



Profile for Comprehensive Multi-signature Verification Reports for OASIS Digital Signature Services Version 1.0

Committee Draft 01

19 July 2009

Specification URIs:

This Version:

<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cd01.html>
<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cd01.doc>
<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cd01.pdf> (Authoritative)

Previous Version:

n/a

Latest Version:

<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr.html>
<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr.doc>
<http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr.pdf>

Technical Committee:

OASIS Digital Signature Services eXtended (DSS-X) TC

Chair(s):

Juan Carlos Cruellas, *UPC-DAC* <cruellas@ac.upc.edu>
Stefan Drees, Individual Member, <stefan@drees.name>.

Editor(s):

Detlef Hühnlein, Federal Ministry of the Interior, Germany <detlef.huehnlein@secunet.com>

Related work:

This specification is based on

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

and may be combined with other existing profiles, such as

- [oasis-dss-profiles-AdES-v1.0-os](#)
- [oasis-dss-profiles-german_signature_law-spec-v1.0-os](#)

for example.

Declared XML Namespace(s):

[urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:schema#](#)

Abstract:

This document defines a protocol and processing profile of the DSS Verifying Protocol specified in Section 4 of **[DSSCore]**, which allows to return individual signature verification reports for each signature in a verification request and include detailed information of the different steps taken during verification.

Status:

This document was last revised or approved by the Digital Signature Services TC on the above date. The level of approval is also listed above. Check the “Latest Version” or “Latest Approved Version” location noted above for possible later revisions of this document.

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-x/>.

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-x/ipr.php>)

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

Notices

Copyright © OASIS® 2009. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

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

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 OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

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' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on 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 implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

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

Table of Contents

1	Introduction.....	5
1.1	Terminology	5
1.2	Normative References	5
1.3	Namespaces.....	6
2	Profile Features.....	8
2.1	Overview.....	8
2.2	Scope	8
2.3	Relationship To Other Profiles	8
2.4	Profile Identifier.....	8
3	Verification Reports within DSS Verifying Protocol.....	9
3.1	Element <ReturnVerificationReport>	9
3.2	Element <VerificationReport>	10
3.3	Element <IndividualReport>.....	11
3.4	VerificationResultType	13
3.5	Element <DetailedSignatureReport>	13
3.5.1	SignatureValidityType.....	14
3.5.2	AlgorithmValidityType.....	15
3.5.3	CertificatePathValidityType	15
3.5.4	PropertiesType	27
3.5.5	Element <IndividualTimeStampReport>	38
3.5.6	Element <IndividualCertificateReport>	38
3.5.7	Element <IndividualAttributeCertificateReport>	38
3.5.8	Element <IndividualCRLReport>	38
3.5.9	Element <IndividualOCSPReport>	38
3.5.10	Element <EvidenceRecordReport>	38
4	Conformance	42
4.1	Level 1 – “Basic”	42
4.2	Level 2 – “Comprehensive”	42
4.3	Level 3 – “Convenient”	43
A.	Acknowledgements	44
B.	Revision History	45

1 Introduction

2 This document defines a protocol and processing profile of the DSS Verifying Protocol specified in
3 Section 4 of [DSSCore], which allows to support the verification of multiple signatures within some
4 <VerifyRequest> and include detailed information of the different steps taken during verification.
5 The following sections describe how to understand the rest of this document.

6 1.1 Terminology

7 The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD
8 NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described
9 in [RFC2119].

10 These keywords are capitalized when used to unambiguously specify requirements over protocol features
11 and behavior that affect the interoperability and security of implementations. When these words are not
12 capitalized, they are meant in their natural-language sense.

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

15 1.2 Normative References

- | | |
|----------------------|--|
| 16 [CAdES] | ETSI, <i>Electronic Signature Formats</i> , Electronic Signatures and Infrastructures
17 (ESI) – Technical Specification, ETSI TS 101 733 V1.7.4, 2008-07,
18 http://www.etsi.org |
| 19 [Core-XSD] | OASIS Standard: <i>DSS Schema</i> , February 2007, http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-schema-v1.0-os.xsd |
| 21 [DSSCore] | OASIS Standard: <i>Digital Signature Service Core Protocols and Elements</i> . OASIS
22 Standard, February 2007, http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.pdf |
| 24 [DSSAdES] | OASIS Standard: <i>Advanced Electronic Signature Profiles of the OASIS Digital
25 Signature Service Version 1.0</i> , OASIS Standard, April 2007, http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-AdES-spec-v1.0-os.pdf |
| 27 [DSSSigG] | OASIS Standard: <i>German Signature Law Profile of the OASIS Digital Signature
28 Service Version 1.0</i> , OASIS Standard, April 2007, http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles_german_signature_law-spec-v1.0-os.pdf |
| 30 [DSSVR-XSD] | OASIS Committee Draft: <i>DSS Verification Report Schema</i> , 19 th July 2009,
31 http://www.oasis-open.org/committees/download.php/33059/VerificationReport-CD1.xsd |
| 33 [DSSVisSig] | OASIS Committee Draft: <i>Visual Signature Profile of the OASIS Digital Signature
34 Services</i> , Committee Draft 01, April 2009, http://docs.oasis-open.org/dss-x/profiles/visualsig/v1.0/cd01/oasis-dssx-1.0-profiles-visualsig-cd1.pdf |
| 36 [EC/1999/93] | <i>Directive 1999/93/EC of the European Parliament and of the Council of 13
37 December 1999 on a Community framework for electronic signatures,</i>
38 http://europa.eu.int/eurlex/pri/en/oj/dat/2000/l_013/l_01320000119en00120020.pdf |
| 39 [ETSI102231] | ETSI: <i>ETSI TS 102231 Electronic Signatures and Infrastructure (ESI): Provision
40 of harmonized Trust-service status information</i> . Version 2.1.1 of March 2006, via
41 http://www.etsi.org |
| 42 [RFC2119] | S. Bradner: Key words for use in RFCs to Indicate Requirement Levels. IETF
43 RFC 2119, http://www.ietf.org/rfc/rfc2119.txt |

44	[RFC2560]	M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams: <i>X.509 Internet Public Key Infrastructure - Online Certificate Status Protocol – OCSP</i> , IETF RFC 2560, http://www.ietf.org/rfc/rfc3161.txt
45		
46		
47	[RFC3161]	C. Adams, P. Cain, D. Pinkas, R. Zuccherato: <i>Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)</i> . IETF RFC 3161, http://www.ietf.org/rfc/rfc3161.txt
48		
49		
50	[RFC3275]	D. Eastlake, J. Reagle, D. Solo: <i>(Extensible Markup Language) XML Signature Syntax and Processing</i> , IETF RFC 3275, http://www.ietf.org/rfc/rfc3275.txt
51		
52	[RFC3281]	S. Farrell, R. Housley: <i>An Internet Attribute Certificate Profile for Authorization</i> , IETF RFC 3281, via http://www.ietf.org/rfc/rfc3281.txt
53		
54	[RFC3852]	R. Housley: <i>Cryptographic Message Syntax (CMS)</i> . IETF RFC 3852, http://www.ietf.org/rfc/rfc3852.txt
55		
56	[RFC4514]	K. Zeilenga, Ed.: <i>Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names</i> , IETF RFC 4514, http://www.ietf.org/rfc/rfc4514.txt
57		
58		
59	[RFC4998]	T. Gondrom, R. Brandner, U. Pordesch: <i>Evidence Record Syntax (ERS)</i> , IETF RFC 4998, via http://www.ietf.org/rfc/rfc4998.txt
60		
61	[RFC5280]	D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk: <i>Internet X.509 Public Key Infrastructure, Certificate and Certificate Revocation List (CRL) Profile</i> , IETF RFC 5280, http://www.ietf.org/rfc/rfc5280.txt
62		
63		
64	[SAMLCore1.1]	E. Maler et al.: <i>Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V 1.1</i> . OASIS, November 2002. http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf
65		
66		
67	[SAMLCore2.0]	S. Cantor et al.: <i>Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0 OASIS Standard, 15 March 2005</i> . http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf
68		
69		
70	[XAdES]	ETSI: <i>XML Advanced Electronic Signatures (XAdES)</i> , ETSI TS 101 903, Version 1.3.2, March 2006, http://www.etsi.org
71		
72	[XML-ns]	T. Bray, D. Hollander, A. Layman: <i>Namespaces in XML</i> , W3C Recommendation, January 1999, http://www.w3.org/TR/1999/REC-xml-names-19990114
73		
74	[XMLSig]	D. Eastlake et al. <i>XML-Signature Syntax and Processing</i> , W3C Recommendation, June 2008, http://www.w3.org/TR/xmlsig-core/
75		

76 1.3 Namespaces

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

80 This schema is associated with the following XML namespace:

81 `urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:schema#`

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

83

84 Conventional XML namespace prefixes are used in this document:

- 85 • The prefix `vr:` (or no prefix) stands for this profiles namespace **[DSSVR-XSD]**.
- 86 • The prefix `ds:` stands for the W3C XML Signature namespace **[XMLSig]**.
- 87 • The prefix `dss:` stands for the DSS core namespace **[Core-XSD]**.
- 88 • The prefix `saml:` stands for the OASIS SAML Schema namespace **[SAMLCore1.1]**.
- 89 • The prefix `tsl:` stands for the ETSI Trust-service status information namespace **[ETSI102231]**.

- 90 • The prefix `xades` : stands for ETSI XML Advanced Electronic Signatures (XAdES) document
91 **[XAdES]**.
- 92
- 93 Applications MAY use different namespace prefixes, and MAY use whatever namespace
94 defaulting/scoping conventions they desire, as long as they are compliant with the Namespaces in XML
95 specification **[XML-ns]**.
- 96

97 **2 Profile Features**

98 **2.1 Overview**

99 While the DSS Verifying Protocol specified in Section 4 of [DSSCore] allows to verify digital signatures
100 and time stamps, this protocol is fairly limited with respect to the verification of multiple signatures in a
101 single request (cf. Section 4.3.1 of [DSSCore]).

102 In a similar manner it is possible to request and provide processing details (cf. Section 4.5.5 of
103 [DSSCore]), but this simple mechanism does not support the verification of multiple signatures in a single
104 request and there are no defined structures yet, which reflect the necessary steps in the verification of a
105 complex signature, like an advanced electronic signature according to the European Directive
106 [EC/1999/93] for example.

107 Therefore the present profile defines how

- 108 • individual verification results may be returned, if multiple signatures are part of a
109 <dss:VerifyRequest> and
- 110 • detailed information gathered in the various steps taken during verification may be included in the
111 response to form a comprehensive verification report.

112 The requester MAY request the activation of this profile by sending a <ReturnVerificationReport>
113 element (cf. Section 3.1) in <dss:OptionalInputs>. A responder, which conforms to the present
114 profile SHALL return a <VerificationReport> element (cf. Section 3.2) in
115 <dss:OptionalOutputs>.

116 **2.2 Scope**

117 This document profiles the DSS Verifying Protocol (cf. [DSSCore], Section 4).

118 It does *not* profile the DSS Signing Protocol (cf. [DSSCore], Section 3) and does *neither specify nor*
119 constrain

- 120 • the type of signature object,
- 121 • the transport binding or
- 122 • the security binding.

123 **2.3 Relationship To Other Profiles**

124 This profile is based directly on the [DSSCore]. This profile is intended to be combined with other profiles
125 freely.

