XAdES Profiles of the OASIS Digital Signature Service

Committee Draft, 24 December 2004 (Working Draft 06)

Document identifier:

oasis-dss-1.0-profiles-XAdES-spec-cd-01

Location:

http://docs.oasis-open.org/dss/

Editor:

Juan Carlos Cruellas, *individual* <cruellas@ac.upc.es>

Contributors:

Nick Pope, individual <pope@secstan.com>
Ed Shallow, Universal Post Union <ed.shallow@rogers.com>
Trevor Perrin, individual

Abstract:

This draft defines one abstract profile of the OASIS DSS protocols for the purpose of creating and verifying XML or CMS based Advanced Electronic Signatures. It also defines two concrete sub-profiles: one for creating and verifying XML Advanced Electronic Signatures and the other for creating and verifying CMS based Advanced Electronic Signatures.

Status:

This is a **Committee Draft** produced by the OASIS Digital Signature Service Technical Committee. Committee members should send comments on this draft to dss@lists.oasis-open.org.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Digital Signature Service TC web page at http://www.oasis-open.org/committees/dss/ipr.php.

Table of Contents

1 Introduction

The DSS signing and verifying protocols are defined in **[DSSCore]**. As defined in that document, the DSS protocols have a fair degree of flexibility and extensibility. This document defines an abstract profile for the use of the DSS protocols for creating and verifying XML and binary Advanced Electronic Signatures as defined in [COMMENT: Bold these and other references:] **[XAdES]** and **[TS 101 733]**. This document also defines two concrete profiles derived from the abstract one: one for creating and verifying XAdES signatures and the other for creating and verifying signatures as defined in TS 101733.

1.1 Notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in IETF RFC 2119 **[RFC 2119]**. These keywords are capitalized when used to unambiguously specify requirements over protocol features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

This specification uses the following typographical conventions in text: `<ns:Element>`, `Attribute`, **Datatype**, `OtherCode`.

1.2 Namespaces

The structures described in this specification are contained in the schema file **[XAdES-ABS-XSD]**. All schema listings in the current document are excerpts from the schema file. In the case of a disagreement between the schema file and this document, the schema file takes precedence.

This schema is associated with the following XML namespace:

```
http://www.docs.oasis-open.org/dss/oasis-dss-1.0-profiles-XAdES-cd-01#
```

If a future version of this specification is needed, it will use a different namespace.

Conventional XML namespace prefixes are used in this document:

- The prefix `dss:` (or no prefix) stands for the DSS core namespace **[Core-XSD]**.
- The prefix `ds:` stands for the W3C XML Signature namespace **[XMLSig]**.
- The prefix xades: stands for ETSI XML Advanced Electronic Signatures (XAdES) document **[XAdES]**.

Applications MAY use different namespace prefixes, and MAY use whatever namespace defaulting/scoping conventions they desire, as long as they are compliant with the Namespaces in XML specification **[XML-ns]**.

2 Overview

This document defines three profiles of the protocols specified in: “Digital Signature Services Core Protocol and Elements” **[DSSCore]**.

The first one is an abstract profile defining messages for supporting the lifecycle of advanced electronic signatures. Both, XML and binary advanced electronic signatures are supported by this profile.

One concrete profile, derived from the aforementioned abstract profile, gives support to the lifecycle of XML advanced electronic signatures as specified in **[XAdES]**.

A second concrete profile, also derived from the abstract one, gives support to the lifecycle of binary advanced electronic signatures as specified in **[TS 101733].**

Implementations should implement one of the concrete profiles (or both) in order to request generation or validation of advanced electronic signatures in one of the two formats (or both).

3 Advanced Electronic Signature abstract profile

3.1 Overview

This abstract profile supports operations within each phase of the lifecycle of two types of advanced electronic signature:

- XML encoded signatures based on **[XMLSig]** such as specified in **[XAdES].**
- Binary encoded signatures based on **[RFC 3161]** such as specified in **[TS 101733].**

Henceforward, the document will use the term **advanced signature** when dealing with issues that affect to both types of signatures. The document will use XAdES or TS 101733 signatures when dealing with issues that affect one or the other but not both of them.

For the generation of advanced signatures, the following operations apply:

- SignRequest. This operation supports requests for:
  - Generating predefined advanced signature forms as defined in **[XAdES]** and **[TS 101733]**.
  - Generating XML signatures incorporating specific signed/unsigned properties whose combination does not fit any predefined XAdES signature form. In such cases, the form MUST have been defined in a proprietary specification and MUST be identified by one URI.
  - Generating CMS signatures incorporating specific signed/unsigned attributes whose combination does not fit any predefined **[TS 101733]** signature forms. In such cases, the form MUST have been defined in a proprietary specification and MUST be identified by one URI.
- SignResponse. This operation supports delivery of:
  - Predefined advanced signature forms as defined in **[XAdES]** and **[TS 101733].**
  - XML signatures with specific properties whose combination does not fit any predefined XAdES signature form. In such cases, the form MUST have been defined in some other specification and MUST be identified by one URI.
  - CMS signatures incorporating specific signed attributes whose combination does not fit any predefined **[TS 101733]** signature form. In such cases, the form MUST have been defined in some other specification and MUST be identified by one URI.

