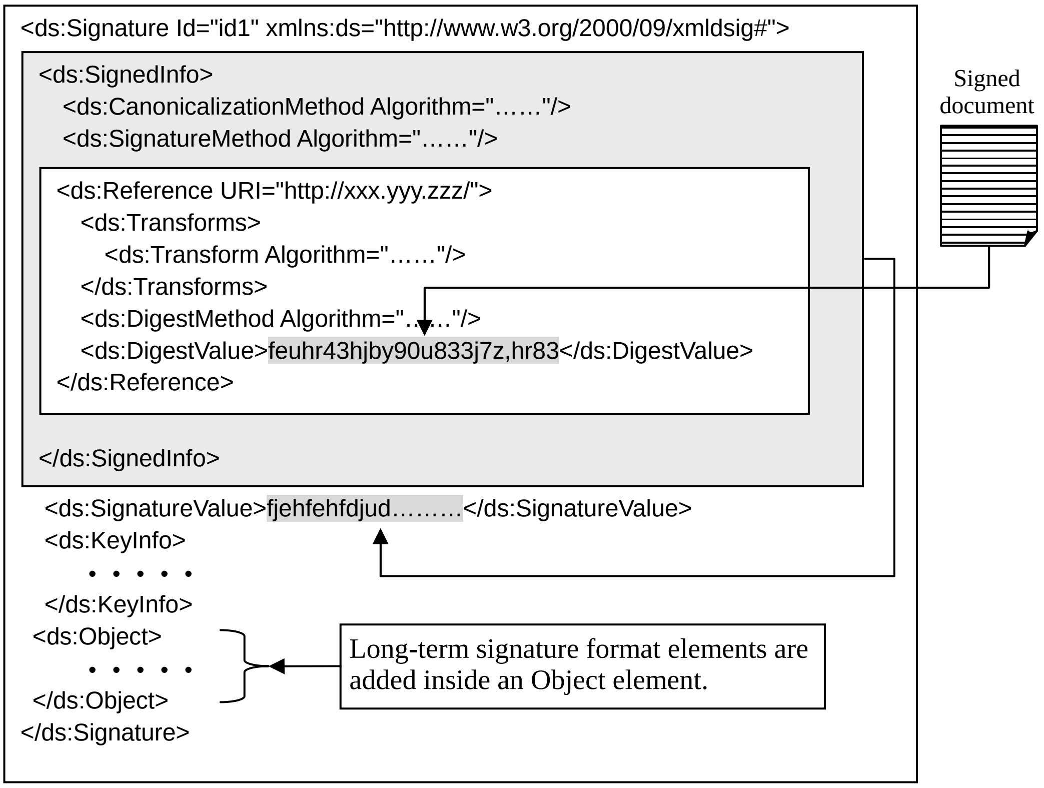
2.1. Basic format of XML signatures

The basic electronic format of XML signature documents is as defined in XMLDSIG. List 1 gives an example of a basic XML signature format.

[1] http://www.etsi.org/

[2] http://www.w3.org/TR/XAdES/

List 1: Example of a basic XML signature format



```
<ds:Signature Id="id1" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod Algorithm="……"/>
    <ds:SignatureMethod Algorithm="……"/>
    <ds:Reference URI="http://xxx.yyy.zzz/">
      <ds:Transforms>
        <ds:Transform Algorithm="……"/>
      </ds:Transforms>
      <ds:DigestMethod Algorithm="……"/>
      <ds:DigestValue>feuhr43hjby90u833j7z,hr83</ds:DigestValue>
    </ds:Reference>

  </ds:SignedInfo>
  <ds:SignatureValue>fjehfehfdjud………</ds:SignatureValue>
  <ds:KeyInfo>
    • • • • •
  </ds:KeyInfo>
 <ds:Object>
    • • • • •
 </ds:Object>
</ds:Signature>
```

Signed document
Long-term signature format elements are added inside an Object element.

In XML signatures, the signed object is specified by the URI attribute of the ds:Reference element within the XML signature document. Several signed objects can be specified, and the following may be specified: ancestor elements of the ds:Signature element within the same XML document (Enveloped); elements contained within the ds:Object element (Enveloping); elements that do not have a parent-child relationship with the ds:Signature element; and documents of arbitrary formats stored in separate files to the XML signature document (Detached). When XML signature documents and detached form XML signature documents that sign documents stored in separate files are stored for long periods, it is recommended that signed documents be stored separately.