126 **2.4 Profile Identifier**

127 The DSS-client MAY use the following identifier in the `Protocol` attribute of a `VerifyRequest`:

128 `urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport`

129 The DSS-server MAY use this identifier in the `VerifyResponse`.

130 3 Verification Reports within DSS Verifying Protocol

131

132 3.1 Element <ReturnVerificationReport>

133 The <ReturnVerificationReport>-element is an optional input for the DSS Verifying Protocol to
134 request an individual report for each signature. It is defined as follows:

135

```
136     <element name="ReturnVerificationReport">
137         <complexType>
138             <sequence>
139                 <element name="IncludeVerifier" type="boolean"
140                     maxOccurs="1"
141                         minOccurs="0" default="true" />
142                 <element name="IncludeCertificateValues" type="boolean"
143                     maxOccurs="1"
144                         minOccurs="0" default="false" />
145                 <element name="IncludeRevocationValues" type="boolean"
146                     maxOccurs="1"
147                         minOccurs="0" default="false" />
148                 <element name="ExpandBinaryValues" type="boolean"
149                     maxOccurs="1"
150                         minOccurs="0" default="false"/>
151                 <element name="ReportDetailLevel" type="anyURI"
152                     maxOccurs="1"
153                         minOccurs="0"
154                     default="urn:oasis:names:tc:dss:1.0:profiles:
155
156                         verificationreport:reportdetail:allDetails" />
157             </sequence>
158         </complexType>
159     </element>
```

160

161 It contains the following elements:

162 <IncludeVerifier> [Default]

163 This option specifies, whether the identity of the verifier should be included into the report or not. This
164 is especially useful when (possibly time stamped) reports are archived. It defaults to 'true'.

165 <IncludeCertificateValues> [Default]

166 With this option it is possible to include the certificate values, which are used to verify the signature (in
167 binary form or as equivalent XML structure) into the report. This option defaults to 'false'.

168 <IncludeRevocationValues> [Default]

169 This option specifies, whether the used revocation values (OCSP responses, CRLs and TSLs) should
170 be included (in binary form or as equivalent XML structure) into the report or not. It defaults to 'false'.

171 <ExpandBinaryValues> [Default]

172 If this element is set to true a server which fulfills the conformance level "Convenient" MUST include
173 the content of certificates and revocation information not only as ASN.1-coded binary values into the
174 verification report, but also as equivalent XML structures. This option defaults to 'false'.

175 <ReportDetailLevel> [Optional]

176 This option specifies the detail level of the verification report. The following options are defined:

- 177 – [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:reportdetail:noDetails](#)
 178 For every signature only the final result of the verification is reported.
- 179 – [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:reportdetail:noPathDetails](#)
 180 Additionally to the final result also the details of the signature verification including the result of
 181 the certificate path validation are reported. The details concerning the validation of individual
 182 certificates in the path are omitted however.
- 183 – [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:reportdetail:allDetails](#)
 184 For every signature, the certificate path details and details on the validation of individual
 185 certificates in the path are requested. For every signature, the certificate path and each individual
 186 certificate the details are reported. If the `<ReportDetailLevel>`-element is missing, this
 187 option is assumed as default.

188 **3.2 Element <VerificationReport>**

189 If the element `<ReturnVerificationReport>` is provided as optional input in the request, the server
 190 MUST include in the response the element `<VerificationReport>` as optional output:

191

```
192   <element name="VerificationReport" type="vr:VerificationReportType" />
```

193

194 The **VerificationReportType** is the base structure for verification reports defined by this profile. It is
 195 defined as follows:

196

```
197   <complexType name="VerificationReportType">
  198     <sequence>
  199       <element ref="dss:VerificationTimeInfo" maxOccurs="1"
  200         minOccurs="0" />
  201       <element name="VerifierIdentity" type="vr:IdentifierType"
  202         maxOccurs="1" minOccurs="0" />
  203       <element name="IndividualReport" maxOccurs="unbounded"
  204         type="vr:IndividualReportType" minOccurs="0" />
  205     </sequence>
  206   </complexType>
```

207

208 It contains the following elements:

209 `<VerificationTimeInfo>` [Optional]

210 This element MAY contain the verification time, which was used by the server and other relevant time
 211 instants.

212 `<VerifierIdentity>` [Optional]

213 This element contains the identity of the verifier, if the report option `<IncludeVerifier>` was set to
 214 ‘true’. It is of type **vr:IdentifierType**, which is defined below.

215 `<IndividualReport>` [Optional, Unbounded]

216 For each *independent* signed object (signature, time stamp, certificate, CRL, OCSP-response,
 217 evidence record etc.) that has been used in the signature verification process there will be one
 218 `<IndividualReport>`-element in the verification report. The details of this element are specified in
 219 the following section.

220 The **IdentifierType** MAY contain different types of identifiers. It is defined as follows:

221

```
222   <complexType name="IdentifierType">
  223     <sequence>
  224       <element ref="ds:X509Data" maxOccurs="1" minOccurs="0">
```

```

225      </element>
226      <element name="SAMLv1Identifier" type="saml:NameIdentifierType"
227          maxOccurs="1" minOccurs="0" />
228      <element name="SAMLv2Identifier" type="saml2:NameIDType"
229          maxOccurs="1" minOccurs="0" />
230      <element name="Other" type="dss:AnyType" maxOccurs="1"
231          minOccurs="0" />
232    </sequence>
233  </complexType>

```

234

235 It MAY contain the following elements or other identifying information:

236 <ds:X509Data> [Optional]

237 This element contains, if present, an X.509-certificate or certificate related information. Please refer to
238 [[RFC3275](#)] for further details with respect to the ds:X509Data-element.

239 <SAMLv1Identifier> [Optional]

240 This element contains, if present, an identifier of type **saml:NameIdentifierType** as defined in
241 [[SAMLCore1.1](#)].

242 <SAMLv2Identifier> [Optional]

243 This element contains, if present, an identifier of type **saml2:NameIDType** as defined in
244 [[SAMLCore2.0](#)].

245 <Other> [Optional]

246 This element MAY contain, if present, other identifying information.

247

248 3.3 Element <IndividualReport>

249

250 The element <IndividualReport> is part of the <VerificationReport>-element (see Section 3.2)
251 and is of type **IndividualReportType**, which is defined as follows:

252

```

253  <complexType name="IndividualReportType">
254    <sequence>
255      <element name="SignedObjectIdentifier"
256          type="vr:SignedObjectIdentifierType"/>
257      <element ref="dss:Result"/>
258      <element name="Details" type="dss:AnyType" maxOccurs="1"
259          minOccurs="0" />
260    </sequence>
261  </complexType>

```

262

263 It contains the following elements:

264 <SignedObjectIdentifier> [Required]

265 This element identifies the signature or validation data under consideration. The details of the
266 SignedObjectIdentifierType are specified below.

267 <Result> [Required]

268 The result of the signature verification as defined in section 2.6 of [[DSSCore](#)].

269 <Details> [Optional]

270 The <Details> element MAY contain a detailed report for the signature or validation data under
271 consideration or any other signature-specific optional output defined in Section 4.5 of [[DSSCore](#)].

272 The corresponding elements, which are specified in this document for this purpose are listed in
273 Section 4.2.

274

275 The **SignedObjectIdentifierType** is defined as follows:

276

```
277 <complexType name="SignedObjectIdentifierType">
278   <sequence>
279     <element name="DigestAlgAndValue"
280       type="XAdES:DigestAlgAndValueType" maxOccurs="1"
281       minOccurs="0"/>
282     <element ref="ds:CanonicalizationMethod" maxOccurs="1" minOccurs="0"
283     />
284     <element name="SignedProperties"
285       type="vr:SignedPropertiesType" maxOccurs="1" minOccurs="0" />
286     <element ref="ds:SignatureValue" maxOccurs="1" minOccurs="0" />
287     <element name="Other" type="dss:AnyType" maxOccurs="1" minOccurs="0"
288     />
289   </sequence>
290   <attribute name="WhichDocument" type="IDREF" use="optional"/>
291   <attribute name="XPath" type="string" use="optional"/>
292   <attribute name="Offset" type="integer" use="optional"/>
293   <attribute name="FieldName" type="string" use="optional"/>
294 </complexType>
```

295

296 The set of child elements of the **SignedObjectIdentifierType** SHOULD be chosen to identify the
297 signature or validation data in a given context in an unambiguous manner.

298 It contains the following attributes and elements:

299 <DigestAlgAndValue> [Optional]

300 This element contains, if present, the hash value of the signature or validation data under
301 consideration, where the signed object itself (e.g. the <ds:Signature>-element in case of an XML-
302 signature according to [RFC3275], the SignedData-structure in case of a CMS-signature according
303 to [RFC3852] or a time stamp according to [RFC3161], the Certificate- or CertificateList-
304 structure in case of an X.509-certificate or CRL according to [RFC5280] or the OCSPResponse-
305 structure in case of an OCSP-response according to [RFC2560] for example) serves as input for the
306 hash-calculation. The structure of the DigestAlgAndValueType is defined in [XAdES]. This
307 element SHOULD NOT be used if the unique identification can be guaranteed by other elements.

308 <ds:CanonicalizationMethod> [Optional]

309 This element indicates, if present, the canonicalization method to be used before hashing XML-
310 formatted data. Please refer to [RFC3275] for details of this element. This element is only necessary if
311 XML-based structures are subject to hashing.

312 <SignedProperties> [Optional]

313 This element contains, if present, any number of signed properties, which may be useful to identify the
314 signature under consideration. This MAY comprise information about the signatory and the signing
315 time for example. The structure of the SignedPropertiesType is defined in Section 3.5.4.2. In case
316 of signatures according to [RFC3275] or [RFC3852] this element SHOULD be present.

317 <ds:SignatureValue> [Optional]

318 This element specifies, if present, the binary signature value of the signature under consideration. This
319 element SHOULD be present – particularly if the used signature algorithm is randomized and hence
320 this element may serve as unique identifier.

321 <Other> [Optional]

322 This element MAY contain other elements, which (help to) identify a signature or related validation
323 data in a unique manner.

324 WhichDocument [Optional]
 325 This attribute MAY specify the document which contains the signature under consideration. Note that
 326 this identifier is only unique with respect to a specific request message (see [**DSSCore**], Section
 327 2.4.1).
 328 XPath [Optional]
 329 This attribute MAY be used to point to a specific signature within an XML document.
 330 Offset [Optional]
 331 This attribute specifies the first byte of some signature and MAY be used to point to a specific
 332 signature within some binary document.
 333 FieldName [Optional]
 334 This attribute specifies the name of a signature field and MAY be used to point to a specific signature
 335 within some document format, in which there are field names such as PDF for example.