For advanced signature verification (and updating) the following operations apply:

- VerifyRequest. This operation supports requests for:
  - Verifying a predefined advanced signature form.
  - Verifying XML signatures incorporating specific properties whose combination does not fit any predefined XAdES signature form.
  - Verifying any of the signatures mentioned above PLUS updating them by addition of additional properties (time-stamps, validation data, etc) leading to a predefined XAdES form.
  - Verifying CMS signatures incorporating specific attributes whose combination does not fit any predefined **[TS 101733]** signature form.

    - Verifying any of the signatures mentioned above PLUS updating them by addition of additional attributes (time-stamps, validation data, etc) leading to a predefined **[TS 101733]** form.
    - Verifying a long-term advanced signature in a certain point of time.
- VerifyResponse. This operation supports delivery of:
    - Advanced signature verification result of signatures mentioned above.
    - Advanced signature verification result PLUS the updated signatures as requested.
    - Updated signatures as requested.

The material for each operation will clearly indicate the lifecycle phase it pertains to.

3.2 Profile Features

3.2.1 Scope

This document profiles the DSS signing and verifying protocols defined in **[DSSCore]**.

3.2.2 Relationship To Other Profiles

The profile in this document is based on the **[DSSCore]**. The profile in this document is not directly implementable, and may be further profiled.

3.2.3 Signature Object

This profile supports the creation and verification of advanced signatures as defined in **[XAdES]** and **[TS 101733]**.

This profile also supports update of advanced signatures by addition of unsigned properties (time-stamps and different types of validation data), as specified in **[XAdES]** and **[TS 101733]**.

3.3 Profile of Signing Protocol

The present profile allows requesting:

- Predefined forms of advanced electronic signatures as defined in **[XAdES]** and [TS **101733]**.
- Other forms of signatures based in **[XMLSig]** or **[RFC 3369]** defined in other specifications,

In both cases, the specific requested form will be identified by an URI.

According to this profile, the following predefined advanced signature forms defined in **[XAdES]** and **[TS 101733]** MAY be requested (those forms whose name begin by XAdES- are forms names for XAdES signatures; the other ones denote forms for TS 101733 signatures):

- BES and XAdES-BES. In this form, the signing certificate is secured by the signature itself.
- EPES and XAdES-EPES. This form incorporates an explicit identifier of the signature policy that will govern the signature generation and verification.
- ES-T and XAdES-T. This form incorporates a trusted time, by means of a time-stamp token or a time-mark.
- ES-C and XAdES-C.
- ES-X and XAdES-X.

- ES-X-L and XAdES-X-L.
- ES-A and XAdES-A.

In addition, the present profile provides means for requesting incorporation in any of the aforementioned forms any of the following properties: SigningTime, CommitmentTypeIndication, SignatureProductionPlace, SignerRole, IndividualDataObjectTimeStamp, AllDataObjectTimeStamp and DataObjectFormat.

Other electronic signature forms based in **[XMLSig]** or **[RFC 3369]**, defined elsewhere, MAY also be requested using the mechanisms defined in this profile.

3.3.1 Element <SignRequest>

3.3.1.1 Element <OptionalInputs>

None of the optional inputs specified in the **[DSS Core]** are precluded in this abstract profile. It only constrains some of them and specifies additional optional inputs.

3.3.1.1.1 Element <SignatureType>

This element is OPTIONAL. If present, `<SignatureType>` SHALL be either:

`urn:ietf:rfc:3275`

for requesting XML Signatures, or

`urn:-ietf:rfc:3369`

for requesting CMS Signatures, as defined in 7.1 of **[DSS Core]**.

If not present the signature type SHALL be implied by the selected `<SignaturePolicy>` or the signature policy applied by the server.

3.3.1.1.2 Element <SignatureForm>

The form of signature required MAY be indicated using the following optional input

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

If not present the signature form SHALL be implied by the selected `<SignaturePolicy>` or the signature policy applied by the server.

Section 8.1 of this abstract profile defines a set of URIs identifying the predefined advanced electronic signature forms specified in **[TS101733]** and **[XAdES]**.

Should other standard or proprietary specification define new signature forms and their corresponding URIs, concrete sub-profiles of this abstract profile could be defined for giving support to their verification and update.

Should a form identified by an URI, admit different properties combinations, the server will produce a specific combination depending on its policy or configuration settings.

3.3.1.1.3 Optional inputs < ClaimedIdentity> / <KeySelector>

As forms defined in **[XAdES]** and **[TS 101733]** require that the signing certificate is protected by the signature, the server MUST gain access to that certificate.

`<dss:ClaimedIdentity>` or `<dss:KeySelector>` optional inputs MAY be present. If they are not present, the server may use means not specified in this profile to identify the signer’s key and gain access to its certificate.

3.3.1.1.4 Element <AddTimeStamp>

This element MAY be used by the client to request the inclusion in the advanced signature of a time-stamp on all the data that are to be signed.

This profile defines the following value for its Type attribute.

urn:oasis:names:tc:dss:1.0:profiles:XAdES:AllDataObjectTimeStamp

Note: IndividualDataObjectsTimeStamp is requested using <SignedProperty> element as defined in section 3.3.1.1.5.5.

3.3.1.1.5 Element <SignedProperties>

The requester MAY request to the server the addition of optional signed properties using the <dss:SignedProperties> element's <dss:Property> child profiled as indicated in clauses below.

Signed properties that MAY be requested are: SigningTime, CommitmentTypeIndication, SignerRole, SignatureProductionPlace, DataObjectFormat, and IndividualDataObjectsTimeStamp.