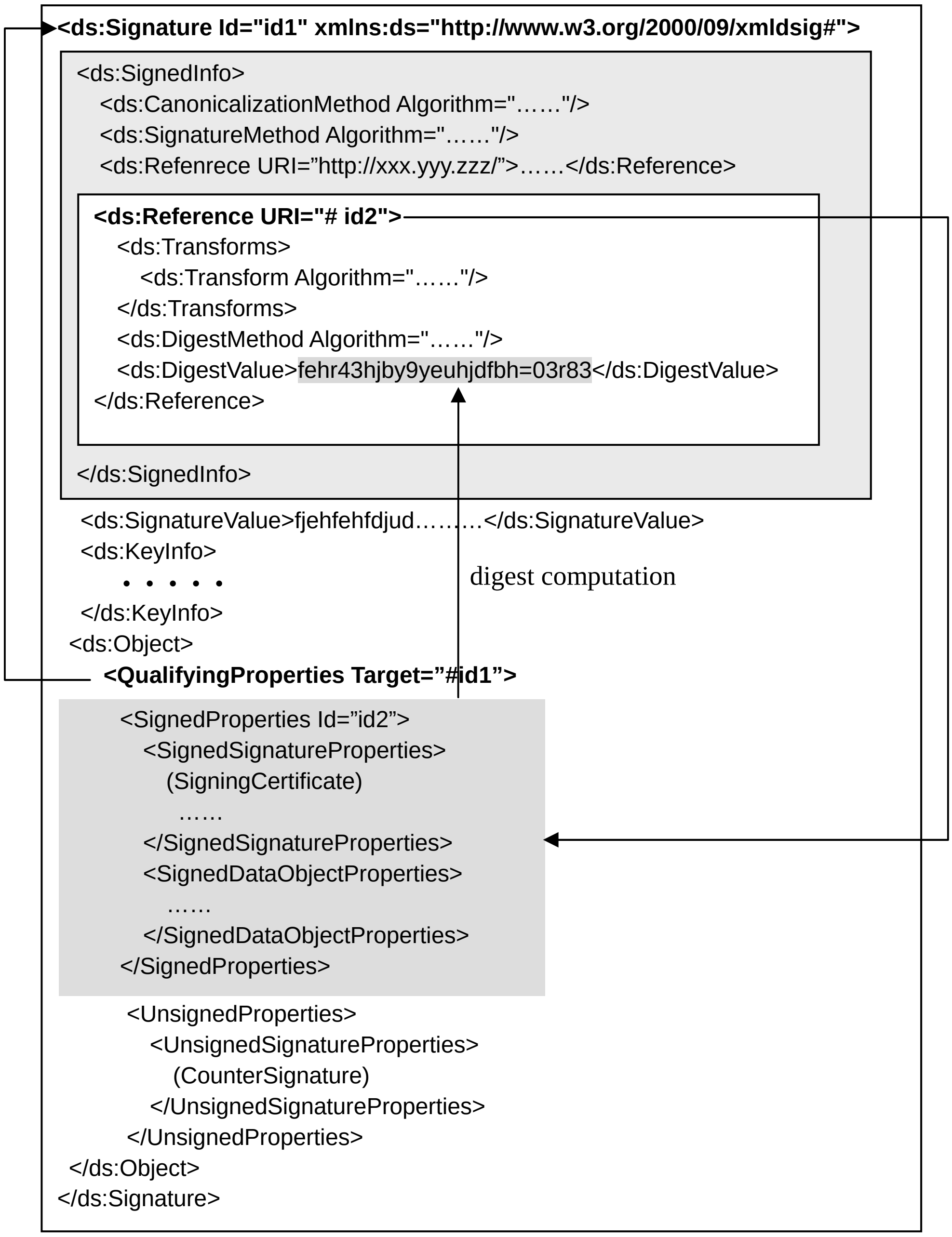
2.2. Long term signature formats

XAdES (V1.3.1) specifies four basic forms of XML advanced electronic signatures, namely the "Basic Electronic Signature" (XAdES-BES), the "Explicit Policy based Electronic Signature" (XAdES-EPES), the Electronic Signature with Time (XAdES-T), and the Electronic Signature with Complete Validation Data References (XAdES-C).

2.2.1. Basic format of electronic signature (XAdES)

A Basic Format of Electronic Signature (XAdES) builds on a XMLDSIG by incorporating required information into one ds:Object element. List 2 displays the basic format of electronic signature for XAdES structure, and the elements appearing in this list are explained.

List 2: The format of the structure of XAdES-BES



```
<ds:Signature Id="id1" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
 <ds:SignedInfo>
   <ds:CanonicalizationMethod Algorithm="……"/>
   <ds:SignatureMethod Algorithm="……"/>
   <ds:Refenrece URI=”http://xxx.yyy.zzz/”>……</ds:Reference>

  <ds:Reference URI="# id2">
    <ds:Transforms>
      <ds:Transform Algorithm="……"/>
    </ds:Transforms>
    <ds:DigestMethod Algorithm="……"/>
    <ds:DigestValue>fehr43hjby9yeuhjdfbh=03r83</ds:DigestValue>
  </ds:Reference>

 </ds:SignedInfo>
 <ds:SignatureValue>fjehfehfdjud………</ds:SignatureValue>
 <ds:KeyInfo>
     • • • • •
 </ds:KeyInfo>
 <ds:Object>
   <QualifyingProperties Target=”#id1”>
     <SignedProperties Id=”id2”>
       <SignedSignatureProperties>
         (SigningCertificate)
          ……
       </SignedSignatureProperties>
       <SignedDataObjectProperties>
         ……
       </SignedDataObjectProperties>
     </SignedProperties>
     <UnsignedProperties>
       <UnsignedSignatureProperties>
         (CounterSignature)
       </UnsignedSignatureProperties>
     </UnsignedProperties>
 </ds:Object>
</ds:Signature>
```

(1) The QualifyingProperties element

The QualifyingProperties element is contained in the ds:Object element and acts as a container element for all of the elements required for long-term storage. List 3 shows the XMLSchema definition for the QualifyingProperties element.

List 3: XMLSchema definition for the QualifyingProperties element

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

Elements within the QualifyingProperties element are split into SignedProperties elements that are cryptographically bound to (i.e. signed by) the XML signature, and UnsignedProperties elements that are not cryptographically bound to the XML signature. There MUST be at least one SignedProperties element. The mandatory Target attribute MUST refer to the Id attribute of the corresponding ds:Signature.

(2) The SignedProperties element

The SignedProperties element is referenced by a ds:Reference tag so as to be included in the XMLDSIG signature. The SignedProperties element MUST contain at least one SignedSignatureProperties element when signed, and may also contain a SignedDataObjectProperties element. List 4 shows the XMLSchema definition for the SignedProperties element.

List 4: XMLSchema definition for the SignedProperties element

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

Additionally, the value of the Type attribute of the ds:Reference element that references the SignedProperties element MUST be set to:

http://uri.etsi.org/01903/V1.3.1 - SignedProperties

(3) The UnsignedProperties element

The UnsignedProperties element contains a number of properties that are not signed. List 5 shows the XMLSchema definition for the UnsignedProperties element.

List 5: XMLSchema definition for the UnsignedProperties element

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

(4) The SignedSignatureProperties element

This element contains properties within its child elements that qualify the XML signature that has been specified with the Target attribute of the QualifyingProperties container element. The content of this element MAY be included in the signature computation so as to be covered by the XML signature. List 6 shows the XMLSchema definition for the SignedSignatureProperties element.