3.4 VerificationResultType

336 The **VerificationResultType** defined below is extensively used in the present profile to indicate the
 337 success or failure of individual verification steps.
 338 This type draws from the **dss:Result**-element and the **dss:DetailType** defined in [**DSSCore**] and is
 339 defined as follows:

```
341 <complexType name="VerificationResultType">
342   <sequence>
343     <element name="ResultMajor" type="anyURI"/>
344     <element name="ResultMinor" type="anyURI" minOccurs="0"/>
345     <element name="ResultMessage" type="dss:InternationalStringType"
346 minOccurs="0"/>
347     <any namespace="#other" processContents="lax" minOccurs="0"
348 maxOccurs="unbounded"/>
349   </sequence>
350 </complexType>
```

351
 352 <ResultMajor> [Required]
 353 This element MUST indicate whether the verification result is valid, invalid or indetermined using the
 354 URIs defined in [**DSSCore**]:
 355 • urn:oasis:names:tc:dss:1.0:detail:valid
 356 • urn:oasis:names:tc:dss:1.0:detail:invalid
 357 • urn:oasis:names:tc:dss:1.0:detail:indetermined
 358 <ResultMinor> [Optional]
 359 In case of an invalid or indetermined verification step, further details MAY be provided using a specific
 360 URI defined in this document or other profiles.
 361 <ResultMessage> [Optional]
 362 Especially in case of an invalid or indetermined verification step, further details MAY be provided in
 363 textual form.
 364 Furthermore an element of type **VerificationResultType** MAY contain other elements.

3.5 Element <DetailedSignatureReport>

365 The <**DetailedSignatureReport**>-element MAY appear in the <**Details**>-element within the
 366 <**IndividualReport**>-element, which is specified in Section 3.3 above. This element is defined as
 367 follows:
 368

```

369     <element name="DetailedSignatureReport"
370      type="vr:DetailedSignatureReportType" />
371
372 The DetailedSignatureReportType in turn is specified as follows:
373
374 <complexType name="DetailedSignatureReportType">
375   <sequence>
376     <element name="FormatOK" type="vr:VerificationResultType" />
377     <element name="Properties" type="vr:PropertiesType"
378       maxOccurs="1" minOccurs="0" />
379     <element ref="dss:VerifyManifestResults" maxOccurs="1"
380       minOccurs="0" />
381     <element name="SignatureHasVisibleContent" type="boolean"
382       maxOccurs="1" minOccurs="0"/>
383     <element name="SignatureOK"
384       type="vr:SignatureValidityType" />
385     <element name="CertificatePathValidity"
386       type="vr:CertificatePathValidityType" />
387   </sequence>
388 </complexType>
389
390 It contains the following elements:
391 <FormatOK> [Required]
392   This element indicates, whether the format of the signature is ok or not. More information on the use of
393   the VerificationResultType may be found in Section 3.4.
394 <Properties> [Optional]
395   This element contains information gathered during the verification of signed or unsigned properties.
396   The structure of the PropertiesType is defined in Section 3.5.4.
397 <VerifyManifestResults> [Optional]
398   This element is present, if a manifest verification has been performed. The structure and the
399   semantics of this element is described in Section 4.5.1 of [DSSCore].
400 <SignatureHasVisibleContent> [Optional]
401   This element is only present if the FieldName-attribute (cf. Section 3.3) is present and indicates
402   whether the signature under consideration has visual signature content as explained in [DSSVisSig].
403 <SignatureOK> [Required]
404   This element contains information about the mathematical validity of the digital signature under
405   consideration. It is of type SignatureValidityType, which is specified in Section 3.5.1.
406 <CertificatePathValidity> [Required]
407   This element contains the results of the certificate path validation. The CertificatePathValidityType is
408   defined in section 3.5.3.
409 

### 3.5.1 SignatureValidityType


410 The SignatureValidityType is used in the definition of the <DetailedSignatureReport>-element
411 above for example and it is specified as follows:
412
413 <complexType name="SignatureValidityType">
414   <sequence>
415     <element name="SigMathOK" type="vr:VerificationResultType" />

```

```
416             <element name="SignatureAlgorithm"
417 type="vr:AlgorithmValidityType" maxOccurs="1" minOccurs="0"/>
418         </sequence>
419     </complexType>
```

421

422 It comprises the following elements:

423 <SigMathOK> [Required]

424 Contains information about the mathematical validity of the digital signature under consideration. More
425 information on the use of the **VerificationResultType** may be found in Section 3.4.

426 <SignatureAlgorithm> [Optional]

427 This element MAY contain information about the applied signature algorithm. It is of type
428 **AlgorithmValidityType**, which is defined below.

429

430 3.5.2 AlgorithmValidityType

431 The **AlgorithmValidityType** is used in the definition of the **SignatureValidityType** above for example
432 and is specified as follows:

433

```
434 <complexType name="AlgorithmValidityType">
435     <sequence>
436         <element name="Algorithm" type="anyURI" />
437         <element name="Parameters" type="dss:AnyType" maxOccurs="1"
438 minOccurs="0" />
439         <element name="Suitability" type="vr:VerificationResultType"
440 maxOccurs="1" minOccurs="0"/>
441     </sequence>
442 </complexType>
```

443

444 <Algorithm> [Required]

445 This element contains the URI for the algorithm.

446 <Parameters> [Optional]

447 This element MAY contain further parameters for the cryptographic algorithm.

448 <Suitability> [Optional]

449 This element MAY contain the information about the suitability of the algorithm under consideration.
450 Note that it MAY depend on the policy of the specific signature and/or the policy under which the DSS
451 server is operated, whether the suitability of the algorithms is verified and what kind of algorithms are
452 considered appropriate under given circumstances and which are not. More information on the use of
453 the **VerificationResultType** may be found in Section 3.4.

454 3.5.3 CertificatePathValidityType

455 The <CertificatePathValidity>-element is of type **CertificatePathValidityType** and is used in the
456 definition of

- 457 • **DetailedSignatureReportType** (see above),
- 458 • **AttributeCertificateValidityType** (see Section 3.5.4.3),
- 459 • **CRLValidityType** (see Section 3.5.3.4),
- 460 • **OCSPValidityType** (see Section 3.5.3.5) and
- 461 • **TimeStampValidityType** (see Section 3.5.4.4).

462

463 It is specified as follows:

464

```
465 <complexType name="CertificatePathValidityType">
466     <sequence>
467         <element name="PathValiditySummary"
468             type="vr:VerificationResultType" />
469         <element name="CertificateIdentifier"
470             type="ds:X509IssuerSerialType" />
471         <element name="PathValidityDetail"
472             type="vr:CertificatePathValidityDetailType"
473             minOccurs="0" maxOccurs="1"/>
474     </sequence>
475 </complexType>
```

476

477 It contains the following elements:

478 <PathValiditySummary> [Required]

479 This element is of type **VerificationResultType** (see Section 3.4) and contains a summary of the
480 result of the certificate path validation.

481 <CertificateIdentifier> [Required]

482 This element is of type **ds:X509IssuerSerialType** (see Section 4.4.4 of [RFC3275]) and contains a
483 unique reference to the certificate whose path has been checked.

484 <PathValidityDetail> [Optional]

485 Contains detailed results of the certificate path validation, if the element <ReportDetailLevel> in
486 the report options (see Section 3.1) was set to **urn:oasis:names:tc:dss:1.0:
487 profiles:verificationreport:reportdetail:allDetails** and the detailed validity information has not been
488 included elsewhere in the verification report.

489

490 The structure of **CertificatePathValidityDetailType** is specified as follows:

491

```
492 <complexType name="CertificatePathValidityDetailType">
493     <sequence>
494         <sequence maxOccurs="unbounded" minOccurs="0">
495             <element name="CertificateValidity"
496                 type="vr:CertificateValidityType" />
497         </sequence>
498         <element name="TSLValidity"
499             type="vr:TrustStatusListValidityType" maxOccurs="1"
500             minOccurs="0" />
501             <element name="TrustAnchor" type="vr:VerificationResultType"
502             />
503         </sequence>
504     </complexType>
```

505

506 It contains the following elements:

507 <CertificateValidity> [Optional, Unbounded]

508 For every certificate in the certificate path there will be a <CertificateValidity>-element, which
509 provides information about the validity of the specific certificate. The structure of the
510 **CertificateValidityType** is defined below.

511 <TSLValidity> [Optional]

512 This element MAY contain information about a Trust-service Status List according to [ETSI102231]
513 and its validity. The **TrustStatusListValidityType** is defined in Section 3.5.3.6.

514 <TrustAnchor> [Required]

515 This element indicates how the trusted root certificate, which is used as trust anchor within the
516 verification process, is stored. The following URIs are defined for this purpose:

- 517 • [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:trustanchor:SSCD](#) – indicates that the
518 trusted root certificate is stored within a secure signature creation device according to
519 [EC/1999/93].
- 520 • [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:trustanchor:otherCard](#) – indicates that
521 the trusted root certificate is stored within some other hardware token.
- 522 • [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:trustanchor:certDataBase](#) – indicates
523 that the trusted root certificate is stored within some certificate data base.
- 524 • [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:trustanchor:other](#) – indicates that the
525 trusted root certificate is stored using other means.

526

527 3.5.3.1 CertificateValidityType

528

529 The **CertificateValidityType** contains information about the validity of a single certificate and is defined
530 as follows:

531

```
532 <complexType name="CertificateValidityType">
533   <sequence>
534     <element name="CertificateIdentifier" type="ds:X509IssuerSerialType"
535   />
536     <element name="Subject" type="string" />
537     <element name="ChainingOK" type="vr:VerificationResultType"
538       maxOccurs="1" minOccurs="0"/>
539     <element name="ValidityPeriodOK" type="vr:VerificationResultType" />
540     <element name="ExtensionsOK" type="vr:VerificationResultType" />
541     <element name="CertificateValue" type="base64Binary"
542       maxOccurs="1" minOccurs="0" />
543     <element name="CertificateContent"
544       type="vr:CertificateContentType" maxOccurs="1" minOccurs="0"
545   />
546     <element name="SignatureOK"
547       type="vr:SignatureValidityType" />
548     <element name="CertificateStatus" type="vr:CertificateStatusType" />
549   </sequence>
550 </complexType>
```

551

552 It contains the following elements:

553 <CertificateIdentifier> [Required]

554 This element is of type **ds:X509IssuerSerialType** (see [RFC3275], Section 4.4.4) and identifies the
555 certificate under consideration.

556 <Subject> [Required]

557 This element contains the subject of the certificate, where the string representation of distinguished
558 names defined in [RFC4514] MUST be used and hence an example of a <Subject>-element may be
559 CN=John Doe,O=Foo Inc.,OU=Sales etc.

