



XML Timestamping Profile of the OASIS Digital Signature Services Version 1.0

OASIS Standard

11 April 2007

Specification URIs:

This Version:

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-timestamping-spec-cs-v1.0-os.html>

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-timestamping-spec-cs-v1.0-os.pdf>

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-timestamping-spec-cs-v1.0-os.doc>

Latest Version:

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-timestamping-spec-cs-v1.0-os.html>

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-timestamping-spec-cs-v1.0-os.pdf>

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-timestamping-spec-cs-v1.0-os.doc>

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):

Trevor Perrin, individual

Juan Carlos Cruellas, Centre d'aplicacions avançades d'Internet (UPC)

Related work:

This specification is related to:

- [oasis-dss-core-spec-v1.0-os](#)

Abstract:

This document profiles the OASIS DSS core protocols for the purpose of creating and verifying XML-encoded time-stamps.

Status:

This document was last revised or approved by the membership of OASIS 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

37 Committee by using the “Send A Comment” button on the Technical Committee’s web
38 page at <http://www.oasis-open.org/committees/dss/>.

39 For information on whether any patents have been disclosed that may be essential to
40 implementing this specification, and any offers of patent licensing terms, please refer to
41 the Intellectual Property Rights section of the Technical Committee web page
42 (<http://www.oasis-open.org/committees/dss/ipr.php>).

43 The non-normative errata page for this specification is located at [http://www.oasis-
open.org/committees/dss/](http://www.oasis-
44 open.org/committees/dss/).

45 **Notices**

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

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

58 Copyright © OASIS® 1993–2007. All Rights Reserved.

59 This document and translations of it may be copied and furnished to others, and derivative works
60 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
61 published and distributed, in whole or in part, without restriction of any kind, provided that the
62 above copyright notice and this paragraph are included on all such copies and derivative works.
63 However, this document itself may not be modified in any way, such as by removing the copyright
64 notice or references to OASIS, except as needed for the purpose of developing OASIS
65 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
66 Property Rights document must be followed, or as required to translate it into languages other
67 than English.

68 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
69 successors or assigns.

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

75 The names "OASIS" are trademarks of OASIS, the owner and developer of this specification, and
76 should be used only to refer to the organization and its official outputs. OASIS welcomes
77 reference to, and implementation and use of, specifications, while reserving the right to enforce
78 its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for
79 above guidance.

80 Table of Contents

81	1	Introduction.....	5
82	1.1	Terminology	5
83	1.2	Normative References	5
84	1.3	Non-Normative References	5
85	1.4	Namespaces	5
86	2	Profile Features	6
87	2.1	Identifier	6
88	2.2	Scope	6
89	2.3	Relationship To Other Profiles	6
90	2.4	Signature Object	6
91	2.5	Transport Binding.....	6
92	2.6	Security Binding	6
93	3	Profile of Signing Protocol	7
94	3.1	Element <SignRequest>	7
95	3.1.1	Element <OptionalInputs>.....	7
96	3.1.2	Element <InputDocuments>.....	8
97	3.2	Element <SignResponse>	8
98	3.2.1	Element <Result>.....	8
99	3.2.2	Element <OptionalOutputs>	8
100	3.2.3	Element <SignatureObject>	8
101	4	Profile of Verifying Protocol.....	9
102	4.1	Element <VerifyRequest>	9
103	4.1.1	Element <OptionalInputs>.....	9
104	4.1.2	Element <SignatureObject>	9
105	4.1.3	Element <InputDocuments>.....	9
106	4.2	Element <VerifyResponse>	9
107	4.2.1	Element <Result>.....	9
108	4.2.2	Element <OptionalOutputs>	9
109	A.	Acknowledgements	11
110			

111 1 Introduction

112 The DSS signing and verifying protocols are defined in **[DSSCore]**. As defined in that document,
113 these protocols have a fair degree of flexibility and extensibility. This document profiles these
114 protocols to limit their flexibility and extend them in concrete ways. The resulting profile is
115 suitable for implementation and interoperability.

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

117 1.1 Terminology

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

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

126 1.2 Normative References

- | | | |
|-----|-------------------|--|
| 127 | [Core-XSD] | S. Drees et al. <i>DSS Schema</i> . OASIS, February 2007 |
| 128 | [DSSCore] | S. Drees et al. <i>Digital Signature Service Core Protocols and Elements</i> .
129 OASIS, February 2007 |
| 130 | [TST-XSD] | T. Perrin et al. <i>Timestamping Profile Schema</i> , OASIS, , February 2007 |
| 131 | [RFC 2119] | S. Bradner. <i>Key words for use in RFCs to Indicate Requirement Levels</i> .
132 http://www.ietf.org/rfc/rfc2396.txt , IETF RFC 2396, August 1998. |
| 133 | [XML-ns] | T. Bray, D. Hollander, A. Layman. <i>Namespaces in XML</i> .
134 http://www.w3.org/TR/1999/REC-xml-names-19990114 , W3C
135 Recommendation, January 1999. |
| 136 | [XMLSig] | D. Eastlake et al. <i>XML-Signature Syntax and Processing</i> .
137 http://www.w3.org/TR/1999/REC-xml-names-19990114 , W3C
138 Recommendation, February 2002. |