List 6: XMLSchema definition for the SignedSignatureProperties element

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

<xsd:complexType name="SignedSignaturePropertiesType">
  <xsd:sequence>
    <xsd:element name="SigningTime"
                    type="xsd:dateTime" minOccurs="0"/>
    <xsd:element name="SigningCertificate"
                    type="CertIDListType" minOccurs="0"/>
    <xsd:element name="SignaturePolicyIdentifer"
                    type="SignaturePolicyIdentifierType" minOccurs="0"/>
    <xsd:element name="SignatureProductionPlace"
                    type="SignatureProductionPlaceType"   minOccurs="0"/>
    <xsd:element name="SignerRole"
                    type="SignerRoleType" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

(5) The UnsignedSignatureProperties element

This element contains properties in its child elements that qualify the XML signature that has been specified with the Target attribute of the QualifyingProperties container element. The content of this element is not covered by the XML signature and may not be included in the signature computation. List 7 shows the XMLSchema definition for the UnsignedSignatureProperties element.

List 7: XMLSchema definition for the UnsignedSignatureProperties element

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

<xsd:complexType name="UnsignedSignaturePropertiesType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="CounterSignature"
                 type="CounterSignatureType" />
    <xsd:element name="SignatureTimeStamp"
                 type="XAdESTimeStampType/>
    <xsd:element name="CompleteCertificateRefs"
                 type="CompleteCertificateRefsType"/>
    <xsd:element name="CompleteRevocationRefs"
                 type="CompleteRevocationRefsType"/>
    <xsd:element name="AttributeCertificateRefs"
                 type="CompleteCertificateRefsType"/>
    <xsd:element name="AttributeRevocationRefs"
                 type="CompleteRevocationRefsType"/>
    <xsd:element name="SigAndRefsTimeStamp"
                 type="XAdESTimeStampType"/>
    <xsd:element name="RefsOnlyTimeStamp"
                 type="XAdESTimeStampType"/>
    <xsd:element name="CertificateValues"
                 type="CertificateValuesType"/>
    <xsd:element name="RevocationValues"
                 type="RevocationValuesType"/>
    <xsd:element name="ArchiveTimeStamp"
                 type="XAdESTimeStampType"/>
  </xsd:choice>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

This element may contain the elements required for long-term storage used in the various XAdES forms, and these are described below.

(6) The SignedDataObjectProperties element

This element contains properties that qualify some of the signed data objects, and is included in the signature value computation. List 8 shows the XMLSchema definition for the SignedDataObjectProperties element.

List 8: XMLSchema definition for the SignedDataObjectProperties element

```
<xsd:element name="SignedDataObjectProperties"
             type="SignedDataObjectPropertiesType"/>

<xsd:complexType name="SignedDataObjectPropertiesType">
  <xsd:sequence>
    <xsd:element name="DataObjectFormat"
                 type="DataObjectFormatType"
                 minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="CommitmentTypeIndication"
                 type="CommitmentTypeIndicationType"
                 minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="AllDataObjectsTimeStamp"
                 type="XAdESTimeStampType"
                 minOccurs="0" maxOccurs="unbounded"/>
    <xsd:element name="IndividualDataObjectsTimeStamp"
                 type="XAdESTimeStampType"
                 minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

(7) The UnsignedDataObjectProperties element

This element contains properties that qualify some of the signed data objects. This element is not included in the signature value computation. List 9 shows the XMLSchema definition for the UnsignedDataObjectProperties element. ETSI TS 101 703 V1.5.1 (2003-12) does not describe an UnsignedDataObjectProperties element, however, it is defined here for the sake of completeness and extensibility.

List 9: XMLSchema definition for the UnsignedDataObjectProperties element

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

<xsd:complexType name="UnsignedDataObjectPropertiesType">
  <xsd:sequence>
    <xsd:element name="UnsignedDataObjectProperty"
                 type="AnyType" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attribute name="Id" type="xsd:ID" use="optional"/>
</xsd:complexType>
```

2.2.2. Data type definitions

(1) The AnyType data type

The AnyType data type is used when you do not wish to restrict the content of elements. The content of elements of this data type may store arbitrary elements and text. Additionally, an element of this data type can bear an unrestricted number of arbitrary attributes. List 10 shows the XMLSchema definition for the AnyType data type.

List 10: XMLSchema definition for the AnyType data type

```
<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>
```

(2) The ObjectIdentifierType data type

The ObjectIdentifierType data type contains an Object IDentifier (OID), to identify a particular data object. List 11 shows the XMLSchema definition for the ObjectIdentifierType data type. It supports both the mechanism that is used to identify objects in ASN.1, an OID, and the mechanism that is usually used to identify XML resources, a URI.

- XML resources are identified by means of a Uniform Resource Identifier (URI) in the Identifier element. The optional Qualifier attribute does not appear.

- When an Object IDentifier (OID) is used to identify an object in ASN.1, it is either encoded as a Uniform Resource Name (URN) or as a Uniform Resource Identifier (URI). The Qualifier attribute is used to specify which encoding is used, and takes a value of either OIDAsURN or OIDAsURI.

The optional Description element contains an informal text describing the object identifier. The optional DocumentationReferences element consists of an arbitrary number of references pointing to further explanatory documentation of the object identifier.

List 11: XMLSchema definition for the ObjectIdentifierType data type

```
<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>
```

(3) The EncapsulatedPKIDataType data type

The EncapsulatedPKIDataType is used to incorporate ASN.1 encoded data into an XML structure. Examples of such data include X.509 certificates and revocation lists, attribute certificates and time-stamp tokens. The content of this data type is base64 encoded when incorporated into the XML structure. The Encoding attribute is a URI identifying the encoding used for the ASN.1 data. URIs that can be used are shown in Table 1. List 12 shows the XMLSchema definition for the EncapsulatedPKIDataType data type.

