

Representing Digital Signatures Using the XML-DSIG (XML Signature) Format

Version 1.0

1 Annotation

This document defines the method of using the XML format to represent data elements that form a digital signature.

2 References

- RFC2560** Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C., X.509 Internet Public Key Infrastructure: Online Certificate Status Protocol - OCSP. June 1999.
- RFC3275** Eastlake 3rd D., Reagle J., Solo D., (Extensible Markup Language) XML-Signature Syntax and Processing. March 2002.
- ETSI TS 101 903** XML Advanced Electronic Signatures (XAdES).
- DAKÜP** Digitaalalkirja kontrolli üldpõhimõtted (“General Principles of Digital Signature Verification”).
- APA** Ajatempliteenuse protokollid ja andmevormingud (“The Protocols and Data Formats of the Timestamping Service”).

3 Terms and Definitions

4 Introduction

XML Advanced Electronic Signatures (XAdES) [ETSI TS 101 903] defines the format that structurally enables to represent signed data, signature, and other security attributes related to digital signature (for example, validity confirmations).

This document defines the methods of representing a digital signature according to requirements of the specification [DAKÜP], using the [XAdES] format.

Obligatory elements and attributes specified in [XAdES] have been acquired without changes. The document describes also the valuation of those elements and attributes.

The document defines additional elements that enable using timestamps to determine the creation time of a signature.

5 General Structure of XML Digital Signature

The following is the general structure of an XML digital signature (using the notation defined in [RFC3275] chapter 2):

```
<ds:Signature ID?>
  <ds:SignedInfo>
    <ds:CanonicalizationMethod/>
```

```

        <ds:SignatureMethod/>
        (<ds:Reference (URI=)? >
            (<ds:Transforms>)?
            <ds:DigestMethod>
            <ds:DigestValue>
        </ds:Reference>)+
    </ds:SignedInfo>
    <ds:SignatureValue>
    (<ds:KeyInfo>)?
<ds:Signature>
<ds:Object>
    <QualifyingProperties>
        <SignedProperties>
            <SignedSignatureProperties>
                (SigningTime)
                (SigningCertificate)
                (SignaturePolicyIdentifier)
            </SignedSignatureProperties>
        </SignedProperties>
        <UnsignedProperties>
            <UnsignedSignatureProperties>
                (CompleteRevocationRefs)
                (CompleteCertificateRefs)
                (RevocationValues)?
                (CertificateValues)?
            </UnsignedSignatureProperties>
        </UnsignedProperties>
    </QualifyingProperties>
</ds:Object>
<ds:Object>
    <AdditionalProperties>
        <SignedProperties>
            (FreshnessDigest)
        </SignedProperties>
        <UnsignedProperties>
            (FreshnessToken)
            (SignatureTimestamp)
            (TimestampInput)
        </UnsignedProperties>
    </AdditionalProperties>
</ds:Object>

```

This specification titles the sub-elements of the `SignedProperties` element as *signed* elements, and the sub-elements of the `UnsignedProperties` element as *unsigned* elements.

6 Obligatory Elements of the [XAdES] Format

6.1 SigningTime

There are no changes compared to requirements in [XAdES] section 7.2.1. The value of this element is the declared signing time of a document. If the signature contains additional, timestamping, elements then the declared signing time must be in the time range specified by the timestamp and freshness token.

6.2 SigningCertificate

Compared to [XAdES], the following limitations are set:

- The signer inserts exactly one field into this field
- To identify the certificate, the verifier uses only the `CertDigest` element and ignores the `IssuerSerial` element.

6.3 SignaturePolicyIdentifier

Compared to [XAdES], the following limitations are set:

- The signer always uses the `SignaturePolicyImplied` option.
- The result of the signature verification is not set if the signer had used the `SignaturePolicyId` option.

6.4 CompleteCertificateRefs

Compared to [XAdES], the following limitation is set:

- The verifier uses only the `CertDigest` element for certificate identification and ignores the `IssuerSerial` element.

6.5 CompleteRevocationRefs

Compared to [XAdES], the following limitations are set:

- The optional attribute `CRLIdentifier` in the `CRLRefType` structure is not used.
- Using the `DigestAlgAndValue` attribute in the `OCSPRefType` structure is obligatory. The attribute `OCSPIdentifier` is not used for signature verification.
- References of type `OtherRefs` are not used.

