Asynchronous Processing Abstract Profile of the OASIS Digital Signature Services Version 1.0

Committee Specification

13 February 2007

Specification URIs:
This Version:
http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-asynchronous_processing-spec-cs-v1.0-r1.html
http://docs.oasis-open.org/dss/v1.0/dss-profiles-asynchronous_processing-spec-cs-v1.0-r1.pdf

Latest Version:
http://docs.oasis-open.org/dss/v1.0/dss-profiles-asynchronous_processing-spec-cs-v1.0-r1.html
http://docs.oasis-open.org/dss/v1.0/dss-profiles-asynchronous_processing-spec-cs-v1.0-r1.pdf

Technical Committee:
OASIS Digital Signature Services TC

Chair(s):
Nick Pope, Thales eSecurity
Juan Carlos Cruellas, Centre d'aplicacions avançades d'Internet (UPC)

Editor(s):
Andreas Kuehne, individual

Related work:
This specification is related to:

- oasis-dss-core-spec-cs-v1.0-r1

Abstract:
This document defines protocol profiles and processing profiles for the purpose of creating and verifying German Signature Law signatures.

Status:
This document was last revised or approved by the OASIS Digital Signature Services TC on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at http://www.oasis-open.org/committees/dss.

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 Technical Committee web page (http://www.oasis-open.org/committees/dss/ipr.php.

The non-normative errata page for this specification is located at http://www.oasis-open.org/committees/dss.

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.

Table of Contents

1 Introduction

This is an *abstract profile*. Further profiles will build on this one to provide a basis for implementation and interoperability.

This draft profiles the OASIS DSS core protocol for asynchronous processing. Although most applications of the OASIS Digital Signature Service supply the results immediately there is a demand for deferred delivering of results. E.g. the German Signature Law explicitly requires the commitment of the certificate holder or at least a time slot for the certificate holder to deny the signing request **[SigG]**.

Another use case for a asynchronous protocol may arise in a verification request if a minimum latency between creation and verification has to be respected.

This profile is intended to be generic, so it may be combined with other profiles freely.

A protocol for asynchronous processing is already defined in the XML Key Management Specification **[XKMS]**. This profile borrows ideas from the XKMS protocol for the OASIS Digital Signature Service.

The following sections describe how to understand the rest of this document.

1.1 Terminology

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 Normative References

[Core-XSD] S. Drees et al. *DSS Schema*. OASIS, February 2007

[DSSCore] S. Drees et al. *Digital Signature Service Core Protocols and Elements*. OASIS, February 2007

[RFC 2119] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997. .

[XML-ns] T. Bray, D. Hollander, A. Layman. *Namespaces in XML*. http://www.w3.org/TR/1999/REC-xml-names-19990114, W3C Recommendation, January 1999.

[XMLSig] D. Eastlake et al. *XML-Signature Syntax and Processing*. http://www.w3.org/TR/1999/REC-xml-names-19990114, W3C Recommendation, February 2002.

[SigG] Framework for Electronic Signatures, Amendment of Further Regulations Act (Signaturgesetz – SigG), 21 May 2001. http://www.regtp.de/imperia/md/content/tech_reg_t/digisign/119.pdf

[XKMS2] Phillip Hallam-Baker *XML Key Management* Specification (XKMS 2.0) http://www.w3.org/TR/2004/CR-xkms2-20040405/, W3C Candidate Recommendation, 5 April 2004.

1.3 Non-Normative References

1.4 Namespaces

The structures described in this specification are contained in the schema file **[XYZ-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:

```
urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0
```

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 `async:` stands for this profiles namespace **[Core-XSD]**.
- 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]**.

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]**.

1.5 Overview (Non-normative)

This profile defines a simple mechanism for asynchronous signing and verification requests. This profile is similar to the asynchronous processing protocol defined in the XKMS spec **[XKMS]**.

In the first call the client supplies its input values as defined in the core and the applied profiles. The server may reply synchronously with the appropriate result.

On the other hand the server may reply with an 'empty' result, giving the `<ResultMajor>` code 'Pending' and a `<async:ResponseID>` element as an `<OptionalOutput>`. The server generates the value of the `<async:ResponseID>` on its own.

The client may initiate a <PendingRequest> call from time to time with the `<async:ResponseID>` of the initial response included in the `<async:ResponseID>` element within the `<dss:OptionalInputs>`.

When the server finally succeeds with its processing the results will be delivered to the client with its next polling call. In this case the `<ResultMajor>` must not be `'Pending'` but the `<ResultMajor>` resulting from the request processing.

A notification mechanism isn't defined yet, but may be subject to following versions of this profile.