Table 1: URIs for the ASN.1 data encoding method

Encoding method	URI
DER	http://uri.etsi.org/01903#DER
BER	http://uri.etsi.org/01903#BER
CER	http://uri.etsi.org/01903#CER
PER	http://uri.etsi.org/01903#PER
XER	http://uri.etsi.org/01903#XER

List 12: XMLSchema definition for the EncapsulatedPKIDataType data type

```
<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>
```

(4) The XAdESTimeStampType data type

The XAdESTimeStampType, derived from the GenericTimeStampType data type, is used for containing time-stamps. Both the XAdESTimeStampType and the OtherTimeStampType data types are derived from the GenericTimeStampType data type, however, OtherTimeStampType is deprecated. List 13 shows the XMLSchema definition for the GenericTimeStampType data type. List 14 shows the XMLSchema definition for the XAdESTimeStampType data type derived from this.

List 13: XMLSchema definition for the GenericTimeStampType data type

```
<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" 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>
```

List 14: XMLSchema definition for the XAdESTimeStampType data type

```
<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>
```

XAdESTimeStampType data type elements contain time-stamp tokens, and two methods are specified for computing the digest value that is sent to the TSA:

- Implicit Mode
  When an Include element is not present in the elements of a XAdESTimeStampType, the XAdES elements included in the calculation of the digest value sent to the TSA are specified implicitly. Processing proceeds in the order shown below.

  For Signed Properties

  1. Retrieve each of the elements specified in the XAdESTimeStampType specification, and the content of signed elements.
  2. If a ds:Canonicalization element is present within the elements of the corresponding XAdESTimeStampType for the retrieved elements and signed contents, canonicalize them using the indicated algorithm. If a ds:Canonicalization element is not present, use the standard canonicalization method specified by XMLDSIG.
  3. Concatenate each of the processed pieces of data.

  For Unsigned Properties

  1. Retrieve all child elements of the UnsignedSignatureProperties appearing before properties that contain a time-stamp token.
  2. If a ds:Canonicalization element is present within the elements of the corresponding XAdESTimeStampType for the retrieved elements and

   signed contents, canonicalize them using the indicated algorithm. If a ds:Canonicalization element is not present, use the standard canonicalization method specified by XMLDSIG.

3. Concatenate each of the processed pieces of data.

In each case, the digest value is computed based on the concatenated data, and sent to the TSA. The following elements are computed in this mode:

- The SignatureTimeStamp element
- The RefsOnlyTimeStamp element
- The SigAndRefsTimeStamp element

- Include Mode
  If an Include element is present within a XAdESTimeStampType, the XAdES elements used in the calculation of the digest value sent to the TSA are specified explicitly. The Include element's URI attribute refers to an object to be used in the computation of the digest value sent to the TSA. When the ds:Reference element itself is referenced, the attribute referencedData MAY be present in a XAdESTimeStampType element. If present with value set to "true", the time-stamp is computed on the result of processing the corresponding ds:Reference element according to the XMLDSIG processing model. If the attribute referencedData is not present or is present with the value "false", then the time-stamp is computed on the ds:Reference element itself. Each Include element MUST be processed in order as detailed below.

  1. Retrieve the data object referenced in the URI attribute.
  2. If the retrieved data 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; if the value is "false" or, referencedData attribute is absent, retrieve the ds:Reference element itself as a computing object.
  3. If the resulting data is an XML node set, canonicalize it. If ds:Canonicalization is present, the algorithm indicated by this element is used. If not, the standard canonicalization method specified by XMLDSIG is used.
  4. Concatenate the results of processing to those resulting from previous processing of Include elements.

  The digest value is computed based on the concatenated data, and sent to the TSA. The following elements are computed in this mode:

  - The AllDataObjectsTimeStamp element
  - The IndividualDataObjectsTimeStamp element

The ArchiveTimeStamp element is used by both modes.

2.2.3. Basic electronic signature (XAdES-BES)

For XAdES-BES, one of the following is required:

- The SigningCertificate element contained in the SignedSignatureProperties element, and so subject to the signature value computation and included in the signature.

- ds:KeyInfo contained in the ds:Signature element, and so subject to the signature value computation and included in the signature.

Each of these is explained below.

(1) The SigningCertificate element

This element MUST contain the reference and the digest value of the signing certificate. It MAY contain digest values of other certificates and serial numbers that form a chain up to the point of trust. However, if specified in the signature policy, it MUST contain digest values of other certificates and issuer serial numbers that form a chain up to the point of trust. List 15 shows the XMLSchema definition for the SigningCertificate element. The reference to the signing certificate mentioned here means a ds:X509IssuerSerialType type IssuerSerial element that contains the DN of the Certifying Authority that issued the certificate, and the certificate serial number.

This element, along with other elements that contain SignedProperties, are subject to the signature value computation and so incorporated into the signature.

These elements are used to prevent simple substitution attacks.

List 15: XMLSchema definition for the SigningCertificate element

```
<xsd:element name="SigningCertificate" type="CertIDListType"/>

<xsd:complexType name="CertIDListType">
   <xsd:sequence>
      <xsd:element name="Cert" type="CertIDType" maxOccurs="unbounded"/>
   </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CertIDType">
   <xsd:sequence>
      <xsd:element name="CertDigest" type="DigestAlgAndValueType"/>
      <xsd:element name="IssuerSerial" type="ds:X509IssuerSerialType"/>
   </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>
```

(2) The ds:KeyInfo element

If a SigningCertificate element does not exist, or is not an object for the signature computation, then ds:KeyInfo element is mandate and the following restrictions apply.

- The ds:KeyInfo element MUST include a ds:X509Data containing the signing certificate;
- The ds:KeyInfo element also MAY contain other certificates forming a chain that MAY reach the point of trust;
- The ds:SignedInfo element MUST contain a ds:Reference element referencing ds:KeyInfo, so that the latter is included in the signature value computation.