6.6 CertificateValues and RevocationValues

There are no changes compared to requirements in [XAdES] sections 7.6.1 and 7.6.2.

It is recommended to add to the signature the certificates referenced by the `SigningCertificate` and `CompleteRevocationRefs` elements, as well as revocation lists and OCSP confirmations, before dispatching a document.

However, both communicating sides can make an agreement to omit easily acquirable elements (for example, Certificate Revocation Lists and certificates) from the signature, to reduce the amount of exchanged data.

7 Additional Elements

If timestamps are used for determining the signing time, then additional elements (freshness token, timestamp) specified in [DAKÜP] must be added to the signature. For this, a new element `AdditionalProperties` has been defined. The XML signature element `Reference` must include the `SignedProperties` sub-element and not include the `UnsignedProperties` sub-element.

7.1 Freshness Token

A freshness token is added to the signature in the unsigned element `FreshnessToken`, which is defined as follows:

```
<FreshnessToken>qZk+NkcGgWq6PiVxeFDCbJzQ2J0=</FreshnessToken>
```

The value of the element `FreshnessToken` is the Base64 encoded DER encoding of the `LinkedTimestampToken` datastructure (specified in [APA]) that represents the freshness token.

Freshness token's message digest is added to the signature in the `FreshnessDigest` element, which is defined as follows:

```
<FreshnessDigest Algorithm="http://www.w3.org/2000/09/xmldsig#sha1">  
    qZk+NkcGgWq6PiVxeFDCbJzQ2J0=  
</FreshnessDigest>
```

The element's value is calculated by hashing the DER encoding of the freshness token's `TimeStampInfo` field, using the hash function defined with the `Algorithm` attribute.

Both elements are obligatory.

7.2 Signature's Timestamp

A signature's timestamp is added to the `TimestampInput` element, which is defined as follows:

```
<TimestampInput>  
    <SignatureDigest Algorithm="http://www.w3.org/2000/09/xmldsig#sha1">  
        qZk+NkcGgWq6PiVxeFDCbJzQ2J0=  
    </SignatureDigest>  
    <CertificateRefsDigest  
CanonicalizationMethod="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"  
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1">  
        qZk+NkcGgWq6PiVxeFDCbJzQ2J0=  
    </CertificateRefsDigest>  
    <RevocationRefsDigest  
CanonicalizationMethod="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"  
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1">  
        qZk+NkcGgWq6PiVxeFDCbJzQ2J0=  
    </RevocationRefsDigest>  
</TimestampInput>
```

The `TimestampInput` element has the following sub-elements:

SignatureDigest – A message digest of the byte stream in the **Signature** element's sub-element **SignatureValue**. The digest is calculated with the hash function indicated by the **Algorithm** attribute.

CertificateRefsDigest – A message digest calculated with the hash function indicated by the attribute **Algorithm** over the **CompleteCertificateRefs** element inside the **QualifyingProperties** element. The **CompleteCertificateRefs** element is processed with the canonization method indicated by the **CanonicalizationMethod** attribute.

RevocationRefsDigest – A message digest calculated with the hash function indicated by the attribute **Algorithm** over the **CompleteRevocationRefs** element inside the **QualifyingProperties** element. The **CompleteRevocationRefs** element is processed with the canonization method indicated by the **CanonicalizationMethod** attribute.

The timestamped **TimestampInput** element is added to the signature in an unsigned element.

The signature's timestamp is added to the signature in the unsigned element **SignatureTimestamp**, which is defined as follows:

```
<SignatureTimestamp>
  <CanonicalizationMethod
    Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
  <TimestampValue>qZk+NkcGgWq6PiVxeFDCbJzQ2J0=</TimestampValue>
</SignatureTimestamp>
```

The following are the meanings of the **SignatureTimestamp** element's sub-elements:

CanonicalizationMethod – A canonical transformation applied to the **TimestampInput** element. A timestamp is added to the resulting byte stream and saved to the **TimestampValue** element.

TimestampValue – The Base64-encoded DER encoding of the **LinkedTimestampToken** data structure (specified in [APA]) that represents the signature's time stamp.

Both elements are obligatory.

8 Formal Document Structures

Below are the formal definitions of new data structures as XML DTD and the corresponding Schema.