139 1.3 Non-Normative References

140 1.4 Namespaces

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

144 This schema is associated with the following XML namespace:

145 `urn:oasis:names:tc:dss:1.0:profiles:TimeStamp:schema#`

146 Conventional XML namespace prefixes are used in this document:

- 147 • The prefix `dss`: stands for the DSS core namespace **[Core-XSD]**.

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

151

152 2 Profile Features

153 2.1 Identifier

154 urn:oasis:names:tc:dss:1.0:profiles:timestamping

155 2.2 Scope

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

157 2.3 Relationship To Other Profiles

158 This profile is based directly on the [DSSCore].

159 2.4 Signature Object

160 This profile supports the creation and verification of isolated `<dss:Timestamp>` elements as
161 defined in [DSSCore]. These elements can wrap different types of time-stamp tokens; this profile
162 does not specify or constrain the internal structure of the `<dss:Timestamp>`, unless the
163 `<dss:SignatureType>` optional input is used (see section 3.1.1).

164 2.5 Transport Binding

165 This profile is transported using the HTTP POST Transport Binding defined in [DSSCore].

166 2.6 Security Binding

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

168

169

170 3 Profile of Signing Protocol

171 3.1 Element <SignRequest>

172 3.1.1 Element <OptionalInputs>

173 The <dss:SignatureType> optional input from **[DSSCore]** is supported and may be sent by
174 the client. The timestamping specific optional input <RenewTimestamp> may also be supported
175 and may be sent by the client. No other optional inputs are supported.

176 3.1.1.1 Element <SignatureType>

177 The <dss:SignatureType> optional input may be one of these values, from section 7. of
178 **[DSSCore]**:

179 urn:oasis:names:tc:dss:1.0:core:schema:XMLTimeStampToken

180 urn:ietf:rfc:3161

181 Servers may support other values. However, servers are under no obligation to support *any*
182 particular values. Thus, clients using the <dss:SignatureType> optional input may not
183 interoperate with certain servers.

184 3.1.1.2 Element <RenewTimestamp>

185 The <RenewTimestamp> optional input element indicates to the server that the current sign
186 request is a request for the renewal of an existing timestamp on data that were timestamped in
187 the past, so that the validity period of the existing timestamp is effectively extended.

188

```
189 <xs:element name="RenewTimestamp">  
190   <xs:complexType>  
191     <xs:sequence>  
192       <xs:element ref="PreviousTimestamp"/>  
193     </xs:sequence>  
194   </xs:complexType>  
195 </xs:element>  
196 <xs:element name="PreviousTimestamp">  
197   <xs:complexType>  
198     <xs:sequence>  
199       <xs:element ref="dss:Timestamp"/>  
200     </xs:sequence>  
201   </xs:complexType>  
202 </xs:element>
```

203

204 If the <RenewTimestamp> optional input is present in the sign request submitted by the client to
205 the server, and it is supported by the server, the <PreviousTimestamp> element contained in
206 this optional input must also be present as an element of the resulting timestamp generated by
207 the server and returned to the client. For XML timestamps of type <ds:signature>, processing
208 rules are described in Section 3.2.3.

209 Before submitting the sign request, the client must verify that the <PreviousTimestamp>
210 element corresponds to the document(s) being re-timestamped, and the client should verify the
211 <PreviousTimestamp> element.

212 Note: Legitimate reasons to renew a timestamp include (a) the public key certificate used to verify
213 the digital signature in the timestamp is nearing its expiration date, or (b) the client needs to
214 replace the hash value used for the timestamped data in the existing timestamp with a hash value
215 using a stronger hash algorithm.

216 **3.1.2 Element <InputDocuments>**

217 The client MAY send any component of <dss:InputDocument> element as input document. The
218 extraction and processing of these elements MUST be carried out as indicated in the core
219 document, with the changes mentioned in the present document.

220 If the client is not sending the <dss:SignatureType> optional input, then the client SHOULD only
221 send a single input document, since some types of time-stamps (e.g. RFC 3161) can only cover
222 one document per time-stamp.

223 If the client is sending the <dss:SignatureType> optional input, then the client MAY send multiple
224 input documents, if the client knows that the specified time-stamp type can handle them.