2.2.4. Explicit Policy based Electronic Signatures (XAdES-EPES)

XAdES-EPES, one of the basic forms of XAdES, builds on XAdES-BES forms by incorporating the SignaturePolicyIdentifier element which concerns signature policy. The SignaturePolicyIdentifier element MUST be present in a XAdES-EPES, and is added to the SignedSignatureProperties element (List 6). These attributes are protected by the authors signature.

(1) The SignaturePolicyIdentifier element

List 16 shows the XMLSchema definition for the SignaturePolicyIdentifier element.

List 16: XMLSchema definition for the SignaturePolicyIdentifier element

```
<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>
```

2.2.5. Option XAdES-BES elements

A XAdES-BES signature MAY contain the following elements in SignedSignatureProperties.

- SigningTime
- SignatureProductionPlace
- SignerRole

The SignedDataObjectProperties element may contain the following elements.

- DataObjectFormat
- CommitmentTypeIndication
- AllDataObjectsTimeStamp
- IndividualDataObjectsTimeStamp

The UnsignedSignatureProperties element may contain the following elements.

- CounterSignature

Each of these is briefly explained below.

(1) The SigningTime element

The SigningTime property specifies the time at which the signer performed the signing process. "XMLSchema Part 2: Datatypes," a W3C Recommendation, defines an XML type xsd:dateTime that allows for the inclusion of the required information. This is the type selected for the SigningTime element. At most one SigningTime element MAY be present in the signature. List 17 shows the XMLSchema definition for the SigningTime element.

List 17: XMLSchema definition for the SigningTime element

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

(2) The SignatureProductionPlace element

This element specifies the location where the signature was created. List 18 shows the XMLSchema definition for the SignatureProductionPlace element.

List 18: XMLSchema definition for the SignatureProductionPlace element

```
<xsd:element name="SignatureProductionPlace"
                 type="SignatureProductionPlaceType"/>

<xsd:complexType name="SignatureProductionPlaceType">
   <xsd:sequence>
      <xsd:element name="City" 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>
```

(3) The SignerRole element

Some contracts may only be valid if signed by a user in a particular position within an organization. To cope with such cases, this element specifies the signer's role.

List 19 shows the XMLSchema definition for the SignerRole element. There are two different ways of specifying the signer's role:

- Using a claimed role name;
  (The role name is contained in the ClaimedRoles element)
- Using an attribute certificate containing a certified role.
  (The CertifiedRoles element contains an attribute certificate)

List 19: XMLSchema definition for the SignerRole element

```
<xsd:element name="SignerRole" type="SignerRoleType"/>
<xsd:complexType name="SignerRoleType">
  <xsd:sequence>
    <xsd:element name="ClaimedRoles"
                 type="ClaimedRolesListType" minOccurs="0"/>
    <xsd:element name="CertifiedRoles"
                 type="CertifiedRolesListType" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>

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

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

(4) The DataObjectFormat element

The DataObjectFormat element provides information that describes the format of the signed data object. This element SHOULD be present when the signed data is to be presented to human users on verification if the presentation format is not implicit within the data that has been signed. This element MAY be added for each signed data object. List 20 shows the XMLSchema definition for the DataObjectFormat element.

List 20: XMLSchema definition for the DataObjectFormat element

```
<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>
```

The mandatory ObjectReference attribute MUST reference the ds:Reference element of the ds:Signature corresponding with the data object qualified by this property. It may also convey the following information.

- Textual information related to the signed data object in a Description element;
- An identifier indicating the type of the signed data object in an ObjectIdentifier element;
- An indication of the MIME type of the signed data object in a MimeType element;
- An indication of the encoding format of the signed data object in an Encoding element.

At least one of the Description, ObjectIdentifier or MimeType elements MUST be present within the DataObjectFormat element.

(5) The CommitmentTypeIndication element

List 21 shows the XMLSchema definition for the CommitmentTypeIndication element. Table 2 shows the types of commitments already available and what they indicate.

List 21: XMLSchema definition for the CommitmentTypeIndication element

```
<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>
```

Table 2: Type of commitments already available and what they indicate

Commitment	Details
Proof of origin	Indicates that the signer recognizes to have created, approved and sent the signed data object.
Proof of receipt	Indicates that signer recognizes to have received the content of the signed data object.
Proof of delivery	Indicates that the TSP (Trusted Service Provider) providing that indication has delivered a signed data object in a local store accessible to the recipient of the signed data object.
Proof of sender	Indicates that the entity providing that indication has sent the signed data object (but not necessarily created it).
Proof of approval	Indicates that the signer has approved the content of the signed data object.
Proof of creation	Indicates that the signer has created the signed data object (but not necessarily approved, nor sent it).

(6) The AllDataObjectsTimeStamp element

The AllDataObjectsTimeStamp element contains the time-stamp computed before the signature production, over all of the signed data. List 22 shows the XMLSchema definition for the AllDataObjectsTimeStamp element.

List 22: XMLSchema definition for the AllDataObjectsTimeStamp element

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

The time-stamp is computed over the sequence formed by ALL the ds:Reference elements within the ds:SignedInfo referencing whatever the signer wants to sign except the SignedProperties element. Generating applications MUST compose the Include elements to refer to those ds:Reference elements, apart from those referenced by the SignedProperties element. Their corresponding referencedData attribute MUST be set to "true".

(7) The IndividualDataObjectsTimeStamp element

The IndividualDataObjectsTimeStamp element contains the time-stamp computed before the signature production, over individual data objects. List 23 shows the XMLSchema definition for the IndividualDataObjectsTimeStamp element.

List 23: XMLSchema definition for the IndividualDataObjectsTimeStamp element

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

The time-stamp is calculated over a sequence formed by SOME ds:Reference elements within the ds:SignedInfo. Note that this sequence cannot contain a ds:Reference references in the SignedProperties element. Generating applications MUST compose the Include elements to refer to those ds:Reference elements, apart from those referenced by the SignedProperties element. Their corresponding referencedData attribute MUST be present and set to "true". Several instances of this property can occur within the same XAdES.

(8) The CounterSignature element

The CounterSignature element is contained in an UnsignedSignatureProperties element. List 24 shows the XMLSchema definition for the CounterSignature element.

List 24: XMLSchema definition for the CounterSignature element

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

<xsd:complexType name="CounterSignatureType">
   <xsd:sequence>
      <xsd:element ref="ds:Signature"/>
   </xsd:sequence>
</xsd:complexType>
```