3.3.1.1.5.1 Requesting SigningTime

Value for <Identifier> element:

urn:oasis:names:tc:dss:1.0:profiles:XAdES:SigningTime

No content is required for Value element, since it will be generated by the server.

3.3.1.1.5.2 Requesting CommitmentTypeIndication

Value for <Identifier> element:

urn:oasis:names:tc:dss:1.0:profiles:XAdES:CommitmentTypeIndication

When the value of the commitment is fixed by the requester, this property will have a value that the server will incorporate to the advanced signature. In such cases the <Value> element MUST have the following content:

```
<xs:element name="Commitment">
    <xs:complexType>
        <xs:choice>
            <xs:element ref="xades:CommitmentTypeIndication"/>
            <xs:element name="BinaryValue" type="xs:base64Binary"/>
        </xs:choice>
    </xs:complexType>
</xs:element>
```

Element <xades:CommitmentTypeIndication> will be present when requesting a XML signature.

Element <BinaryValue> will be present when requesting a ASN.1 signature. Its contents MUST be the base64 encoding of **CommitmentTypeIndication** ASN.1 type defined in **[TS 101733]**, DER-encoded.

3.3.1.1.5.3 Requesting SignatureProductionPlace

Value for <Identifier> element:

urn:oasis:names:tc:dss:1.0:profiles:XAdES:SignatureProductionPlace

No content is required for <Value> element, as it will be generated by the server.

3.3.1.1.5.4 Requesting SignerRole

Value for <Identifier> element:

urn:oasis:names:tc:dss:1.0:profiles:XAdES:SignerRole

When the value of the role is fixed by the requester, this property will have a value that the server will incorporate to the advanced signature. In such cases the <Value> element MUST have the following content:

```
<xs:element name="SignerRole">
    <xs:complexType>
        <xs:choice>
            <xs:element ref="xades:SignerRole"/>
            <xs:element name="BinaryValue" type="xs:base64Binary"/>
        </xs:choice>
    </xs:complexType>
</xs:element>
```

Element <xades:SignerRole> will be present when requesting a XML signature.

Element <BinaryValue> will be present when requesting a ASN.1 signature. Its contents MUST be the base64 encoding of **SignerAttribute** ASN.1 type defined in **[TS 101733]**, DER-encoded.

3.3.1.1.5.5 Requesting <xades:IndividualDataObjectTimeStamp>

This property is only incorporated in XAdES signatures, not in TS101733 signatures, because an XML signature is able to sign several documents.

Value for <Identifier> element:

urn:oasis:names:tc:dss:1.0:profiles:XAdES:IndividualDataObjectTimeStamp

In this case, the content of <Value> element will be the element <DocsToBeTimeStamped>, defined as shown below.

```
<xs:element name="DocsToBeTimeStamped" type="DocReferencesType"/>

<xs:complexType name="DocReferencesType">
   <xs:sequence>
      <xs:element name="DocReference" maxOccurs="unbounded"
         type="DocReferenceType"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="DocReferenceType">
   <xs:attribute name="WhichDocument" type="xs:IDREF"
      use="required"/>
   <xs:attribute name="RefId" type="xs:string" use="optional"/>
</xs:complexType>
```

WhichDocument attribute contains the reference to the document whose time-stamp is requested (see attribute ID in [CoreDSS] section 2.4.1).

[XAdES] mandates that <ds:Reference> elements corresponding to signed documents that have been individually time-stamped before being signed, must include an Id attribute. **[XAdES]** also mandates <xades:IndividualDataObjectsTimeStamp> element to use this Id attribute to indicate what signed documents have actually been time-stamped before signing. See **[XAdES]** <xades:TimeStampType> and <xades:IndividualDataObjectsTimeStamp> definitions for more details.

The client MAY request a value for the <ds:Reference> element’s Id attribute using the RefId optional attribute if a <dss:SignedReference> forcing a value for such an attribute is not present in the request. If the request does not specify a value for this attribute, then the server will automatically generate it.

3.3.1.1.5.6 Requesting data objects format

Both **[XAdES]** and **[TS101733]** specify signed properties containing information on the format of the signed documents.

Value for `Identifier` element:

`urn:oasis:names:tc:dss:1.0:profiles:XAdES:DataObjectFormat`

When the value of the data object formats are fixed by the requester, this property will have a value that the server will incorporate to the advanced signature. The content of `<Value>` element will be the element `<DocsFormat>`, defined as shown below.

```

<xs:element name="DocsFormat" type="DocsFormatType"/>

<xs:complexType name="DocsFormatType">
    <xs:sequence>
        <xs:choice>
            <xs:element name="DocFormat" type="DocFormatType"
maxOccurs="unbounded"/>
            <xs:element name="BinaryValue" type="xs:base64Binary"/>
        </xs:choice>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="DocFormatType">
    <xs:complexContent>
        <xs:extension base="DocReferenceType">
            <xs:sequence>
                <xs:element ref="xades:DataObjectFormat"/>
            </xs: sequence >
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```

Element `<DocFormat>` will be present when requesting a XML signature.

Element `<BinaryValue>` will be present when requesting a ASN.1 signature. Its contents MUST be the base64 encoding of **ContentHints** ASN.1 type defined in [RFC 2634] DER-encoded.

3.3.2 Element <SignResponse>

This clause profiles the `dss:SignResponse` element for the requests profiled in clause 3.3.1

3.3.2.1 Element <Result>

This profile does not apply any restriction to the `dss:Result` element.

3.3.2.2 Element <SignatureObject>

The following restrictions apply to the contents of `dss:SignatureObject`:

- When the generation of a CMS based signature is requested, the base-64 encoded signature MUST be present in the `dss:Base64Signature` element.

- When the generation of an enveloping or detached XMLSig based signature is requested, this element will contain a `ds:Signature` element.

3.3.2.3 Element <DocumentWithSignature>

This element will only appear if an enveloped XML signature is requested.

3.4 Profile of Verifying Protocol

3.4.1 Element <VerifyRequest>

This clause specifies the profile for the contents of the `dss:VerifyRequest` when used for:

- Requesting verification of advanced signatures.
- Requesting verification of advanced signatures AND update of signatures to other predefined forms.

3.4.1.1 Attribute Profile

The value for the `Profile` attribute, indicating the concrete sub-profile of this abstract profile, MUST be present.

3.4.1.2 Element <OptionalInputs>

This profile specifies restrictions for the following possible children of `dss:OptionalInputs` element:

`<dss:ReturnUpdatedSignature>`. This element SHALL be present when the client requests verification of a signature and update to other predefined form of advanced signature.

3.4.1.2.1 Element <ReturnUpdatedSignature>

This element MUST be present when the client requests verification of a signature and update to a predefined form of advanced signature.

The `Type` attribute identifies the advanced signature form requested.

Acceptable predefined values for this attribute are the URIs specified in table 1 corresponding to the following forms predefined in [TS101733] and **[XAdES]**: XAdES-T/ES-T, XAdES-C/ES-C, XAdES-X/ES-X,XAdES-X-L,ES-X-L, XAdES-A, ES-A.

Should other standard or proprietary specification define new signature forms and their corresponding URIs, concrete sub-profiles of this abstract profile could be defined for giving support to their verification and update.

When the requested form allows for different contents, the server MUST decide the specific contents of the updated signature delivered, according to its configuration and settings.

4 Element <VerifyResponse>

4.1.1.1 Element <OptionalOutputs>

This profile specifies restrictions for the following optional outputs:

`<dss:UpdatedSignature>`. This element MUST be present in a successful response of a request containing `<dss:ReturnUpdatedSignature>`.

No additional restrictions are applied by this profile to the contents of any additional outputs.

4.1.1.1.1 Element <UpdatedSignature>

The content of the `dss:UpdatedSignature` will be a `dss:SignatureObject` element profiled according to the following rules:

- When the update of a CMS based signature is requested, the base-64 encoded signature itself MUST be present in the dss:Base64Signature element.
- When the update of a XMLSig based signature is requested, one of the following elements MUST appear:
- The ds:Signature containing a XMLSig based signature.
- The dss:SignaturePtr pointing to the XMLSig based signature embedded in one of the input documents.

5 XML Advanced Electronic Signatures concrete Profile

5.1 Overview

This concrete profile supports operations within each phase of the lifecycle of XML Advanced Electronic Signature based on **[XMLSig]** such as specified in **[XAdES]**. It will then provide all the features related to XAdES signatures that are specified in the abstract profile defined in section 3.

For the generation of XAdES signatures, the following operations apply:

- SignRequest. This operation supports requests for:
    - Generating predefined advanced signature forms as defined in **[XAdES]**.
    - Generating XML signatures incorporating specific signed/unsigned properties whose combination does not fit any predefined XAdES signature form. In such cases, the form MUST have been defined in a proprietary specification and MUST be identified by one URI.
    - SignResponse. This operation supports delivery of:
    - Predefined advanced signature forms as defined in **[XAdES]**.
    - XML signatures with specific properties whose combination does not fit any predefined XAdES signature form. In such cases, the form MUST have been defined in a proprietary specification and MUST be identified by one URI.

For verification [and updating] of XAdES signatures the following operations apply:

- VerifyRequest. This operation supports requests for:
    - Verifying a predefined XAdES signature form.
    - Verifying XML signatures incorporating specific properties whose combination does not fit any predefined XAdES signature form.