560 <ChainingOK> [Optional]

561 If present, this element indicates whether the chaining to a previous certificate in the certificate path is
 562 ok or not. If the certificate under consideration is the first certificate in the certificate path, this element
 563 SHOULD be omitted. More information on the use of the **VerificationResultType** may be found in
 564 Section 3.4.
 565 <ValidityPeriodOK> [Required]
 566 This element indicates, whether the reference point in time is within the validity period of the
 567 certificate. More information on the use of the **VerificationResultType** may be found in Section 3.4.
 568 <ExtensionsOK> [Required]
 569 This element indicates, whether the certificate extensions are correct. More information on the use of
 570 the **VerificationResultType** may be found in Section 3.4.
 571 <CertificateValue> [Optional]
 572 If present, this element contains the certificate in binary form (coded in ASN.1), if the report option
 573 <IncludeCertificateValues> is set to 'true' and if the certificate is not already included in the
 574 verification report.
 575 <CertificateContent> [Optional]
 576 If present, this element contains detailed information about the content of the certificate, if the report
 577 option <ExpandBinaryValues> is set to 'true' and if the certificate content is not already included in
 578 the verification report.
 579 <SignatureOK> [Required]
 580 This element indicates, whether the digital signature of the certificate is mathematically correct or not.
 581 The **SignatureValidityType** is defined in section 3.5.1.
 582 <CertificateStatus> [Required]
 583 This element contains information about the result of the certificate revocation check. The
 584 **CertificateStatusType** is defined in Section 3.5.3.3.
 585

586 3.5.3.2 CertificateContentType

587
 588 The **CertificateContentType** is used in **CertificateValidityType** and derived from the
 589 TBSCertificate-structure defined in [RFC5280] specified as follows:
 590

```

591   <complexType name="CertificateContentType">
592     <sequence>
593       <element name="Version" type="integer" maxOccurs="1"
594       minOccurs="0" />
595       <element name="SerialNumber" type="integer" />
596       <element name="SignatureAlgorithm" type="anyURI" />
597       <element name="Issuer" type="string" />
598       <element name="ValidityPeriod" type="vr:ValidityPeriodType" />
599       <element name="Subject" type="string" />
600       <element name="Extensions" type="vr:ExtensionsType"
601       minOccurs="0" />
602     </sequence>
603   </complexType>
  
```

604
 605 It contains the following elements:
 606 <Version> [Optional]
 607 This element contains, if present, the version of the certificate structure.
 608 <SerialNumber> [Required]

609 This element MUST contain the serial number of the certificate.
 610 <SignatureAlgorithm> [Required]
 611 This element MUST contain an identifier of the used signature algorithm. The
 612 vr:VerificationResultType is defined in Section 3.4.
 613 <Issuer> [Required]
 614 This element MUST contain the issuer of the certificate, where different relative distinguished names
 615 in a sequence MAY be separated by ":".
 616 <ValidityPeriod> [Required]
 617 This element MUST contain the validity period of the certificate. The **ValidityPeriodType** is defined
 618 below.
 619 <Subject> [Required]
 620 This element contains the subject of the certificate, where the string representation of distinguished
 621 names defined in [RFC4514] MUST be used and hence an example of a <Subject>-element may be
 622 CN=John Doe,O=Foo Inc.,OU=Sales etc.
 623
 624 <Extensions> [Optional]
 625 If present, this element contains information about the list of extensions present in the certificate under
 626 consideration. The **ExtensionsType** is defined below.
 627
 628 The **ValidityPeriodType** is specified as follows:
 629
 630 <complexType name="ValidityPeriodType">
 631 <sequence>
 632 <element name="NotBefore" type="dateTime" />
 633 <element name="NotAfter" type="dateTime" />
 634 </sequence>
 635 </complexType>
 636
 637 It contains the following elements:
 638 <NotBefore> [Required]
 639 The certificate is not valid before this point in time.
 640 <NotAfter> [Required]
 641 The certificate is not valid after this point in time.
 642
 643 The **ExtensionsType** is specified as follows:
 644
 645 <complexType name="ExtensionsType">
 646 <sequence minOccurs="0" maxOccurs="unbounded">
 647 <element name="Extension" type="vr:ExtensionType" />
 648 </sequence>
 649 </complexType>
 650
 651 It contains an unbounded number <Extension>-elements of type **ExtensionType**. This type is defined
 652 as follows:
 653
 654 <complexType name="ExtensionType">

```

655      <sequence>
656          <element name="ExtnId" type="XAdES:ObjectIdentifierType" />
657          <element name="Critical" type="boolean" />
658          <element name="ExtnValue" type="dss:AnyType" maxOccurs="1"
659          minOccurs="0" />
660          <element name="ExtensionOK" type="vr:VerificationResultType"
661      />
662      </sequence>
663  </complexType>

664
665 It contains the following elements:
666 <ExtnId> [Required]
667 This element MUST contain the identifier of the extension as urn:oid: ... in the <Identifier>-element and MAY contain further information in the <Description>- and <DocumentationReferences>-elements. Please refer to [XAdES] for more information on the XAdES:ObjectIdentifierType.
668 <Critical> [Required]
669 This element specifies, whether the extension is critical or not.
670
671 <ExtnValue> [Optional]
672 This element SHOULD contain the value of the extension as an XML-structure, which mirrors the original ASN.1-definition of the extension.
673
674 <ExtensionOK> [Required]
675 This element contains information about the validity of the specific extension within the given context of the certificate.
676
677
678
679
680

```

681 3.5.3.3 CertificateStatusType

```

682
683 The CertificateStatusType is defined as follows:
684

```

```

685 <complexType name="CertificateStatusType">
686     <sequence>
687         <element name="CertStatusOK" type="vr:VerificationResultType"
688     />
689         <element name="RevocationInfo" maxOccurs="1"
690         minOccurs="0">
691             <complexType>
692                 <sequence>
693                     <element name="RevocationDate"
694                     type="dateTime" />
695                     <element name="RevocationReason"
696                     type="vr:VerificationResultType" />
697                 </sequence>
698             </complexType>
699         </element>
700         <element name="RevocationEvidence" maxOccurs="1"
701         minOccurs="0">
702             <complexType>
703                 <choice>
704                     <element name="CRLValidity"
705                     type="vr:CRLValidityType" />
706                     <element name="CRLReference"
707

```

```

707                               type="XAdES:CRLIdentifierType" />
708                         <element name="OCSPValidity"
709                               type="vr:OCSPValidityType" />
710                         <element name="OCSPReference"
711                               type="XAdES:OCSPIdentifierType" />
712                         <element name="Other" type="dss:AnyType"/>
713                     </choice>
714                 </complexType>
715             </element>
716         </sequence>
717     </complexType>

```

718

719 It contains the following elements:

720 <CertStatusOK> [Required]

721 This element MUST contain the status of the certificate.

722 <RevocationInfo> [Optional]

723 If the certificate is revoked this element will contain more information about the revocation. It is defined
724 to be a sequence, which contains the following elements:

- <RevocationDate>

725 contains the date and time of revocation.

- <RevocationReason>

726 contains the reason for revocation. Following the definition of CRLReason in [RFC5280] there are
727 the following URIs to specify the revocation reason:

- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:unspecified
- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:keyCompromise
- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:cACompromise
- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:affiliationChanged
- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:superseded
- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:cessationOfOperation
- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:certificateHold
- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:removeFromCRL
- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:privilegeWithdrawn
- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:aACompromise

741 <RevocationEvidence> [Optional, Choice]

742 This element contains, if present, the used source of revocation information. This can be one of the
743 following elements:

- <CRLValidity>

744 This element contains information about the used CRL and its validity. The **CRLValidityType** is
745 defined in Section 3.5.3.4.

- <CRLReference>

746 This element contains a reference to the CRL in case it is already included elsewhere in the
747 present verification report. The **XAdES:CRLIdentifierType** is defined in [XAdES].

- <OCSPValidity>

750 This element contains information about the used OCSP response and its validity. The
751 **OCSPValidityType** is defined in Section 3.5.3.5.

- 753 • <OCSPReference>
 754 This element contains a reference to the used OCSP response, if it is already included elsewhere
 755 in the present verification report. The **XAdES:OCSPIdentifierType** is defined in [XAdES].
 756 • <Other>
 757 This element MAY contain information about alternative sources of revocation information.

758 **3.5.3.4 CRLValidityType**

759 The **CRLValidityType** contains information about a CRL and its validity and is specified as follows:

760

```
761 <complexType name="CRLValidityType">
762   <sequence>
763     <element name="CRLIdentifier" type="XAdES:CRLIdentifierType"
764       maxOccurs="1" minOccurs="1" />
765     <element name="CRLValue" type="base64Binary"
766       maxOccurs="1" minOccurs="0" />
767     <element name="CRLContent" type="vr:CRLContentType"
768       maxOccurs="1" minOccurs="0" />
769     <element name="SignatureOK" type="vr:SignatureValidityType" />
770     <element name="CertificatePathValidity"
771       type="vr:CertificatePathValidityType" />
772   </sequence>
773   <attribute name="Id" type="ID" use="optional" />
774 </complexType>
```

775

776 It contains the following attributes and elements:

777 <Id> [Optional]

778 This attribute contains an optional identifier for the element.

779 <CRLIdentifier> [Required]

780 This element refers to an X.509v2 CRL according to [RFC5280].

781 <CRLValue> [Optional]

782 If present, this element contains the CRL (encoded in ASN.1) if the report option
 783 <IncludeRevocationValues> is set to 'true'.

784 <CRLContent> [Optional]

785 This element contains, if present, the CRL in form of an equivalent XML structure if the report option
 786 <ExpandBinaryValues> is set to 'true'. The **CRLContentType** is defined below.

787 <SignatureOK> [Required]

788 This element indicates, whether the digital signature of the CRL is mathematically correct or not. The
 789 **SignatureValidityType** is defined in section 3.5.1.

790 <CertificatePathValidity> [Required]

791 This element contains the result of the validation of the certificate path of the certificate which has
 792 been used to sign the CRL. The **CertificatePathValidityType** is defined at the beginning of Section
 793 3.5.3.

794

795 The **CRLContentType** is aligned to [RFC5280] specified as follows:

796

```
797 <complexType name="CRLContentType">
798   <sequence>
799     <element name="Version" minOccurs="0" type="integer" />
800     <element name="Signature" type="anyURI" />
```

```

801             <element name="Issuer" type="string" />
802             <element name="ThisUpdate" type="dateTime" />
803             <element name="NextUpdate" minOccurs="0" type="dateTime" />
804             <element name="RevokedCertificates" minOccurs="0">
805                 <complexType>
806                     <sequence minOccurs="0" maxOccurs="unbounded">
807                         <element name="UserCertificate"
808                             type="integer" />
809                         <element name="RevocationDate"
810                             type="dateTime" />
811                         <element name="CrlEntryExtensions"
812                             minOccurs="0"
813                             type="vr:ExtensionsType" />
814                     </sequence>
815                 </complexType>
816             </element>
817             <element name="CrlExtensions" type="vr:ExtensionsType"
818                 minOccurs="0" />
819         </sequence>
820     </complexType>

```

821

822 It contains the following elements:

823 <Version> [Optional]
824 This element contains, if present, the version of the CRL-structure.

825 <Signature> [Required]
826 This element contains the algorithm identifier for the algorithm used to sign the CRL.

827 <Issuer> [Required]
828 This element contains the issuer of the CRL, where different relative distinguished names in a sequence MAY be separated by ":".

829 <ThisUpdate> [Required]
830 This element contains the issue date of the CRL.

831 <NextUpdate> [Optional]
832 This element contains, if present, the date by which the next CRL will be issued.

833 <RevokedCertificates> [Optional]
834 The revoked certificates are contained in an unbounded sequence. They are listed by their serial numbers (element <UserCertificate>). Certificates revoked by the CA are uniquely identified by their certificate serial number. The date on which the revocation occurred is contained in the element <RevocationDate>. Additional information MAY be supplied in the element <CrlEntryExtensions>.

835 <CrlExtensions> [Optional]
836 If present, this element contains information about the list of extensions present in the CRL under
837 consideration. The **ExtensionType** is defined in Section 3.5.3.2.

843 3.5.3.5 OCSPValidityType

844 The **OCSPValidityType** contains information about an OCSP-response and its validity and is specified as
845 follows:

```

846
847     <complexType name="OCSPValidityType">
848         <sequence>
849             <element name="OCSPIdentifier" type="XAdES:OCSPIdentifierType"
850         />

```

```

851             <element name="OCSPValue" type="base64Binary"
852                     maxOccurs="1" minOccurs="0" />
853             <element name="OCSPContent" type="vr:OCSPContentType"
854                     maxOccurs="1" minOccurs="0" />
855             <element name="SignatureOK" type="vr:SignatureValidityType" />
856             <element name="CertificatePathValidity"
857                     type="vr:CertificatePathValidityType" />
858         </sequence>
859         <attribute name="Id" type="ID" use="optional" />
860     </complexType>

```

861

862 It contains the following attributes and elements:

863 **Id** [Optional]
864 This attribute contains an optional identifier for the element.