225 **3.2 Element <SignResponse>**

226 **3.2.1 Element <Result>**

227 This profile defines no additional <ResultMinor> codes.

228 **3.2.2 Element <OptionalOutputs>**

229 The server MUST NOT return any optional outputs.

230 **3.2.3 Element <SignatureObject>**

231 The server MUST return a <dss:Timestamp> signature object.

232 If the <RenewTimestamp> optional input is present in the sign request submitted by the client to
233 the server, and it is supported by the server, the <PreviousTimestamp> element contained in
234 this optional input must also be present as an element of the resulting timestamp generated by
235 the server and returned to the client. Specifically, for XML processing rules for XML timestamps
236 of type <ds:signature>, the server must include the <PreviousTimestamp> element
237 contained in the optional input as a child of an additional <ds:Signature>/<ds:Object> in
238 the newly generated timestamp (i.e. in addition to the <ds:object> containing the
239 <TstInfo>). An additional <ds:SignedInfo>/<ds:Reference> referencing the
240 <ds:Object>/<dss:PreviousTimestamp> must be included in the signature of the new
241 timestamp signature.

242 The server generating the new timestamp in response to a request carrying the
243 <RenewTimestamp> optional input need make no assertions about the validity of the
244 <PreviousTimestamp> element submitted to it within this optional input.

245 A server that does not support the <RenewTimestamp> optional input must reject the sign
246 request with a <ResultMajor> code of RequesterError and a <ResultMinor> code
247 urn:oasis:names:tc:dss:1.0:resultminor:NotSupported.

248 4 Profile of Verifying Protocol

249 4.1 Element <VerifyRequest>

250 4.1.1 Element <OptionalInputs>

251 The client may submit the <UseVerificationTime> optional input to instruct the server to
252 determine the timestamp's validity at the specified time, instead of the current time. No other
253 optional inputs are supported.

254 4.1.2 Element <SignatureObject>

255 The client MUST send a <dss:Timestamp> signature object.

256 Note: A timestamp T_2 that was generated by a server in response to a renewal request for
257 timestamp T_1 , that is, in response to a sign request on the same data as for timestamp T_1 and
258 carrying timestamp T_1 within the <PreviousTimestamp> element of the <RenewTimestamp>
259 optional input, may be used to assert current time validity for timestamp T_1 . This situation applies
260 when timestamp T_1 's current time validity can no longer be asserted independently, for example,
261 because the cryptographic primitives in timestamp T_1 are considered compromised. Specifically,
262 the client may:

- 263 • submit a verify request for timestamp T_2 ,
- 264 • submit a verify request for timestamp T_1 and include the optional input
265 <UseVerificationTime> with a value set to the issue time of timestamp T_2 (i.e. using element
266 <SpecificTime>).

267 If the result codes in the server verify responses indicate that both timestamps are valid as
268 requested, the client may assert that timestamp T_1 is currently valid, as supported by the fact that
269 timestamp T_1 is considered valid at the issue time of timestamp T_2 , and timestamp T_2 is
270 considered valid currently. This process may be generalized to timestamps that were generated
271 after multiple renewal requests on the same data, that is, timestamp T_1 , renewed by timestamp
272 T_2 , renewed by timestamp T_3 , and so on.

273 4.1.3 Element <InputDocuments>

274 The client MAY send any component of <dss:InputDocuments> element as input documents. The
275 extraction and processing of these elements MUST be carried out as indicated in the core
276 document, with the changes mentioned in the present document.

277 4.2 Element <VerifyResponse>

278 4.2.1 Element <Result>

279 This profile defines no additional <dss:ResultMinor> codes.

280 4.2.2 Element <OptionalOutputs>

281 The server MUST return the <dss:SigningTimeInfo> optional output.

282 4.2.2.1 Element <SigningTimeInfo>

283 The server MUST return this optional output profiled as detailed below:

- 284 1. Its `<dss:SigningTime>` child will contain the time indicated in the timestamp token (the
285 value in `<dss:CreationTime>` element of DSS XML timestamps or the `genTime` field
286 in RFC 3161 timestamp tokens).
- 287 2. If the timestamp token verified includes an indication of the deviation around the time
288 present in the timestamp token (like the `accuracy` field in RFC 3161 timestamps or the
289 `<dss:ErrorBound>` element in DSS XML timestamps), its
290 `<dss:SigningTimeBoundaries>` child **MUST** be present and it **MUST** contain the
291 lower and the upper boundaries suitably computed within its children.
- 292 The server **MUST NOT** return any other optional outputs.

293 **A. Acknowledgements**

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

296 **Participants:**

- 297 Dimitri Andivahis, Surety
- 298 Frederick Hirsch, Nokia
- 299 Pieter Kasselmann, Betrusted
- 300 Andreas Kuehne, individual
- 301 Paul Madsen, Entrust
- 302 John Messing, American Bar Association
- 303 Tim Moses, Entrust
- 304 Nick Pope, Thales eSecurity
- 305 Rich Salz, DataPower
- 306 Ed Shallow, Universal Postal Union
- 307