    - Verifying any of the signatures mentioned above PLUS updating them by addition of additional properties (time-stamps, validation data, etc) leading to a predefined XAdES form.
    - Verifying a long-term advanced signature in a certain point of time.
- VerifyResponse. This operation supports delivery of:
    - Advanced signature verification result of signatures mentioned above.
    - Advanced signature verification result PLUS the updated signatures as requested.
    - Updated signatures as requested.

5.2 Profile features

5.2.1 Identifier

`urn:oasis:names:tc:dss:1.0:profiles:XAdES.`

5.2.2 Scope

This document profiles the DSS abstract profile defined in section 3 of the present document.

5.2.3 Relationship To Other Profiles

The profile in this section is based on the abstract profile for Advanced Electronic Signatures defined in section 3.

5.2.4 Signature Object

This profile supports the creation and verification of XML advanced signatures as defined in **[XAdES]]**.

This profile also supports verification and update of advanced signatures by addition of unsigned properties (time-stamps and different types of validation data), as specified in **[XAdES]**

5.2.5 Transport Binding

This profile does not specify or constrain the transport binding.

5.2.6 Security Binding

This profile does not specify or constrain the security binding.

5.3 Profile of Signing Protocol

The present profile allows requesting:

- Predefined forms of advanced electronic signatures as defined in **[XAdES]**.
- Other forms of signatures based in **[XMLSig]** defined in other specifications,

In both cases, the specific requested form will be identified by an URI.

According to this profile, the following predefined advanced signature forms defined in **[XAdES]** MAY be requested: XAdES-BES, XAdES-EPES, XAdES-T, XAdES-C, XAdES-X, XAdES-X-L., and XAdES-A.

In addition, the present profile provides means for requesting incorporation in any of the aforementioned forms any of the following properties: SigningTime, CommitmentTypeIndication, SignatureProductionPlace, SignerRole, IndividualDataObjectTimeStamp, AllDataObjectTimeStamp and DataObjectFormat.

Other electronic signature forms based in **[XMLSig]** defined elsewhere MAY also be requested using the mechanisms defined in this profile.

5.3.1 Element <SignRequest>

5.3.1.1 Attribute Profile

`urn:oasis:names:tc:dss:1.0:profiles:XAdES.`

5.3.1.2 Element <OptionalInputs>

None of the optional inputs specified in the **[DSS Core]** are precluded in this abstract profile. It only constrains some of them and specifies additional optional inputs.

5.3.1.2.1 Element <SignatureType>

This element is MANDATORY. Its vaule MUST be:

`urn: ietf:rfc:3275`

5.3.1.2.2 Element <SignatureForm>

Usage of these elements is according to what is stated in section 3.3.1.1.2 .

5.3.1.2.3 Optional inputs < ClaimedIdentity> / <KeySelector>

Usage of these elements is according to what is stated in section 3.3.1.1.3.

5.3.1.2.4 Element <AddTimeStamp>

Usage of these elements is according to what is stated in section 3.3.1.1.4.

5.3.1.2.5 Element <SignedProperties>

5.3.1.2.5.1 Requesting SigningTime

Usage of these elements is according to what is stated in section 3.3.1.1.5.1.

5.3.1.2.5.2 Requesting CommitmentTypeIndication

The value for `<Identifier>` element is the one defined in section 3.3.1.1.5.2:

`urn:oasis:names:tc:dss:1.0:profiles:XAdES:CommitmentTypeIndication`

When the value of the commitment is established by the requester, the `<Value>` element MUST contain a `<Commitment>` element as defined in section 3.3.1.1.5.2 with the `<xades:CommitmentTypeIndication>` child.

5.3.1.2.5.3 . Requesting SignatureProductionPlace

Usage of these elements is according to what is stated in section 3.3.1.1.5.3

5.3.1.2.5.4 Requesting SignerRole

Value for `<Identifier>` element:

`urn:oasis:names:tc:dss:1.0:profiles:XAdES:SignerRole`

When the value of the role is fixed by the requester, the `<Value>` element MUST contain a `<SignerRole>` element as defined in section 3.3.1.1.5.4 with the `<xades:SignerRole>` child.

5.3.1.2.5.5 Requesting <xades:IndividualDataObjectTimeStamp>

Usage of these elements is according to what is stated in section 3.3.1.1.5.5.

5.3.1.2.5.6 Requesting data objects format

Value for `<Identifier>` element:

`urn:oasis:names:tc:dss:1.0:profiles:XAdES:DataObjectFormat`

When the value of the data object formats are fixed by the requester, the `<Value>` element MUST contain a `<DocsFormat>` element as defined in section 3.3.1.1.5.6 with the `<DocFormat>` child.

5.3.2 Element <SignResponse>

This section profiles the `dss:SignResponse` element for the requests profiled in clause 5.3.1.

5.3.2.1 Element <Result>

This profile does not apply any restriction to the `dss:Result` element.

5.3.2.2 Element <SignatureObject>

The content of this element MUST be one of the following:

- A `ds:Signature` element containing a XMLSig based signature.
- A `dss:SignaturePtr` pointing to the XMLSig based signature embedded in one of the output documents. In such a case, the optional output element, containing the signature created, MUST be present within the `dss:DocumentWithSignature` element

5.4 Profile of Verifying Protocol

5.4.1 Element <VerifyRequest>

5.4.1.1 Attribute Profile

`urn:oasis:names:tc:dss:1.0:profiles:XAdES.`

5.4.1.2 Element <OptionalInputs>

5.4.1.2.1 Element <ReturnUpdatedSignature>

Usage of these elements is according to what is stated in section 3.4.1.2.1.

5.4.1.3 Element <SignatureObject>

The `dss:SignatureObject` element will have one of the following contents:

- A ds:Signature containing a XMLSig based signature.
- A dss:SignaturePtr pointing to the XMLSig based signature embedded in one of the inputdocuments.

5.4.2 Element <VerifyResponse>

5.4.2.1 Element <OptionalOutputs>

Usage of these elements is according to what is stated in section 4.1.1.1.

5.4.2.1.1 Element <UpdatedSignature>