865 <OCSPIdentifier> [Required]
866 This element refers to an OCSP response according to [\[RFC2560\]](#).

867 <OCSPValue> [Optional]
868 This element contains the OCSP response (encoded in ASN.1) if the report option
869 <IncludeRevocationValues> has been set to 'true'.

870 <OCSPContent> [Optional]
871 This element contains the OCSP response in form of an equivalent XML structure if the report option
872 <ExpandBinaryValues> has been set to 'true'. The **OCSPContentType** is defined below.

873 <SignatureOK> [Required]
874 This element indicates whether the digital signature of the OCSP-response is mathematically correct
875 or not. The **SignatureValidityType** is defined in section 3.5.1.

876

877

878 <CertificatePathValidity> [Required]
879 This element contains the result of the validation of the certificate path of the certificate which has
880 been used to sign the OCSP-response. The **CertificatePathValidityType** is defined at the beginning
881 of Section 3.5.3.

882

883 The **OCSPContentType** is aligned to [\[RFC2560\]](#) specified as follows:

884

```

885 <complexType name="OCSPContentType">
886     <sequence>
887         <element name="Version" type="integer" />
888         <element name="ResponderID" type="string" />
889         <element name="producedAt" type="dateTime" />
890         <element name="Responses">
891             <complexType>
892                 <sequence maxOccurs="unbounded" minOccurs="0">
893                     <element name="SingleResponse"
894                         type="vr:SingleResponseType" />
895                 </sequence>
896             </complexType>
897             <element name="ResponseExtensions" type="vr:ExtensionsType"
898                         maxOccurs="1" minOccurs="0"/>
899         </sequence>
900     </complexType>

```

902
903 It contains the following elements:
904 <Version> [Required]
905 This element contains the version of the OCSP-response syntax.
906 <ResponderID> [Required]
907 This element contains the name of the OCSP-responder.
908 <producedAt> [Required]
909 This element contains the time at which the OCSP-responder produced the response.
910 <Responses> [Required]
911 This element contains an unbounded sequence of <SingleResponse> entries. The
912 **SingleResponseType** is defined below.
913 <ResponseExtensions> [Optional]
914 If present, this element contains information about the list of extensions present in the OCSP-response
915 under consideration. The **ExtensionsType** is defined in Section 3.5.3.2.

916
917 The **SingleResponseType** is specified as follows:
918

```
919 <complexType name="SingleResponseType">
920   <sequence>
921     <element name="CertID">
922       <complexType>
923         <sequence>
924           <element name="HashAlgorithm"
925             type="anyURI" />
926           <element name="IssuerNameHash"
927             type="hexBinary" />
928           <element name="IssuerKeyHash"
929             type="hexBinary" />
930           <element name="SerialNumber"
931             type="integer" />
932         </sequence>
933       </complexType>
934     </element>
935     <element name="CertStatus" type="vr:VerificationResultType" />
936     <element name="ThisUpdate" type="dateTime" />
937     <element name="NextUpdate" type="dateTime" maxOccurs="1"
938       minOccurs="0" />
939     <element name="SingleExtensions" type="vr:ExtensionsType"
940       maxOccurs="1" minOccurs="0" />
941   </sequence>
942 </complexType>
```

943
944 It contains the following elements:
945 <CertID> [Required]
946 This element contains a sequence of elements, which uniquely identify the certificate (cf. **[RFC2560]**,
947 Section 4.1.1).
948 <CertStatus> [Required]
949 This element contains information about the status of the certificate according to **[RFC2560]** using the
950 following URI in the ResultMajor-element:
951 • urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:certstatus:good

952 • [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:certstatus:revoked](#)
 953 • [urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:certstatus:unknown](#)
 954 If the certificate is revoked and the revocation reason is present, this information MUST be included in
 955 the `ResultMinor`-element as a URI defined in Section 3.5.3.4. In a similar fashion the revocation
 956 time MUST be indicated in the `ResultMessage`-element.
 957 <ThisUpdate> [Required]
 958 This element contains the time at which the status being indicated is known to be correct (cf.
 959 [\[RFC2560\]](#), Section 2.4).
 960 <NextUpdate> [Optional]
 961 This element contains, if present, the time until more recent information about the status of the
 962 certificate will be available (cf. [\[RFC2560\]](#), Section 2.4).
 963 <SingleExtensions> [Optional]
 964 If present, this element contains information about the list of extensions present in the
 965 SingleResponse-element. The **ExtensionType** is defined in Section 3.5.3.2.
 966

967 **3.5.3.6 TrustStatusListValidityType**

968
 969 The **TrustStatusListValidityType** is specified as follows:
 970

```

971       <complexType name="TrustStatusListValidityType">
972        <sequence>
973           <element ref="tsl:SchemeInformation" />
974           <element ref="tsl:TrustServiceProviderList" minOccurs="0" />
975           <element name="SignatureOK" type="vr:SignatureValidityType" />
976        </sequence>
977        <attribute name="TSLTag" type="tsl:TSLTagType" use="required" />
978        <attribute name="Id" type="ID" use="optional" />
979     </complexType>
  
```

980
 981 It contains the following attributes and elements:
 982 TSLTag [Required]
 983 This attribute shall facilitate the identification of the TSL as such. It will be a string with a fixed value.
 984 Its schema is defined in Section B.1.3.1 of [\[ETSI102231\]](#)
 985 Id [Optional]
 986 This attribute contains an optional identifier for the element.
 987 <SchemeInformation> [Required]
 988 This element contains general information about the circumstances how the TSL was issued. For
 989 details see Section B.2 of [\[ETSI102231\]](#).
 990 <TrustServiceProviderList> [Optional]
 991 This element contains, if present, a list of trustworthy service providers. For details see Section B.2.17
 992 of [\[ETSI102231\]](#).
 993 <SignatureOK> [Required]
 994 This element indicates, whether the digital signature of the TSL is mathematically correct or not. The
 995 **SignatureValidityType** is defined in section 3.5.1.

996 **3.5.4 PropertiesType**

997 The **PropertiesType** is used in the definition of the <DetailedReport>-element (see Section 3.5) and
998 is specified as follows:

999

```
1000 <complexType name="PropertiesType">
1001     <sequence>
1002         <element name="SignedProperties"
1003             type="vr:SignedPropertiesType" minOccurs="0" />
1004         <element name="UnsignedProperties"
1005             type="vr:UnsignedPropertiesType" minOccurs="0" />
1006     </sequence>
1007     <attribute name="Id" type="ID" use="optional" />
1008 </complexType>
```

1009

1010 It contains the following attributes and elements:

1011 **Id** [Optional]

1012 This attribute contains, if present, an optional identifier for the element.

1013 <SignedProperties> [Optional]

1014 This element contains information gathered during the verification of signed properties. Details of the
1015 SignedPropertiesType are specified in Section 3.5.4.1.

1016 <UnsignedProperties> [Optional]

1017 This element contains information gathered during the verification of unsigned properties. Details of the
1018 UnsignedPropertiesType are specified in Section 3.5.4.2.

1019 **3.5.4.1 Signed Properties**

1020 The **SignedPropertiesType** is aligned to [XAdES] structured as follows:

1021

```
1022 <complexType name="SignedPropertiesType">
1023     <sequence>
1024         <element name="SignedSignatureProperties"
1025             type="vr:SignedSignaturePropertiesType" maxOccurs="1"
1026             minOccurs="0" />
1027         <element name="SignedDataObjectProperties"
1028             type="vr:SignedDataObjectPropertiesType" minOccurs="0" />
1029     </sequence>
1030     <element name="Other" type="dss:AnyType" maxOccurs="1"
1031             minOccurs="0" />
1032     </sequence>
1033     <attribute name="Id" type="ID" use="optional" />
1034 </complexType>
```

1035

1036 It contains the following attributes and elements:

1037 **Id** [Optional]

1038 This attribute contains an optional identifier for the element.

1039 <SignedSignatureProperties> [Optional]

1040 This element contains information gathered during the verification of signed properties related to the
1041 signature itself. The **SignedSignaturePropertiesType** is defined in Section 3.5.4.1.1.

1042 <SignedDataObjectProperties> [Optional]

1043 This element contains information gathered during the verification of signed properties related to the
1044 signed data object. The **SignedDataObjectPropertiesType** is defined in Section 3.5.4.1.2.

1045 <other> [Optional]

1046 This element contains, if present, information about other signed properties.

1047 **3.5.4.1.1 SignedSignaturePropertiesType**

1048 The **SignedSignaturePropertiesType** is aligned to [RFC3275] defined as follows:

1049

```
1050 <complexType name="SignedSignaturePropertiesType">
1051   <sequence>
1052     <element ref="XAdES:SigningTime" maxOccurs="1" minOccurs="0" />
1053     <element ref="XAdES:SigningCertificate" maxOccurs="1" minOccurs="0" /
1054   />
1055     <element ref="XAdES:SignaturePolicyIdentifier" maxOccurs="1"
1056       minOccurs="0" />
1057     <choice maxOccurs="1" minOccurs="0">
1058       <element ref="XAdES:SignatureProductionPlace" />
1059       <element name="Location" type="string" />
1060     </choice>
1061     <element name="SignerRole" type="vr:SignerRoleType"
1062       minOccurs="0" />
1063   </sequence>
1064 </complexType>
```

1065

1066 It MAY contain the following elements:

1067 <XAdES:SigningTime> [Optional]

1068 This element contains, if present, the signing time (see Section 5.2.1 of [XAdES]).

1069 <XAdES:SigningCertificate> [Optional]

1070 This element contains, if present, a reference to the certificate upon which the signature is based (see
1071 Section 5.2.2 of [XAdES]).

1072 <XAdES:SignaturePolicyIdentifier> [Optional]

1073 This element references, if present, the policy under which the signature was produced (see Section
1074 5.2.3 of [XAdES]).

1075 <XAdES:SignatureProductionPlace> [Optional, Choice]

1076 This element contains, if present, information about the place where the signature was generated (see
1077 Section 5.2.7 of [XAdES]). This element SHOULD be used in case of a XAdES- or CAdES-based
1078 signature.

1079 <Location> [Optional, Choice]

1080 This element contains, if present, information about the place where the signature was generated (see
1081 Section 5.2.7 of [XAdES]). This element SHOULD be used in case of a PDF-based signature.

1082 <SignerRole> [Optional]

1083 This element contains, if present, information about the role of the signer (see Section 5.2.8 of
1084 [XAdES]).

1085

1086 The **SignerRoleType** is specified as follows:

1087

```
1088 <complexType name="SignerRoleType">
1089   <sequence>
1090     <element name="ClaimedRoles" />
```

```

1091             type="XAdES:ClaimedRolesListType" minOccurs="0" />
1092             <element name="CertifiedRoles"
1093                 type="vr:CertifiedRolesListType" minOccurs="0" />
1094         </sequence>
1095     </complexType>

```

1096

1097 It MAY contain the following elements:

1098 <ClaimedRoles> [Optional]

1099 This element contains information about the claimed roles of the signer. The information is directly extracted from the signature.