8.1 DTD

```
<!ELEMENT AdditionalProperties (SignedProperties, UnsignedProperties)>
```

```
<!ELEMENT SignedProperties (FreshnessDigest)>
<!ELEMENT UnsignedProperties (FreshnessToken, SignatureTimestamp,
                             TimestampInput)>

<!ELEMENT FreshnessDigest (#PCDATA)>
<!ATTLIST FreshnessDigest Algorithm CDATA #REQUIRED>

<!ELEMENT FreshnessToken (#PCDATA)>

<!ELEMENT SignatureTimestamp (CanonicalizationMethod, TimestampValue)>

<!ELEMENT CanonicalizationMethod EMPTY>
<!ATTLIST CanonicalizationMethod Algorithm CDATA #REQUIRED>
<!ELEMENT TimestampValue (#PCDATA)>

<!ELEMENT TimestampInput (SignatureDigest, CertificateRefsDigest?,
                          RevocationRefsDigest?)>
<!ELEMENT SignatureDigest (#PCDATA)>
<!ATTLIST SignatureDigest Algorithm CDATA #REQUIRED>
<!ELEMENT CertificateRefsDigest (#PCDATA)>
<!ATTLIST CertificateRefsDigest Algorithm CDATA #REQUIRED>
<!ATTLIST CertificateRefsDigest CanonicalizationMethod CDATA #REQUIRED>
<!ELEMENT RevocationRefsDigest (#PCDATA)>
<!ATTLIST RevocationRefsDigest Algorithm CDATA #REQUIRED>
<!ATTLIST RevocationRefsDigest CanonicalizationMethod CDATA #REQUIRED>
```

8.2 Schema

```
<?xml version="1.0"?>

<schema
  xmlns='http://www.w3.org/2000/10/XMLSchema'
  targetNamespace='http://www.w3.org/namespace/'
  xmlns:t='http://www.w3.org/namespace/'>

  <element name='AdditionalProperties'>
    <complexType>
      <sequence>
        <element ref='t:SignedProperties' />
        <element ref='t:UnsignedProperties' />
      </sequence>
    </complexType>
  </element>

  <element name='SignedProperties'>
    <complexType>
      <sequence>
        <element ref='t:FreshnessDigest' />
      </sequence>
    </complexType>
  </element>

  <element name='UnsignedProperties'>
```

```
<complexType>
  <sequence>
    <element ref='t:FreshnessToken' />
    <element ref='t:SignatureTimestamp' />
    <element ref='t:TimestampInput' />
  </sequence>
</complexType>
</element>

<element name='FreshnessDigest'>
  <complexType mixed='true'>
    <attribute name='Algorithm' type='string' use='required' />
  </complexType>
</element>

<element name='FreshnessToken'>
  <complexType mixed='true'>
  </complexType>
</element>

<element name='SignatureTimestamp'>
  <complexType>
    <sequence>
      <element ref='t:CanonicalizationMethod' />
      <element ref='t:TimestampValue' />
    </sequence>
  </complexType>
</element>

<element name='CanonicalizationMethod'>
  <complexType>
    <attribute name='Algorithm' type='string' use='required' />
  </complexType>
</element>

<element name='TimestampValue'>
  <complexType mixed='true'>
  </complexType>
</element>

<element name='TimestampInput'>
  <complexType>
    <sequence>
      <element ref='t:SignatureDigest' />
      <element ref='t:CertificateRefsDigest' minOccurs='0' maxOccurs='1' />
      <element ref='t:RevocationRefsDigest' minOccurs='0' maxOccurs='1' />
    </sequence>
  </complexType>
</element>

<element name='SignatureDigest'>
  <complexType mixed='true'>
    <attribute name='Algorithm' type='string' use='required' />
  </complexType>
</element>
```

```
<element name='CertificateRefsDigest'>
  <complexType mixed='true'>
    <attribute name='Algorithm' type='string' use='required' />
    <attribute name='CanonicalizationMethod' type='string' use='required' />
  </complexType>
</element>

<element name='RevocationRefsDigest'>
  <complexType mixed='true'>
    <attribute name='Algorithm' type='string' use='required' />
    <attribute name='CanonicalizationMethod' type='string' use='required' />
  </complexType>
</element>
</schema>
```