68 # 2 Profile Features

69 ## 2.1 Identifier

70 urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing

71 Add an `<AdditionalProfile>` element containing this URI to use this profile.

72 ## 2.2 Scope

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

74 ## 2.3 Relationship To Other Profiles

75 This profile is based directly on the **[DSSCore]**.
76 This profile is an abstract profile which is not implementable directly.
77 This profile is intended to be combined with other profiles freely.

78 ## 2.4 Signature Object

79 This profile does not specify or constrain the type of signature object.

80 ## 2.5 Transport Binding

81 This profile does not specify or constrain the transport binding.

82 ## 2.6 Security Binding

83 This profile does not specify or constrain the security binding.

3 Polling Protocol

The polling protocol extends the core protocol using the <PendingRequest> element for initiating a polling request. This is different from the initial request because the request specific data was already transmitted.

3.1 Element <PendingRequest>

The <PendingRequest> element is sent by the client to request the result from a pending signature or verification initiated earlier. It contains the following attributes and elements inherited from <RequestBaseType> :

RequestID [Optional]

This attribute is used to correlate requests with responses. When present in a request, the server MUST return it in the response.

Profile [Optional]

This attribute indicates a particular DSS profile. It may be used to select a profile if a server supports multiple profiles, or as a sanity-check to make sure the server implements the profile the client expects. In this special case of a <PendingRequest> the required profile information is already defined within the initial call to the server. So Profile MUST be omitted in a <PendingRequest>. Consequently there MUST NOT be any <AdditionalProfile> optional input elements in a <PendingRequest>.

<OptionalInputs> [Optional]

Any additional inputs to the request. This element may be used e.g. for authentication data.

In addition to <RequestBaseType> the <PendingRequest> element defines the following <ResponseID> element:

3.1.1 Element <OptionalInputs>

This profile defines the new input element of <async:ResponseID>.

<async:ResponseID>
To correlate subsequent <PendingRequest> calls to the initial request the <async:ResponseID> element is introduced by this profile. The client MUST take care of the value returned by the initial <SignRequest> in <async:ResponseID>.

3.2 Element <Response>

The <PendingRequest> may response with a generic <Response> in cases where the service is unable to specialise down to <SignResponse> or <VerificationResponse>.

116 This will happen when the service doesn't recognise the given ResponseID. The <ResultMinor> is
117 set to the special value of ResponseIdUnknown.

118

119 Urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultminor:ResponseIdU
120 nknown

121

122 The <ResultMajor> code in this case is RequesterError . This result code shows up only in response
123 to a <PendingRequest>.

124 In the case of successful interpretation of the ResponseID attribute the service returns a
125 <SignResponse> or <VerifyResponse> as intended by the initial request.

4 Profile of Signing Protocol

4.1 Element <SignRequest>

No additional elements of <SignRequest> defined by this profile.

4.2 Element <SignResponse>

4.2.1 Element <ResultMajor>

This profile defines the additional `<ResultMajor>` code, which may show up in response to a <SignRequest> or <PendingRequest>:

```
urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending
```

This result value means that the operation did not finish yet. Subsequent requests may return this result code again. After the server has finished the operation the call will return the signing result indicated by the `urn:oasis:names:tc:dss:1.0:resultmajor:Success` value or an error code.

In case an asynchronous service is unable to reply in a synchronous manner and a requests to this service is made without profiling the call as asynchronous (using the given profile identifier within the `Profile` attribute or the `<AdditionalProfiles>` element), the service returns a `<ResultMajor>` of `RequesterError` and a `<ResultMinor>` of:

```
urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultminor:asynchronousOnly
```

4.2.2 Element <OptionalOutputs>

This profile defines the new optional output element of `<async:ResponseID>`.

<async:ResponseID>
To correlate subsequent `<PendingRequest>` calls to the initial request the `<async:ResponseID>` element is introduced by this profile. The service will generate a suitable value on its own behalf. So the client MUST take care of the value returned in `<async:ResponseID>` for subsequent `<PendingRequest>`.

If the server returns the `<ResultMajor>` code

`urn:oasis:names:tc:dss:1.0:`profiles:asynchronousprocessing:`resultmajor:Pending`
the contents of the `<OptionalOutputs>` element children other than `<async:ResponseID>` are undefined.

If the server returns the `<ResultMajor>` code

`urn:oasis:names:tc:dss:1.0:resultmajor:Success`
the `<OptionalOutputs>` MUST contain the results defined by the accompanying profiles as expected in synchronous operation.

4.2.3 Element <SignatureObject>

If the server returns the `<ResultMajor>` code

`urn:oasis:names:tc:dss:1.0:`profiles:asynchronousprocessing:`resultmajor:Pending`
the content of the `<SignatureObject>` element is undefined.

If the server returns the `<ResultMajor>` code

`urn:oasis:names:tc:dss:1.0:resultmajor:Success`
the `<SignatureObject>` MUST contain the results defined by the accompanying profiles as expected in synchronous operation.