1100

1101 <CertifiedRoles> [Optional]

1102 This element contains information gathered during the verification of attribute certificates.

1103

1104 The **CertifiedRolesListType** is specified as follows:

1105

```

1106     <complexType name="CertifiedRolesListType">
1107         <sequence>
1108             <element name="AttributeCertificateValidity"
1109                 type="vr:AttributeCertificateValidityType"
1110                 maxOccurs="unbounded" />
1111         </sequence>
1112     </complexType>

```

1113

1114 It contains at least one <AttributeCertificateValidity>-element, which contains information about the content and validity of an attribute certificate according to [\[RFC3281\]](#). The **AttributeCertificateValidityType** is defined in Section 3.5.4.3.

1115

1116 3.5.4.1.2 SignedDataObjectPropertiesType

1117 The **SignedDataObjectPropertiesType** is defined as follows:

1118

```

1119
1120     <complexType name="SignedDataObjectPropertiesType">
1121         <sequence>
1122             <element ref="XAdES:DataObjectFormat" maxOccurs="unbounded"
1123                 minOccurs="0" />
1124             <choice maxOccurs="1" minOccurs="0">
1125                 <element ref="XAdES:CommitmentTypeIndication"
1126                     maxOccurs="unbounded" minOccurs="1"/>
1127                 <element name="Reason" type="string" />
1128             </choice>
1129             <element name="AllDataObjectsTimeStamp"
1130                 type="vr:TimeStampValidityType" minOccurs="0"
1131                 maxOccurs="unbounded" />
1132                 <element name="IndividualDataObjectsTimeStamp"
1133                     type="vr:TimeStampValidityType" minOccurs="0"
1134                     maxOccurs="unbounded" />
1135             </sequence>
1136             <attribute name="Id" type="ID" use="optional" />
1137     </complexType>

```

1138

1139 It contains the following attributes and elements:

1140 Id [Optional]

1141 This attribute contains an optional identifier for the element.

1142 <XAdES : DataObjectFormat> [Optional, Unbounded]

1143 This element contains information about the format of the signed data object (see Section 5.2.5 of

1144 [**XAdES**]). This information is simply extracted from the signature.

1145 <XAdES : CommitmentTypeIndication> [Choice, Unbounded]

1146 This element contains, if present, an indication of the type of commitment implied by the signature

1147 (see Section 5.2.6 of [**XAdES**]). This element SHOULD be used in case of a XAdES- or CAdES-based

1148 signature.

1149 <Reason> [Choice]

1150 This element contains, if present, a description of the reason of the signature generation. This element

1151 is only relevant in case of a PDF-based signature identified by a `FieldName`-attribute (cf. Section

1152 3.3).

1153 <AllDataObjectsTimeStamp> [Optional, Unbounded]

1154 This element contains, if present, verification results for time stamps covering all data objects (see

1155 Section 5.2.6 of [**XAdES**]). The **TimeStampValidityType** is described in Section 3.5.4.4.

1156 <IndividualDataObjectsTimeStamp> [Optional, Unbounded]

1157 This element contains, if present, verification results for time stamps covering only certain data objects

1158 (see Section 5.2.10 of [**XAdES**]). The **TimeStampValidityType** is described in section 3.5.4.4.

3.5.4.2 Unsigned Properties

1160 The **UnsignedPropertiesType** is specified as follows:

1161

```
1162 <complexType name="UnsignedPropertiesType">
1163   <sequence>
1164     <element name="UnsignedSignatureProperties"
1165       type="vr:UnsignedSignaturePropertiesType" minOccurs="0" />
1166     <element ref="XAdES:UnsignedDataObjectProperties"
1167       maxOccurs="1" minOccurs="0" />
1168     <element name="Other" type="dss:AnyType" maxOccurs="1"
1169       minOccurs="0">
1170       </element>
1171     </sequence>
1172     <attribute name="Id" type="ID" use="optional" />
1173   </complexType>
```

1174

1175 It contains the following attributes and elements:

1176 `Id` [Optional]

1177 This attribute contains an optional identifier for the element.

1178 <UnsignedSignatureProperties> [Optional]

1179 This element contains information gathered during the verification of the unsigned properties related to

1180 the signature itself. The **UnsignedSignaturePropertiesType** is defined below.

1181 <XAdES:UnsignedDataObjectProperties> [Optional]

1182 This element contains unsigned properties referring to the signed data objects. These properties are

1183 directly extracted from the signature.

1184 <Other> [Optional]

1185 This element MAY contain information about other unsigned properties.

1186

1187 The **UnsignedSignaturePropertiesType** is defined as follows:

1188

```
1189 <complexType name="UnsignedSignaturePropertiesType">
1190     <choice maxOccurs="unbounded">
1191         <element name="CounterSignature"
1192 type="vr:SignatureValidityType" />
1193         <element name="SignatureTimeStamp"
1194 type="vr:TimeStampValidityType" />
1195         <element ref="XAdES:CompleteCertificateRefs" />
1196         <element ref="XAdES:CompleteRevocationRefs" />
1197         <element ref="XAdES:AttributeCertificateRefs" />
1198         <element ref="XAdES:AttributeRevocationRefs" />
1199         <element name="SigAndRefsTimeStamp" type="vr:TimeStampValidityType"
1200 />
1201         <element name="RefsOnlyTimeStamp"
1202 type="vr:TimeStampValidityType" />
1203         <element name="CertificateValues"
1204 type="vr:CertificateValuesType" />
1205         <element name="RevocationValues"
1206 type="vr:RevocationValuesType" />
1207         <element name="AttrAuthoritiesCertValues"
1208             type="vr:CertificateValuesType" />
1209         <element name="AttributeRevocationValues"
1210             type="vr:RevocationValuesType" />
1211         <element name="ArchiveTimeStamp"
1212 type="vr:TimeStampValidityType" />
1213     </choice>
1214     <attribute name="Id" type="ID" use="optional" />
1215 </complexType>
```

1216

1217 It contains the following attributes and elements:

1218 `Id` [Optional]

1219 This attribute contains an optional identifier for the element.

1220 `<CounterSignature>` [Choice]

1221 This element contains the results of the verification of a counter signature (see Section 7.2.4 of
1222 **[XAdES]**). The **SignatureValidityType** is described in section 3.5.1.

1223 `<SignatureTimeStamp>` [Choice]

1224 This element contains verification results of a time stamp of the signature (see Section 7.3 of
1225 **[XAdES]**). The **TimeStampValidityType** is described in section 3.5.4.4.

1226 `<XAdES:CompleteCertificateRefs>` [Choice]

1227 This element contains references to the certificates used during verification of the signature (see
1228 Section 7.4.1 of **[XAdES]**). This information is simply extracted from the signature.

1229 `<XAdES:CompleteRevocationRefs>` [Choice]

1230 Contains references to the revocation data used for the verification of the signature (see Section 7.4.2
1231 of **[XAdES]**). This information is simply extracted from the signature.

1232 `<XAdES:AttributeCertificateRefs>` [Choice]

1233 Contains the references to the full set of attribute authorities certificates that have been used to
1234 validate the attribute certificate (see section 7.4.3 of **[XAdES]**). This information is simply extracted
1235 from the signature.

1236 `<XAdES:AttributeRevocationRefs>` [Choice]

1237 Contains the references to the full set of revocation data that have been used in the validation of the
1238 attribute certificate(s) present in the signature (see section 7.4.4 of **[XAdES]**).

1239 `<SigAndRefsTimeStamp>` [Choice]

1240 Contains verification results for a time stamp referring to the signature and references on certificates
 1241 and revocation data (see section 7.5.1 of [XAdES]). The **TimeStampValidityType** is described in
 1242 section 3.5.4.4.
 1243 <RefsOnlyTimeStamp> [Choice]
 1244 Contains verification results for a time stamp referring only to references on certificates and revocation
 1245 data (see section 7.5.2 of [XAdES]). The **TimeStampValidityType** is described in section 3.5.4.4.
 1246 <CertificateValues> [Choice]
 1247 Contains verification results for the certificates, which were used in the verification of the signature
 1248 (see section 7.6.1 of [XAdES]). The **CertificateValuesType** is defined below.
 1249 <RevocationValues> [Choice]
 1250 Contains verification results of the revocation data used in the verification of the signature (see section
 1251 7.6.2 of [XAdES]). The **RevocationValuesType** is defined below.
 1252 <AttrAuthoritiesCertValues> [Choice]
 1253 Contains verification results of the certificates of Attribute Authorities that have been used to validate
 1254 the attribute certificates, which are contained in the signature (see section 7.6.3 of [XAdES]). The
 1255 **CertificateValuesType** is defined below.
 1256 <AttributeRevocationValues> [Choice]
 1257 Contains verification results of the revocation data that have been used to validate the attribute
 1258 certificate when present in the signature (see section 7.6.4 of [XAdES]). The **RevocationValuesType**
 1259 is defined below.
 1260 <ArchiveTimeStamp> [Choice]
 1261 Contains verification results for a time stamp covering the complete signature including all attributes
 1262 (see section 7.7 of [XAdES]). The **TimeStampValidityType** is described in section 3.5.4.4.
 1263

1264 The **CertificateValuesType** is defined as follows:

1265

```

1266   <complexType name="CertificateValuesType">
1267     <choice minOccurs="0" maxOccurs="unbounded">
1268       <element name="EncapsulatedX509Certificate"
1269         type="vr:CertificateValidityType" />
1270       <element name="OtherCertificate" />
1271     </choice>
1272     <attribute name="Id" type="ID" use="optional" />
1273   </complexType>
  
```

1274

1275 It defines the following attributes and elements:

1276 **Id** [Optional]
 1277 This attribute contains an optional identifier for the element.
 1278 <EncapsulatedX509Certificate> [Optional, Unbounded, Choice]
 1279 Contains verification results for an X.509 certificate included in the signature. The
 1280 **CertificateValidityType** is defined in Section 3.5.3.1.
 1281 <OtherCertificate> [Optional, Unbounded, Choice]
 1282 This element contains verification results for other certificates included in the signature. If a certificate
 1283 with unknown format is included in the signature, a warning (error code
 1284 [urn:oasis:names:tc:dss:1.0:resultminor:certificateFormatNotCorrectWarning](#)) SHOULD be returned.
 1285

1286 The **RevocationValuesType** is defined as follows:

1287

```
1288 <complexType name="RevocationValuesType">
1289     <sequence>
1290         <element name="CRLValues" minOccurs="0">
1291             <complexType>
1292                 <sequence maxOccurs="unbounded" minOccurs="1">
1293                     <element name="VerifiedCRL"
1294 type="vr:CRLValidityType" />
1295                 </sequence>
1296             </complexType>
1297         </element>
1298         <element name="OCSPValues" minOccurs="0">
1299             <complexType>
1300                 <sequence maxOccurs="unbounded" minOccurs="1">
1301                     <element name="VerifiedOCSPResponse"
1302 type="vr:OCSPValidityType" />
1303                 </sequence>
1304             </complexType>
1305         </element>
1306         <element name="OtherValues" type="dss:AnyType" minOccurs="0"
1307 />
1308     </sequence>
1309     <attribute name="Id" type="ID" use="optional" />
1310 </complexType>
```

1311

1312 It contains the following attributes and elements:

1313 `Id` [Optional]

1314 This attribute contains an optional identifier for the element.

1315 `<CRLValues>` [Optional]

1316 Contains the verification results for all CRLs included in a signature. The **CRLValidityType** is defined
1317 in Section 3.5.3.4.