2.2.6. Electronic Signature with Time (XAdES-T)

XML Advanced Electronic Signature with Time (XAdES-T) involves the addition of a time-stamp token obtained from TSA, associated with the signature value in the electronic signature (the SignatureValue element within the ds:SignedInfo element contained within a ds:Signature element), for the purpose of providing a trusted time of existence for the digital signature. A time-stamp token generated from a signature value has the effect of providing a trusted time for the signature, as well as for the electronic data. XAdES-T forms may involve the addition of a SignatureTimeStamp element into the UnsignedSignatureProperties element of XAdES-BES or XAdES-EPS (refer to List 7 for the XMLSchema definition for the UnsignedSignatureProperties element).

(1) The SignatureTimeStamp element

The SignatureTimeStamp element encapsulates the time-stamp over the ds:SignatureValue element. A XAdES form MAY contain several SignatureTimeSamp elements, containing time-stamps obtained from different TSAs but corresponding to the one signature. List 25 shows the XMLSchema definition for the SignatureTimeStamp element.

List 25: XMLSchema definition for the SignatureTimeStamp element

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

The computation of the time-stamp token contained in this element is executed in Implicit Mode. Specifically, the ds:SignatureValue is used as input for the computation of the digest value sent to the TSA.

The validation data (certification path and revocation data) of the time-stamp token itself is contained within either of the following:

1) Included in the time-stamp

2) In the same place as the validation data for the signing certificate (CompleteCertificateRefs, CompleteRevocationRefs, CertificateValues, RevocationValues).

2.2.7. Electronic Signature with Complete validation data references (XAdES-C)

XML Advanced Electronic Signature with Complete validation data references (XAdES-C) adds to XAdES-T the CompleteCertificateRefs and CompleteRevocationRefs elements. The CompleteCertificateRefs element contains a sequence of references to the full set of CA certificates that have been used to validate the electronic signature up to (but not including) the signing certificate. CompleteRevocationRefs element contains a full set of references to the revocation data that have been used in the validation of the signer and CA certificates.

Note that CompleteCertificateRefs and CompleteRevocationRefs are optional elements when forming XAdES-A signatures as described below.

(1) The CompleteCertificateRefs element

A XAdES signature MAY contain at most one CompleteCertificateRefs element. List 26 shows the XMLSchema definition for the CompleteCertificateRefs element.

List 26: XMLSchema definition for the CompleteCertificateRefs element

```
<xsd:element name="CompleteCertificateRefs"
             type="CompleteCertificateRefsType"/>

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

(2) The CompleteRevocationRefs element

List 27 shows the XMLSchema definition for the CompleteRevocationRefs element.

List 27: XMLSchema definition for the CompleteRevocationRefs element

```
<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="OCSPIdentifierType">
  <xsd:sequence>
    <xsd:element name="ResponderID" type="xsd:string"/>
    <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>
```

ETSI TS 101 903 V1.3.1 (2005-05) indicates that if attribute certificates appear in the signature, the AttributeCertificateRefs and the AttributeRevocationRefs elements may also be added. However, in accordance with the policy of this document to present effective and efficient (bare required minimum) profiles, AttributeCertificate is not mentioned.

2.2.8. Extended signatures with time forms (XAdES-X)

XAdES-X builds on XAdES-C in order to protect against the later compromise of a CA, to ensure the completeness of validation data, and to protect against difficulties in obtaining validation data. 2 types of profiles for extending XAdES-C are defined.

Note that the SigAndRefsTimeStamp and RefsOnlyTimeStamp elements, included for XAdES-X formation, are optional when forming XAdES-A signatures as described below.

- XAdES-X type 1
  XAdES-X type 1 is built by obtaining and adding a time-stamp corresponding to a whole XAdES-C signature, and specifically it involves adding a SigAndRefsTimeStamp element to the UnsignedSignatureProperties element (refer to List 7 for the XMLSchema definition for the UnsignedSignatureProperties element). This profile is defined so that one or more SigAndRefsTimeStamp elements each containing one time-stamp obtained from different TSPs can be added. The hash value sent to the TSPs for obtaining the time-stamps are computed on the SignatureValue element, the SignatureTimeStamp element, the CompleteCertificateRefs element and the CompleteRevocationRefs element.

- XAdES-X type2
  XAdES-X type 2 is built by obtaining and adding a time-stamp corresponding only to references to validation data, and specifically it involves adding a RefsOnlyTimeStamp element to the UnsignedSignatureProperties element (refer to List 7 for the XMLSchema definition for the UnsignedSignatureProperties element). This profile is defined so that one or more RefsOnlyTimeStamp elements each containing one time-stamp obtained from different TSPs can be added. The hash

value sent to the TSPs for obtaining the time-stamps are computed on the CompleteCertificateRefs element and the CompleteRevocationRefs element.

(1) The SigAndRefsTimeStamp element

The computation of the time-stamp token contained in this element is executed in Implicit Mode. Specifically, the ds:SignatureValue element and all of the SignatureTimeStamp elements are canonicalized and concatenated. Next, the CompleteCertificateRefs and CompleteRevocationRefs elements are each canonicalized, and concatenated according to the order in which they appear in the XAdES document. The result of concatenation is used as input in the computation of the digest value sent to the TSA. List 28 shows the XMLSchema definition for the SigAndRefsTimeStamp element.

List 28: XMLSchema definition for the SigAndRefsTimeStamp element

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

(2) The RefsOnlyTimeStamp element

This element contains the time-stamp calculated over the concatenation of CompleteCertificateRefs and CompleteRevocationRefs elements. The computation of the time-stamp token contained in this element is executed in Implicit Mode. Specifically, each of the CompleteCertificateRefs and CompleteRevocationRefs elements are canonicalized, and concatenated according to their order of appearance in the XAdES document. The result of concatenation is used as input in the computation of the digest value sent to the TSA. List 29 shows the XMLSchema definition for the RefsOnlyTimeStamp element.

List 29: XMLSchema definition for the RefsOnlyTimeStamp element

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

2.2.9. Extended long electronic signatures with time (XAdES-X-L)

XAdES-X-L builds on XAdES-X types 1 or 2 by adding the CertificateValues and RevocationValues to the UnsignedSignatureProperties element (refer to List 7 for the XMLSchema definition for the UnsignedSignatureProperties element).

Note that when forming XAdES-A signatures as described below, it is not necessary to form a XAdES-X-L corresponding to XAdES-X type 1 or 2. It is sufficient to have the CertificateValues and RevocationValues elements.

(1) The CertificateValues element

The CertificateValues element must contain the signing certificate and the full chain of certificates referenced in CompleteCertificateRefs elements. However, certificates already present in a ds:KeyInfo element contained in a ds:Signature element do not need to be placed in a CertificateValues element. List 30 shows the XMLSchema definition for the CertificateValues element.

List 30: XMLSchema definition for the CertificateValues element