5 Profile of Verifying Protocol

5.1 Element <VerifyRequest>

5.1.1 Element <OptionalInputs>

This profile doesn't interfere with the element defined from **[DSSCore]**.

5.1.2 Element <SignatureObject>

This profile doesn't interfere with the element defined from **[DSSCore]**.

5.1.3 Element <InputDocuments>

This profile doesn't interfere with the element defined from **[DSSCore]**.

5.2 Element <VerifyResponse>

5.2.1 Element <ResultMajor>

This profile defines the additional <ResultMajor> code, which may show up in response to a <SignRequest> or <PendingRequest>:

```
urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending
```

This result value means that the operation did not finish yet. Subsequent requests may return this result code again. After the server has finished the operation the call will return the verification result indicated by the urn:oasis:names:tc:dss:1.0:resultmajor:Success value or an error code.

In case an asynchronous service is unable to reply in a synchronous manner and a requests to this service is made without profiling the call as asynchronous (using the given profile identifier within the Profile attribute or the <AdditionalProfiles> element), the service returns a <ResultMajor> of RequesterError and a <ResultMinor> of:

```
urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultminor:asynchronousOnly
```

5.2.2 Element <OptionalOutputs>

This profile defines the new optional output element of <async:ResponseID>.

<async:ResponseID>
To correlate subsequent <PendingRequest> calls to the initial request the <async:ResponseID> element is introduced by this profile. The service will generate a suitable value on its own behalf. So the client MUST take care of the value returned in <async:ResponseID> for subsequent <PendingRequest>.

If the server returns the <ResultMajor> code

urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending

204 the contents of the <OptionalOutputs> element children other than <async:ResponseID> are
205 undefined.

206

207 If the server returns the <ResultMajor> code

208 urn:oasis:names:tc:dss:1.0:resultmajor:Success
209 the <OptionalOutputs> MUST contain the results defined by the accompanying profiles as expected
210 in synchronous operation.

211

A. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Trevor Perrin, individual
Pieter Kasselman, Betrusted
Tommy Lindbert, individual

B. Example – (Non-Normative)

Example of an initial signing request :

```
<dss:SignRequest Profile="urn:oasis:names:tc:dss:1.0:profile:dss_interop"
    RequestID="I0d2f1de663c75dc52f468e678af1bfd6"
    xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema">
  <dss:OptionalInputs>
    <dss:SignatureType>...</dss:SignatureType>
    <dss:AdditionalProfile>
      urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing
    </dss:AdditionalProfile>
  </dss:OptionalInputs>
  <dss:InputDocuments>
    <dss:Document ID="..." RefType="..." RefURI="...">
    ...
    </dss:Document>
  </dss:InputDocuments>
</dss:SignRequest>
```

The request above may result in an response like this :

```
<dss:SignResponse RequestID="I0d2f1de663c75dc52f468e678af1bfd6"
    Profile="urn:oasis:names:tc:dss:1.0:profile:dss_interop"
    xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema"
    xmlns:async="urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0">
  <dss:Result>
    <dss:ResultMajor>
      urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:resultmajor:Pending
    </dss:ResultMajor>
  </dss:Result>
  <dss:OptionalOutputs>
    <async:ResponseID>I517f0e98752098c7245f2892f59ef9fc</async:ResponseID>
  </dss:OptionalOutputs>
</dss:SignResponse>
```

The server return a <dss :ResultMajor> value ‘Pending’ with no Signature returned. So the client will send a PendingRequest using the value of <async:ResponseID> from this response. A PendingRequest may look like this :

```
<async:PendingRequest RequestID="If82506cfa678bedf2cdc1549f5970641"
    xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema"
    xmlns:async="urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0">
  <dss:OptionalInputs>
    <dss:AdditionalProfile>
      urn:oasis:names:tc:dss:1.0:profiles:asynchronousprocessing:1.0
    </dss:AdditionalProfile>
    <async:ResponseID>I517f0e98752098c7245f2892f59ef9fc</async:ResponseID>
  </dss:OptionalInputs>
</async:PendingRequest>
```

The server may respond with a <dss:ResultMajor> value ‘Pending’ again. But finally server side processing will be finished and the server replies such a Response :

```
<dss:SignResponse RequestID="If82506cfa678bedf2cdc1549f5970641"
    Profile="urn:oasis:names:tc:dss:1.0:profile:dss_interop"
    xmlns:dss="urn:oasis:names:tc:dss:1.0:core:schema">
  <dss:Result>
    <dss:ResultMajor>
      urn:oasis:names:tc:dss:1.0:resultmajor:Success
    </dss:ResultMajor>
  </dss:Result>
  <dss:SignatureObject>
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      ...
```

```
    </ds:Signature>
  </dss:SignatureObject>
</dss:SignResponse>
```