1318 `<OCSPValues>` [Optional]

1319 Contains the verification results for all OCSP responses included in a signature. The **OCSPValidityType** is defined in Section 3.5.3.5.

1321 `<OtherValues>` [Optional]

1322 This element MAY contain verification results for other revocation data included in the signature. If
1323 other revocation data with unknown format is included in the signature, a warning (error
1324 `urn:oasis:names:tc:dss:1.0:resultminor:improperRevocationInformation`) SHOULD be returned.

1325

1326 3.5.4.3 AttributeCertificateValidityType

1327 The **AttributeCertificateValidityType** is defined as follows:

1328

```
1329 <complexType name="AttributeCertificateValidityType">
1330     <sequence>
1331         <element name="AttributeCertificateIdentifier"
1332             type="vr:AttrCertIDType" maxOccurs="1" minOccurs="0" />
1333         <element name="AttributeCertificateValue" type="base64Binary"
1334             maxOccurs="1" minOccurs="0" />
1335         <element name="AttributeCertificateContent"
1336             type="vr:AttributeCertificateContentType" maxOccurs="1"
1337             minOccurs="0" />
1338         <element name="SignatureOK" type="vr:SignatureValidityType" />
1339         <element name="CertificatePathValidity"
```

```

1340                               type="vr:CertificatePathValidityType" />
1341                         </sequence>
1342                   </complexType>
1343
1344 It contains the following elements:
1345 <AttributeCertificateIdentifier> [Optional]
1346 This element MAY refer to an X.509v3 attribute certificate according to [RFC3281]. The structure of
1347 the AttrCertIDType is defined below.
1348 <AttributeCertificateValue> [Optional]
1349 This element MAY contain the certificate in binary form (coded in ASN.1), if the report option
1350 <IncludeCertificateValues> is set to 'true'.
1351 <AttributeCertificateContent> [Optional]
1352 This element MAY contain an XML-based analogue of the content of the certificate, if the report option
1353 <ExpandBinaryValues> is set to 'true'. The structure of the
1354 AttributeCertificateContentType is defined below.
1355 <SignatureOK> [Required]
1356 This element indicates, whether the digital signature is mathematically valid or not. The
1357 SignatureValidityType is defined in section 3.5.1.
1358 <CertificatePathValidity> [Required]
1359 This element contains the result of the validation of the certificate path of the certificate which has
1360 been used to sign the attribute certificate. The CertificatePathValidityType is defined at the
1361 beginning of Section 3.5.3.
1362
1363 The AttrCertIDType is structured as follows:
1364
1365 <complexType name="AttrCertIDType">
1366   <sequence>
1367     <element name="Holder" type="vr:EntityType" maxOccurs="1"
1368 minOccurs="0"/>
1369     <element name="Issuer" type="vr:EntityType" />
1370     <element name="SerialNumber" type="integer"></element>
1371   </sequence>
1372 </complexType>
1373
1374 It contains the following elements:
1375 <Holder> [Optional]
1376 This element contains, if present, information about the holder of the certificate. The structure of the
1377 EntityType is defined below.
1378 <Issuer> [Required]
1379 This element contains information about the issuer of the attribute certificate. The structure of the
1380 EntityType is defined below.
1381 <SerialNumber> [Required]
1382 This element contains the serial number of the attribute certificate, which (together with the information
1383 provided in the <Issuer>-element) uniquely identifies the attribute certificate.
1384
1385 The EntityType is aligned to the structure of Holder and V2Form in [RFC3281] and is defined as
1386 follows:
```

1387

```
1388 <complexType name="EntityType">
1389     <sequence>
1390         <element name="BaseCertificateID"
1391             type="ds:X509IssuerSerialType" maxOccurs="1"
1392             minOccurs="0"/>
1393         <element name="Name" type="string" maxOccurs="1"
1394             minOccurs="0"/>
1395         <element name="Other" type="dss:AnyType" maxOccurs="1"
1396             minOccurs="0"></element>
1397     </sequence>
1398 </complexType>
```

1399

1400 It SHOULD contain sufficient information to identify the entity uniquely and MAY contain the following
1401 optional elements:

1402 <BaseCertificateID> [Optional]

1403 This element identifies, if present, the public-key certificate of the entity. The structure of the
1404 ds:X509IssuerSerialType is defined in [\[RFC3275\]](#).

1405 <Name> [Optional]

1406 This element contains, if present, the name of the entity.

1407 <Other> [Optional]

1408 This element MAY contain other information, which is used to identify the entity.

1409

1410 The **AttributeCertificateContentType** contains the content of an attribute certificate according to
1411 [\[RFC3281\]](#) as XML structure and is structured as follows:

1412

```
1413 <complexType name="AttributeCertificateContentType">
1414     <sequence>
1415         <element name="Version" minOccurs="0" type="integer" />
1416         <element name="Holder" type="vr:EntityType" />
1417         <element name="Issuer" type="vr:EntityType" />
1418         <element name="SignatureAlgorithm" type="anyURI" />
1419         <element name="SerialNumber" type="integer" />
1420         <element name="AttCertValidityPeriod"
1421             type="vr:ValidityType" />
1422         <element name="Attributes">
1423             <complexType>
1424                 <sequence minOccurs="0" maxOccurs="unbounded">
1425                     <element name="Attribute"
1426                         type="vr:AttributeType" />
1427                 </sequence>
1428             </complexType>
1429         </element>
1430         <element name="IssuerUniqueID" type="hexBinary" maxOccurs="1"
1431             minOccurs="0"/>
1432         <element name="Extensions" minOccurs="0"
1433             type="vr:ExtensionsType" />
1434     </sequence>
1435 </complexType>
```

1436

1437 It contains the following elements:

1438 <Version> [Optional]

1439 This element contains, if present, the version of the attribute certificate.

1440 <Holder> [Required]
1441 This element contains information about the holder of the certificate. The structure of the **EntityType**
1442 is defined above.
1443 <Issuer> [Required]
1444 This element contains the issuer of the attribute certificate. The structure of the **EntityType** is defined
1445 above.
1446 <SignatureAlgorithm> [Required]
1447 This element contains an identifier of the used signature algorithm.
1448 <SerialNumber> [Required]
1449 This element contains the serial number of the attribute certificate.
1450 <AttCertValidityPeriod> [Required]
1451 This element contains the validity period of the attribute certificate. The **ValidityType** is defined in
1452 section 3.5.3.2.
1453 <Attributes> [Optional, Unbounded]
1454 This element contains, if present, a list of attributes. The **AttributeType** is defined below.
1455 <IssuerUniqueID> [Optional]
1456 This element contains, if present, a unique identifier of the issuer of the attribute certificate.
1457 <Extensions> [Optional]
1458 If present, this element contains information about the list of extensions present in the attribute
1459 certificate. The **ExtensionType** is defined in Section 3.5.3.2.
1460
1461 The **AttributeType** is defined as follows:
1462

```
1463 <complexType name="AttributeType">  
1464     <sequence>  
1465         <element name="Type" type="anyURI" />  
1466         <element name="Value" type="dss:AnyType" maxOccurs="unbounded"  
1467             minOccurs="0"></element>  
1468     </sequence>  
1469 </complexType>
```

1470
1471 It contains the following elements:
1472 <Type> [Required]
1473 This element MUST contain an identifier for the type of the attribute in the <Code>-element and MAY
1474 contain further information.
1475 <Value> [Optional, Unbounded]
1476 This element MAY contain any number of attribute values.
1477

1478 3.5.4.4 TimeStampValidityType

1479 The **TimeStampValidityType** is structured as follows:
1480

```
1481 <complexType name="TimeStampValidityType">  
1482     <sequence>  
1483         <element name="FormatOK" type="vr:VerificationResultType" />  
1484         <element name="TimeStampContent" type="vr:TstContentType"
```

```

1485                               maxOccurs="1" minOccurs="0" />
1486                         <element name="MessageHashAlgorithm"
1487                           type="vr:AlgorithmValidityType"
1488                             maxOccurs="1" minOccurs="0" />
1489                           <element name="SignatureOK"
1490                             type="vr:SignatureValidityType" />
1491                           <element name="CertificatePathValidity"
1492                             type="vr:CertificatePathValidityType" />
1493                         </sequence>
1494                         <attribute name="Id" type="ID" use="optional" />
1495                     </complexType>

```

1496

1497 It contains the following elements and attributes:

1498 **Id** [Optional]

1499 This attribute contains an optional identifier for the element.

1500 <FormatOK> [Required]

1501 This element indicates, whether the format of the time stamp is ok or not. More information on the use
 1502 of the **VerificationResultType** may be found in Section 3.4.

1503 <TimeStampContent> [Optional]

1504 This element contains the content of time stamp in form of an XML structure, if the report option
 1505 <ExpandBinaryValues> is set to 'true'. The **TstContentType** is specified below.

1506 <MessageHashAlgorithm> [Optional]

1507 This element contains, if present, information about the message hash algorithm and its suitability.
 1508 The **AlgorithmValidityType** is defined in Section 3.5.2.

1509 <SignatureOK> [Required]

1510 This element indicates, whether the digital signature is mathematically valid or not. The
 1511 **SignatureValidityType** is defined in Section 3.5.1.

1512 <CertificatePathValidity> [Required]

1513 This element contains the result of the validity check of the certificate. The
 1514 **CertificatePathValidityType** is defined in Section 3.5.3.

1515

1516 The **TstContentType** complex type is defined as follows:

1517

```

1518     <complexType name="TstContentType">
1519       <sequence>
1520         <element ref="dss:TstInfo" maxOccurs="1" minOccurs="0"/>
1521         <element name="Other" type="dss:AnyType" maxOccurs="1"
1522           minOccurs="0"/>
1523       </sequence>
1524     </complexType>

```

1525 It contains the following elements:

1526 <dss:TstInfo> [Optional]

1527 This element MAY contain the standard content of a time stamp as defined in Section 5.1.2 of
 1528 **[DSSCore]**. Note that there is a straightforward mapping from the **TSTInfo-Element** according to
 1529 **[RFC3161]** to the present structure.