```
<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>
```

(2) The RevocationValues element

The RevocationValues element contains the certificate validation data required for signature validation. A XAdES signature MAY contain at most one RevocationValues element. List 31 shows the XMLSchema definition for the RevocationValues element.

List 31: XMLSchema definition for the RevocationValues element

```
<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>
```

2.2.10. Archival electronic signatures (XAdES-A)

For the long-term storage of electronic signatures, the signed content, the digital signature, the time-stamp and all of the validation data must be protected by a time-stamp, and by a tamper resistant framework. For long-term signature formats, this is achieved with archive time-stamps. In these cases, references to validation data, and time-stamps for the validation data references used for XAdES-C and XAdES-X are not required. It is sufficient if validation data itself is secured by CertificateValues and RevocationValues and an ArchiveTimeStamp is incorporated.

When attempting to extend the period of validity of an electronic signature to cover very long time frames, time-stamp signatures may be compromised, or TSA certificates may expire, and so several embedded time-stamp signatures may be required. The ArchiveTimeStamp element is used for this. The time-stamp process is iterated periodically.

(1) The ArchiveTimeStamp element

The ArchiveTimeStamp element contains archive time-stamps. The computation of the hash value included in the time-stamp contained in this element is performed as follows:

1. In the ArchiveTimeStamp element, generate an Include element for each ds:Reference in the ds:SignedInfo element. The URI attribute of the Include elements references each of the corresponding ds:Reference elements. The referencedData attribute must be set to "true".

2. Retrieve the following elements:

    - ds:SignedInfo
    - ds:SignatureValue
    - ds:KeyInfo

3. Retrieve the following elements in the order that they appear within the XAdES documents.

    - Any present SignatureTimeStamp element in the XAdES documents.
    - Any present CounterSignatureProperties element
    - Any present CompleteCertificateRefs element
    - Any present CompleteRevocationRefs element
    - The CertificateValues element. This element MUST be added if it is not already present.
    - The RevocationValues element. This element MUST be added if it is not already present.
    - The SigAndRefsTimeStamp element
    - The RefsOnlyTimeStamp element
    - Any previous ArchiveTimeStamp elements
    - Any ds:Object elements not containing QualifyingProperties, and not referenced by any ds:Reference

4. Canonicalize each retrieved element, concatenate the result and use this as input in the hash computation.

List 32 shows the XMLSchema definition for the ArchiveTimeStamp element. When several time-stamp requests are sent to different TSAs at the same time, the returned time-stamp tokens must all be contained in the same ArchiveTimeStamp element. Broadly, there are 2 possible ways of incorporating the validation data of the archive time-stamp itself (the certificate path from the time-stamp certificate all the way to its root CA, and the revocation data for each of the certificates).

[1] Incorporate the validation data into the time-stamp token

[2] Store the time-stamp token separately

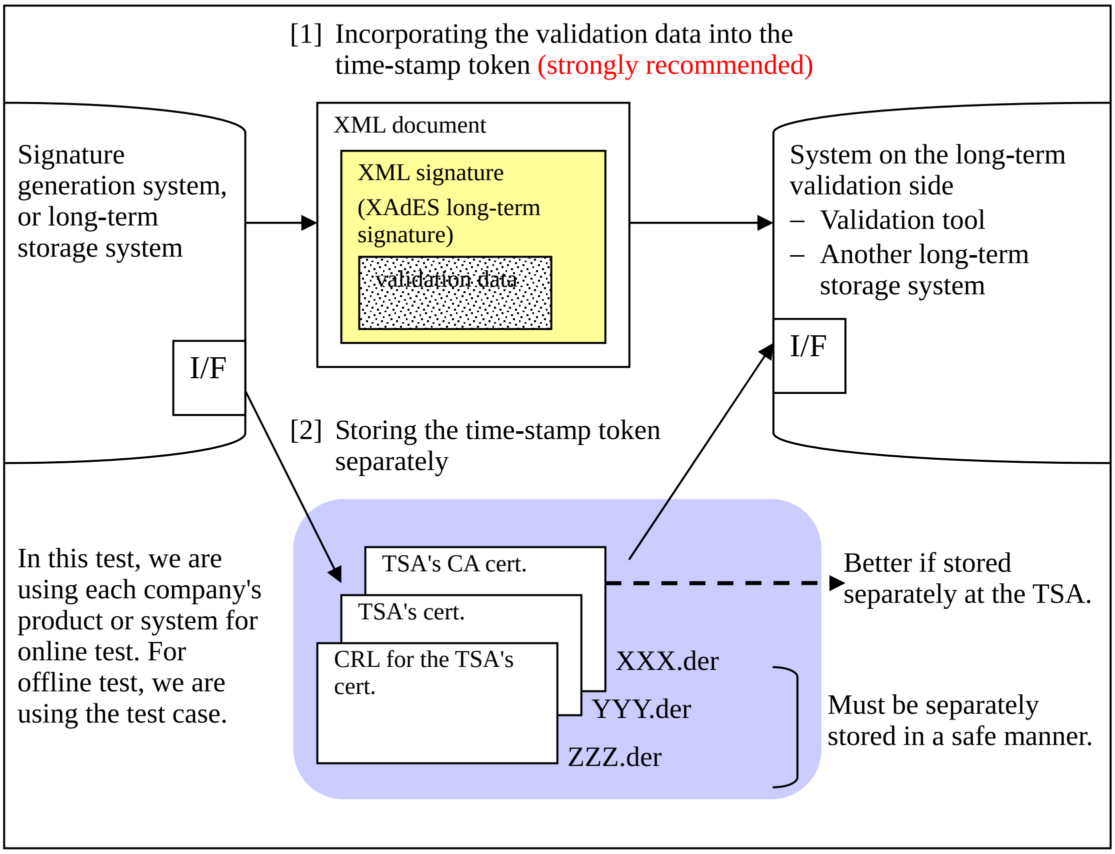

[1] Incorporating the validation data into the time-stamp token (strongly recommended)
Signature generation system, or long-term storage system
I/F
XML document
XML signature (XAdES long-term signature)
validation data
System on the long-term validation side
– Validation tool
– Another long-term storage system
I/F
[2] Storing the time-stamp token separately
In this test, we are using each company's product or system for online test. For offline test, we are using the test case.
TSA's CA cert.
TSA's cert.
CRL for the TSA's cert.
XXX.der