The content of the `dss:UpdatedSignature` will be a `dss:SignatureObject` element with one of the following contents:

- A ds:Signature containing a XMLSig based signature.
- A dss:SignaturePtr pointing to the XMLSig based signature embedded in one of the inputdocuments.

5.5 Profile Bindings

5.5.1 Transport Bindings

Messages transported in this profile MAY be transported by the HTTP POST Transport Binding and the SOAP 1.2 Transport Binding defined in **[DSSCore]**.

5.5.2 Security Bindings

5.5.2.1 Security Requirements

This profile MUST use security bindings that:

- Authenticates the requester to the DSS server
- Authenticates the DSS server to the DSS client
- Protects the integrity or a request, response and the association of response to the request.
- Optionally, protects the confidentiality of a request and response.

The following MAY be used to meet these requirements.

5.5.2.2 TLS X.509 Mutual Authentication

This profile is secured using the TLS X.509 Mutual Authentication Binding defined in **[DSSCore]**.

6 CMS-based Advanced Electronic Signature profile

6.1 Overview

This concrete profile supports operations within each phase of the lifecycle of CMS based Advanced Electronic Signature based on [**RFC 3369**] such as specified in **[TS 101733]**. It will then provide all the features related to TS 101733 signatures that are specified in the abstract profile defined in section 3.

For the generation of TS101733 signatures, the following operations apply:

- SignRequest. This operation supports requests for:
  - Generating predefined advanced signature forms as defined in [TS101733].
  - Generating CMS signatures incorporating specific signed/unsigned attributes whose combination does not fit any predefined **[TS 101733]** signature forms. In such cases, the form MUST have been defined in a proprietary specification and MUST be identified by one URI.
  - SignResponse. This operation supports delivery of:
  - Predefined advanced signature forms as defined in [TS101733].
  - CMS signatures incorporating specific signed attributes whose combination does not fit any predefined **[TS 101733]** signature form. In such cases, the form MUST have been defined in a proprietary specification and MUST be identified by one URI.

For verification [and updating] of signatures as specified in **[TS 101733]** the following operations apply:

- VerifyRequest. This operation supports requests for:
  - Verifying a predefined **[TS 101733]** signature form.
  - Verifying CMS signatures incorporating specific attributes whose combination does not fit any predefined **[TS 101733]** signature form.
  - Verifying any of the signatures mentioned above PLUS updating them by addition of additional attributes (time-stamps, validation data, etc) leading to a predefined **[TS 101733]** form.
  - Verifying a long-term advanced signature in a certain point of time.
- VerifyResponse. This operation supports delivery of:
  - Advanced signature verification result of signatures mentioned above.
  - Advanced signature verification result PLUS the updated signatures as requested.
  - Updated signatures as requested.

6.2 Profile features

6.2.1 Identifier

`urn:oasis:names:tc:dss:1.0:profiles:CAdES.`

6.2.2 Scope

This document profiles the DSS abstract profile defined in section 3 of the present document.

6.2.3 Relationship To Other Profiles

The profile in this document is based on the abstract profile for Advanced Electronic Signatures defined in section 3.

6.2.4 Signature Object

This profile supports the creation and verification of CMS based advanced signatures as defined in **[TS101733]]**.

This profile also supports verification and update of advanced signatures by addition of unsigned properties (time-stamps and different types of validation data), as specified in **[TS101733]**

6.2.5 Transport Binding

This profile does not specify or constrain the transport binding.

6.2.6 Security Binding

This profile does not specify or constrain the security binding.

6.3 Profile of Signing Protocol

The present profile allows requesting:

- Predefined forms of advanced electronic signatures as defined in **[TS 101733]**.
- Other forms of signatures based in [RFC 3369] defined in other specifications,

In both cases, the specific requested form will be identified by an URI.

According to this profile, the following predefined advanced signature forms defined in **[TS 101733]** MAY be requested: BES, EPES, ES-T, ES-C, ES-X, ES-X-L, and ES-A

In addition, the present profile provides means for requesting incorporation in any of the aforementioned forms any of the following properties: SigningTime, CommitmentTypeIndication, SignatureProductionPlace, SignerRole, IndividualDataObjectTimeStamp, AllDataObjectTimeStamp and DataObjectFormat.

Other electronic signature forms based in **[RFC 3369]**, defined elsewhere, MAY also be requested using the mechanisms defined in this profile.

6.3.1 Element <SignRequest>

6.3.1.1 Attribute Profile

`urn:oasis:names:tc:dss:1.0:profiles:CAdES.`

6.3.1.2 Element <OptionalInputs>

None of the optional inputs specified in the **[DSS Core]** are precluded in this abstract profile. It only constrains some of them and specifies additional optional inputs.

6.3.1.2.1 Element <SignatureType>

This element is MANDATORY. Its vaule MUST be:

`urn: ietf:rfc:3369`

6.3.1.2.2 Element <SignatureForm>

Usage of these elements is according to what is stated in section 3.3.1.1.2 .

6.3.1.2.3 Optional inputs < ClaimedIdentity> / <KeySelector>

Usage of these elements is according to what is stated in section 3.3.1.1.3.

6.3.1.2.4 Element <AddTimeStamp>

Usage of these elements is according to what is stated in section 3.3.1.1.4.

6.3.1.2.5 Element <SignedProperties>

This section profiles section 3.3.1.1.5.

6.3.1.2.5.1 Requesting SigningTime

Usage of these elements is according to what is stated in section 3.3.1.1.5.1.

6.3.1.2.5.2 Requesting CommitmentTypeIndication

The value for `<Identifier>` element is the one defined in section 3.3.1.1.5.2:

`urn:oasis:names:tc:dss:1.0:profiles:XAdES:CommitmentTypeIndication`

When the value of the commitment is established by the requester, the `<Value>` element MUST contain a `<Commitment>` element as defined in section 3.3.1.1.5.2 with the `<BinaryValue>` child containing the base64encoding of **CommitmentTypeIndication** ASN.1 type as specified in [TS101733], DER-encoded.

6.3.1.2.5.3 . Requesting SignatureProductionPlace

Usage of these elements is according to what is stated in section 3.3.1.1.5.3

6.3.1.2.5.4 Requesting SignerRole

Value for `<Identifier>` element:

`urn:oasis:names:tc:dss:1.0:profiles:XAdES:SignerRole`

When the value of the role is fixed by the requester, the `<Value>` element MUST contain a `<SignerRole>` element as defined in section 3.3.1.1.5.4 with the `<BinaryValue>` child containing the base64encoding of **SignerAttribute** ASN.1 type as specified in [TS101733], DER-encoded.

6.3.1.2.5.5 Requesting data objects format

Value for `<Identifier>` element:

`urn:oasis:names:tc:dss:1.0:profiles:XAdES:DataObjectFormat`

When the value of the data object formats are fixed by the requester, the `<Value>` element MUST contain a `<DocsFormat>` element as defined in section 3.3.1.1.5.6 with the `<BinaryValue>` child containing the base64encoding of **ContentHints** ASN.1 type as specified in [TS101733], DER-encoded.

6.3.2 Element <SignResponse>

This section profiles the `dss:SignResponse` element for the requests profiled in clause 5.3.1.

6.3.2.1 Element <Result>

This profile does not apply any restriction to the `dss:Result` element.

6.3.2.2 Element <SignatureObject>

The `dss:SignatureObject` MUST contain the `dss:Base64Signature` child with a CMS based signature base-64 encoded.

6.4 Profile of Verifying Protocol

6.4.1 Element <VerifyRequest>

6.4.1.1 Attribute Profile

`urn:oasis:names:tc:dss:1.0:profiles:CAdES.`

6.4.1.2 Element <OptionalInputs>

6.4.1.2.1 Element <ReturnUpdatedSignature>

Usage of these elements is according to what is stated in section 3.4.1.2.1.

6.4.1.3 Element <SignatureObject>

The `dss:SignatureObject` element MUST contain the `dss:Base64Signature` child with a CMS based signature base64 encoded.

6.4.2 Element <VerifyResponse>

6.4.2.1 Element <OptionalOutputs>

Usage of these elements is according to what is stated in section 4.1.1.1.

6.4.2.1.1 Element <UpdatedSignature>

The content of the `dss:UpdatedSignature` will be a `dss:SignatureObject` element with one of the following contents:

- A dss:Base64Signature element with the CMS based signature base64 encoded.

6.5 Profile Bindings

6.5.1 Transport Bindings

Messages transported in this profile MAY be transported by the HTTP POST Transport Binding and the SOAP 1.2 Transport Binding defined in **[DSSCore]**.

6.5.2 Security Bindings

6.5.2.1 Security Requirements

This profile MUST use security bindings that:

- Authenticates the requester to the DSS server
- Authenticates the DSS server to the DSS client
- Protects the integrity or a request, response and the association of response to the request.

- Optionally, protects the confidentiality of a request and response.

The following MAY be used to meet these requirements.

6.5.2.2 TLS X.509 Mutual Authentication

This profile is secured using the TLS X.509 Mutual Authentication Binding defined in **[DSSCore]**.

7

8 Identifiers defined in this specification

8.1 Predefined advanced electronic signature forms identifiers

The table below shows the URIs for standard forms of advanced electronic signature:

Advanced signature FORM	**URI**
XAdES-BES BES	`urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:BES`
XAdES-EPES EPES	`urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:EPES`
XAdES-T ES-T	`urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:ES-T`
XAdES-C ES-C	`urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:ES-C`
XAdES-X ES-X	`urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:ES-X`
XAdES-X-L ES-X-L	`urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:ES-X-L`
XAdES-A ES-X-A	`urn:oasis:names:tc:dss:1.0:profiles:XAdES:forms:ES-A`

Table 1.

9 Editorial Issues

1. The current text only allows request of signature forms identified by an URI defined somewhere. There are in **[XAdES]** and in **[TS 101733]** certain forms that allow different combination of properties. If we leave the standard as it is now, we are leaving the server to decide which combination of properties to select or we assume that any combination of properties in each form will have its unique identifier.

   Report: Comments received in favor of leave the text as it is. Forms requested by URI. If possible different combinations of properties, the server decides

   Satus: CLOSED if no objections.

2. Should not this abstract profile also allow requesting the updating of the signature by enumeration of the properties desired?. In this way, this profile would allow both mechanisms to update signatures: by identifying the form or by identifying the set of unsigned properties.

   Report: Comments received in favor of leave the text as it is

   Satus: CLOSED if no objections.

3. Section 3.3.1.1.3. A proposal has been made to associate the presence in the request of `<ClaimedIdentity>` to the production of a signature where the signer's certificate is protected by `<xades:SigningCertificate>` property, and if instead, `<KeySelector>` is present then the signature will contain a signed `<ds:KeyInfo>` with a `<ds:X509Certificate>` element. I would prefer not linking the information required by the server to gain access to the signer's certificate with the mechanism selected in the signature for protecting this certificate.

   Report: Comments received in favor of leave the text as it is

   Satus: CLOSED if no objections.

4. Section 3.3.1.1.4. Proposal for suppress any mention to time-marking and capabilities for the protocol to differentiate, when requesting ES-T or XAdES-T a time-stamp or a time-mark.

   Satus: CLOSED. Leave as it is now.