YYY.der
ZZZ.der
Better if stored separately at the TSA.
Must be separately stored in a safe manner.

Figure 1: Handling validation data for time-stamp tokens: there are two methods for dealing with time-stamp validation data: [1] incorporating the validation data into the time-stamp token itself, and [2] storing the time-stamp token separately. Method [1] is strongly recommended.

From an interoperability stand point, method [1] is strongly recommended since the validation method can be clearly defined. In addition, when method [1] is chosen, the following two possible methods (neither are mentioned in the standard specification) for incorporating the validation data into the long-term signature format. The first method is recommended at the time of formation, and it is recommended that both methods can be carried out for validation.

1) Certificate and CRLs within the time-stamp token
2) An unsigned attribute (extended validation data form) within the time-stamp token

On the other hand, consider method [2] from an interoperability stand point. The validator must be able to validate the time-stamp using the separately stored time-stamp validation data. Therefore, the validator's validating program must be able to read-in time-stamp validation data, and be able to perform validation of long-term signature formats using that data.

List 32: XMLSchema definition for the ArchiveTimeStamp element

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

2.3. Mandatory elements for the XAdES long-term signature format

This document has provided definitions of various forms of XAdES signatures. Table 3 shows which elements are mandatory, and which are optional for each XAdES form.

Table 3: XAdES forms and their elements

	XAdES-BES	XAdES-EPES	XAdES-T	XAdES-A	
QualifyingProperties	○	○	○	○	
SignedProperties	○	○	○	○	
SignedSignatureProperties	○	○	○	○	
SigningTime	△	△	△	△	Mandatory in XAdES (V1.1.1)
SigningCertificate	△1	△1	△1	△1	Mandatory in XAdES (V1.1.1)
SignaturePolicyIdentifier	×	○	△2	△2	Mandatory in XAdES (V1.1.1)
SignatureProductionPlace	△	△	△	△	
SignerRole	△	△	△	△	
SignedDataObjectProperties	△	△	△	△	
DataObjectFormat	△	△	△	△	
CommitmentTypeIndication	△	△	△	△	
AllDataObjectsTimeStamp	△	△	△	△	
IndividualDataObjectsTimeStamp	△	△	△	△	
UnsignedProperties	△	△	○	○	
UnsignedSignatureProperties	△	△	○	○	
CounterSignature	△	△	△	△*	
SignatureTimeStamp	×	×	○	○	
CompleteCertificateRefs	×	×	×	△*	
CompleteRevocationRefs	×	×	×	△*	
AttributeCertificateRefs	×	×	×	△3*	Not defined in XAdES (V1.1.1)
AttributeRevocationRefs	×	×	×	△3*	Not defined in XAdES (V1.1.1)
SigAndRefsTimeStamp	×	×	×	△3*	
RefsOnlyTimeStamp	×	×	×	△3*	
CertificateValues	×	×	×	○	
RevocationValues	×	×	×	○	
ArchiveTimeStamp	×	×	×	○	

○: Mandatory element

△: Optional element (the existence of these attributes at the time of formation or validation will not give rise to errors.)

△1: When the signing certificate is contained in ds:KeyInfo, and the following conditions are met, these elements are not required:

- The ds:KeyInfo element MUST include a ds:X509Data containing the signing certificate;
- The ds:KeyInfo element also MAY contain other certificates forming a chain that MAY reach the point of trust;
- The ds:SignedInfo element MUST contain a ds:Reference element referencing ds:KeyInfo, so that the latter is included in the signature value computation as a computation object for the signature.

△2: The SignaturePolicyIdentifier is a mandatory element when XAdES-T or XAdES-A signatures are formed based on XAdES-EPES.

△3: Optional but not recommended.

*: If present, must be included in the ArchiveTimeStamp computation.

×: Not required (elements that should not be present)

The ETSI 101 903 "XML Advanced Electronic Signatures (XAdES)" version to be implemented should be chosen from Table 4.

Table 4: Versions of ETSI 101 903 "XML Advanced Electronic Signatures (XAdES)" to be implemented

	ETSI TS 101 903 V1.1.1 (2002-02)	ETSI TS 101 903 V1.2.2 (2004-4)	ETSI TS 101 903 V1.3.1 (2005-05)
Implemented version	△	△	△