5. Sections 3.3.1.1.5.2, 3.3.1.1.5.3, 3.3.1.1.5.4, and 3.3.1.1.5.6. In the former version the values of the signed properties passed to the server was left open. A proposal was made to force them to be elements whose types would be those defined in **[XAdES]**. The current version allows them to be elements of types defined in [**XAdES]** when requesting XML Sig based signatures OR `<BinaryValue>` containing the base 64 encoded value of the corresponding property defined in **[TS 101733]**, DER-encoded, since both kind of signatures may be requested to the server.

   Report: Comments received accepting changes.

   Satus: CLOSED.

6. Section 3.3.1.1.5.5. A proposal was made to use `<dss:DocumentBaseType>` for supporting request of `individualDataObjectsTimeStamp` property.

   `<dss:DocumentBaseType>` does contain much more information than the required for pointing to the data object to be time-stamped before signing and give indication of the `Id` attribute that its corresponding `<ds:Reference>` must have. In fact the `<DocsToBeTimeStamped>` element has quite a lot of commonality with `<dss:SignedReference>`, except the `<ds:Transforms>`. But as there is not a type definition, a new schema definition must be generated.

   Report: Comments received accepting changes.

Satus: CLOSED.

7. Previous version explicitly mentioned that this profile would allow for requesting verification of signatures in one specific time. As this is something covered by the core, this mention has been suppressed. Comments have been raised relating this time with the time appearing in <xades:SigningTime>. From my point of view one thing is the time that the signer claims to have signed and a different issue is the time when the verifier verifies the signature.

   Report: Comments received accepting changes.

   Satus: CLOSED.

8. A new element has been defined as optional input to allow requesting the server update of a signature without verification.

   Report: Comments pointing out the weakness that this element could bring in future as it would mean that some servers could update signatures without verifying them, and this would bring other ways to ascertain that a updated signature had actually been verified.

   Decision: to suppress this feature. The corresponding section has been eliminated from this version.

   Status: CLOSED.

9. Requesting CounterSignature is not explicitly mentioned here. I think that it would always be possible to use the core to get a copy of `<ds:SignatureValue>` element from the `<ds:Signature>` and send it to the server using core protocol. Does anyone see any reason for including a new `<GenerateAsCounterSignature>` optional input?

   Report: Comments received suggesting not go into this issue.

   Satus: CLOSED until further consideration.

10. The comment has been made that in order to identify the signer's key and gain access to the signer's certificate, there may be mechanisms out of the dss-core, so that it does not make sense to make mandatory to use <dss:ClaimedIdentity> or <dss:KeySelector> in section 3.3.1.1.3. I agree with that comment. Text has been accordingly modified.

    Satus: CLOSED.

11. Section 3.3.1.1.5.5. A comment has been made to suppress optional `RefId` attribute and leave the server decide. So far I have kept it, as I still think that it is worth to give the client the opportunity to request it. In addition I have added text clarifying the relationship with `RefId` attribute in `<dss:SignedReferences>`.

    Report: Comments received accepting text as it is.

    Satus: CLOSED.

10 References

10.1 Normative

[Core-XSD] T. Perrin et al. *DSS Schema*. OASIS, **(MONTH/YEAR TBD)**

[DSSCore] T. Perrin et al. *Digital Signature Service Core Protocols and Elements*. OASIS, **(MONTH/YEAR TBD)**

[RFC 2119] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. IETF RFC 2396, August 1998.

http://www.ietf.org/rfc/rfc2396.txt.

[TS 101733] Advanced Electronic Signatures. ETSI TS 101 733.

[XAdES] XML Advanced Electronic Signatures. ETSI TS 101 903, February 2002 (shortly to be reissued).

[XML-ns] T. Bray, D. Hollander, A. Layman. *Namespaces in XML.* W3C Recommendation, January 1999.

http://www.w3.org/TR/1999/REC-xml-names-19990114

[XMLSig] D. Eastlake et al. *XML-Signature Syntax and Processing.* W3C Recommendation, February 2002.

http://www.w3.org/TR/1999/REC-xml-names-19990114

[RFC 2634] P. Hoffman (ed.). Enhanced Security Services for S/MIME, June 1999.

[RFC 3369] Message Syntax (CMS). R. Housley. August 2002.

925 Appendix A. Revision History

Rev	Date	By Whom	What
wd-01	2004-03-08	Juan Carlos Cruellas	Initial, incomplete version: SignRequest for predefined forms + optional properties.
wd-02	2004-03-08	Juan Carlos Cruellas	Second version of the initial version: it incorporates SignRequest-SignResponse and VerifyRequest-VerifyResponse. No capability for requesting individually any property. This is still an on-going discussion.
wd-03	2004-06-18	Juan Carlos Cruellas	Third version. Quite a lot of editorial work done. No capability for requesting individually any property. This is still an on-going discussion.
wd-04	2004-08-09	Juan Carlos Cruellas	Fourth version: Suppressed <UpdateSignatureOnly> element. So far: signature forms identified by URI. Not possibility of requesting properties by enumeration. Solved most of editorial issues. Small editorial changes.
wd-05	2004-10-08	Juan Carlos Cruellas	Fifth version: Addition of two concrete sub-profiles: one for XAdES and the other for TS 101733
wd-06	2004-11-09	Juan Carlos Cruellas	Sixth version: Addition of bindings for concrete profiles. Additional changes from comments raised before voting as a CD

Appendix B. Notices

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification, can be obtained from the OASIS Executive Director.

OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to implement this specification. Please address the information to the OASIS Executive Director.

This document and the information contained herein is provided on an “AS IS” basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.