1530 <other> [Optional]

1531 This element MAY contain other information included in the time stamp.

1532 **3.5.5 Element <IndividualTimeStampReport>**

1533 The <IndividualTimeStampReport>-element MAY appear in the <Details>-element within the
1534 <IndividualReport>-element defined in Section 3.3. This element is defined as follows:

```
1535 <element name="IndividualTimeStampReport" type="vr:TimeStampValidityType" />
```

1536 The **TimeStampValidityType** is defined in Section 3.5.4.4.

1537 **3.5.6 Element <IndividualCertificateReport>**

1538 The <IndividualCertificateReport>-element MAY appear in the <Details>-element within the
1539 <IndividualReport>-element defined in Section 3.3. This element is defined as follows:

```
1540 <element name="IndividualCertificateReport" type="vr:CertificateValidityType" />
```

1542 The **CertificateValidityType** is defined in Section 3.5.3.1.

1543 **3.5.7 Element <IndividualAttributeCertificateReport>**

1544 The <IndividualAttributeCertificateReport>-element MAY appear in the <Details>-
1545 element within the <IndividualReport>-element defined in Section 3.3. This element is defined as
1546 follows:

```
1547 <element name="IndividualAttributeCertificateReport" type="vr:AttributeCertificateValidityType" />
```

1549 The **AttributeCertificateValidityType** is defined in Section 3.5.4.3.

1550 **3.5.8 Element <IndividualCRLReport>**

1551 The <IndividualCRLReport>-element MAY appear in the <Details>-element within the
1552 <IndividualReport>-element defined in Section 3.3. This element is defined as follows:

```
1553 <element name="IndividualCRLReport" type="vr:CRLValidityType" />
```

1554 The **CRLValidityType** is defined in Section 3.5.3.4.

1555 **3.5.9 Element <IndividualOCSPReport>**

1556 The <IndividualOCSPReport>-element MAY appear in the <Details>-element within the
1557 <IndividualReport>-element defined in Section 3.3. This element is defined as follows:

```
1558 <element name="IndividualOCSPReport" type="vr:OCSPValidityType" />
```

1559 The **OCSPValidityType** is defined in Section 3.5.3.5.

1560 **3.5.10 Element <EvidenceRecordReport>**

1561 The <EvidenceRecordReport>-element MAY appear in the <Details>-element within the
1562 <IndividualReport>-element defined in Section 3.3. This element is defined as follows:

```
1563 <element name="EvidenceRecordReport" type="vr:EvidenceRecordValidityType" />
```

1564 The **EvidenceRecordValidityType** is based on the definition of the EvidenceRecord-element in
1565 [RFC4998] defined as follows:

```
1566 <complexType name="EvidenceRecordValidityType">
1567   <sequence>
1568     <element name="FormatOK" type="vr:VerificationResultType" />
```

```

1569      <element name="Version" type="integer"
1570          maxOccurs="1" minOccurs="0">
1571      </element>
1572      <element name="DigestAlgorithm"
1573          type="vr:AlgorithmValidityType" maxOccurs="unbounded"
1574      minOccurs="0">
1575          </element>
1576          <element name="CryptoInfos" maxOccurs="1" minOccurs="0">
1577              <complexType>
1578                  <sequence>
1579                      <element name="Attribute"
1580                          type="vr:AttributeType"
1581                      maxOccurs="unbounded" minOccurs="1">
1582                          </element>
1583                      </sequence>
1584                  </complexType>
1585          </element>
1586          <element name="EncryptionInfo" maxOccurs="1" minOccurs="0">
1587              <complexType>
1588                  <sequence>
1589                      <element name="EncryptionInfoType"
1590                          type="vr:AlgorithmValidityType">
1591                      </element>
1592                      <element name="EncryptionInfoValue"
1593                          type="dss:AnyType">
1594                      </element>
1595                  </sequence>
1596              </complexType>
1597          </element>
1598          <element name="ArchiveTimeStampSequence" maxOccurs="1"
1599              minOccurs="1">
1600              <complexType>
1601                  <sequence maxOccurs="unbounded" minOccurs="0">
1602                      <element name="ArchiveTimeStampChain">
1603                          <complexType>
1604                              <sequence maxOccurs="unbounded"
1605                                  minOccurs="0">
1606                                  <element
1607                                      name="ArchiveTimeStamp"
1608
1609                                      type="vr:ArchiveTimeStampValidityType">
1610
1611                                      </element>
1612
1613                                      </sequence>
1614
1615                  </complexType>
1616
1617              </sequence>
1618              <attribute name="Id" type="ID" use="optional" />
1619          </complexType>

```

1620
1621 It contains the following elements and attributes:

1622 Id [Optional]

1623 This attribute contains an optional identifier for the element.

1624 <FormatOK> [Required]

1625 This element indicates, whether the format of the evidence record according to **[RFC4998]** is ok or
1626 not. More information on the use of the **VerificationResultType** may be found in Section 3.4.

1627 <Version> [Optional]

1628 This element contains, if present, the version of the Evidence Record Syntax.
 1629 <DigestAlgorithm> [Optional, unbounded]
 1630 This element appears for each hash algorithm used to produce the evidence record and contains
 1631 information about the hash algorithm and possibly its suitability. The **AlgorithmValidityType** is
 1632 defined in Section 3.5.2.
 1633 <CryptoInfos> [Optional]
 1634 This element MAY contain further data useful in the validation of the <ArchiveTimeStampSequence>-
 1635 element. As explained in [RFC4998] this MAY include possible Trust Anchors, certificates, revocation
 1636 information, or the information concerning the suitability of cryptographic algorithms.
 1637 <EncryptionInfo> [Optional]
 1638 This element MAY contain the necessary information to support encrypted content (cf. [RFC4998],
 1639 Section 6.1).
 1640 <ArchiveTimeStampSequence> [Required]
 1641 This element is required and MAY contain a sequence of <ArchiveTimeStampChain>-elements (cf.
 1642 [RFC4998], Section 5), which in turn MAY contain a sequence of <ArchiveTimeStamp>-elements,
 1643 which are of type **ArchiveTimeStampValidityType** defined below.
 1644
 1645 The **ArchiveTimeStampValidityType** is based on the definition of the **ArchiveTimeStamp**-element in
 1646 [RFC4998] defined as follows:
 1647

```

1648 <complexType name="ArchiveTimeStampValidityType">
1649   <sequence>
1650     <element name="FormatOK" type="vr:VerificationResultType" />
1651     <element name="DigestAlgorithm" type="vr:AlgorithmValidityType"
1652       maxOccurs="1" minOccurs="0" />
1653     <element name="Attributes" maxOccurs="1" minOccurs="0">
1654       <complexType>
1655         <sequence>
1656           <element name="Attribute" type="vr:AttributeType"
1657             maxOccurs="unbounded" minOccurs="1"/>
1658         </sequence>
1659       </complexType>
1660     </element>
1661     <element name="ReducedHashTree" maxOccurs="1" minOccurs="0">
1662       <complexType>
1663         <sequence maxOccurs="unbounded" minOccurs="1">
1664           <element name="PartialHashTree">
1665             <complexType>
1666               <sequence maxOccurs="unbounded"
1667               minOccurs="1">
1668                 <element name="HashValue"
1669                   type="vr:HashValueType">
1670                   <complexType>
1671                     <sequence>
1672                       <element name="HashValue" type="vr:HashValueType" />
1673                     </sequence>
1674                   </complexType>
1675                 </element>
1676               </sequence>
1677             </complexType>
1678           </element>
1679           <element name="TimeStamp"
1680             type="vr:TimeStampValidityType" />
1681         </sequence>
1682         <attribute name="Id" type="ID" use="optional" />
1683       </complexType>
1684     </element>
1685   </sequence>
1686 </complexType>
  
```

1683

1684 It contains the following elements and attributes:

1685 **Id** [Optional]
 This attribute contains an optional identifier for the element.

1686 **<FormatOK>** [Required]
 This element indicates, whether the format of the evidence record according to **[RFC4998]** is ok or not. More information on the use of the **VerificationResultType** may be found in Section 3.4.

1687 **<DigestAlgorithm>** [Optional]
 This element contains, if present, information about the hash algorithm and possibly its suitability. The **AlgorithmValidityType** is defined in Section 3.5.2.

1688 **<Attributes>** [Optional]
 This element contains, if present, information about further attributes related to the archive time stamp.

1689 **<ReducedHashTree>** [Optional]
 This element MAY contain a sequence of **<PartialHashTree>**-elements, which in turn contain a list of **<HashValue>**-elements of type **HashValueType** defined below.

1690 **<TimeStamp>** [Required]
 This element is of type **TimeStampValidityType** (cf. Section 3.5.4.4) and contains information about the validity of the conventional time stamp, which is included in the present archive time stamp.

1691

1692

1693 The **HashValueType** is used for the **<HashValue>**-element within the **<PartialHashTree>**-element above and is defined as follows:

```

1694 <complexType name="HashValueType">
1695     <sequence>
1696         <element name="HashValue" type="hexBinary" />
1697     </sequence>
1698     <attribute name="HashedObject" type="IDREF" use="optional"/>
1699 </complexType>

```

1700 It contains the following elements and attributes:

1701 **HashedObject** [Optional]
 This attribute MAY be used to point to the object, which served as pre-image of the hash value.

1702 **<HashValue>** [Required]
 This element contains the hash value produced by applying the hash algorithm specified by the **<DigestAlgorithm>**- or **<TimeStamp>**-element to the data specified by the **HashedObject** attribute.

1703

1704

1705

1706

1707

1708

1709

1710

1711

1712

1713

1714

1715

1716

1717

1718

1719 4 Conformance

1720 This profile defines three conformance levels:

- 1721 • Level 1 - "Basic",
- 1722 • Level 2 - "Comprehensive" and
- 1723 • Level 3 - "Comfortable".

1724 4.1 Level 1 – “Basic”

1725 The conformance level “Basic” allows to return individual verification results for each signature contained
1726 in a `<dss:VerifyRequest>`. For this purpose the `<dss:VerifyResponse>` MUST contain in
1727 `<dss:OptionalOutputs>` a `<VerificationReport>`-element, as specified in Section 3.2. The
1728 `<VerificationReport>`-element MUST contain an `<IndividualSignatureReport>`-element (see
1729 Section 3.3) for each signature or time stamp (i.e. `<dss:SignatureObject>`) contained in the
1730 `<VerifyRequest>`-element.

1731 The `<Details>`-element within `<IndividualSignatureReport>` MAY contain other elements, such
1732 as the Optional Outputs defined in Section 4.5 of [DSSCore].

1733 4.2 Level 2 – “Comprehensive”

1734 The conformance level “Advanced” comprises all requirements of conformance Level 1 (“Basic”), as
1735 explained in Section 4.1. Furthermore the `<Details>`-element within each `<IndividualReport>`
1736 MUST contain exactly one object-specific element, which documents the detailed verification results for
1737 the signatures or validation data under consideration. While it is REQUIRED in this conformance level
1738 that certificate values and revocation values are included into the verification report if requested by the
1739 `IncludeCertificateValues-` and `IncludeRevocationValues`-element within the
1740 `ReturnVerificationReport`-element (cf. Section 3.1), it is NOT REQUIRED in this conformance level
1741 to expand those values and other relevant validation data to XML-structures if requested by the
1742 `ExpandBinaryValues`-element.

1743 The object-specific detail elements defined in this specification are given as follows:

- 1744 • `<DetailedSignatureReport>` (cf. Section 3.5) - is used for the verification of (advanced)
1745 electronic signatures.
- 1746 • `<IndividualTimeStampReport>` (cf. Section 3.5.5) – is used for the verification of individual time
1747 stamps according to [RFC3161], which are not included in a signature.
- 1748 • `<IndividualCertificateReport>` (cf. Section 3.5.6) – is used for the verification of individual
1749 certificates according to [RFC5280], which are not included in a signature.
- 1750 • `<IndividualAttributeCertificateReport>` (cf. Section 3.5.7) - is used for the verification of individual attribute certificates according to [RFC3281], which are not included in a signature.
- 1752 • `<IndividualCRLReport>` (cf. Section 3.5.8) - is used for the verification of individual CRLs
1753 according to [RFC5280], which are not included in a signature.
- 1754 • `<IndividualOCSPReport>` (cf. Section 3.5.9) - is used for the verification of individual OCSP-
1755 responses according to [RFC2560], which are not included in a signature.
- 1756 • `<EvidenceRecordReport>` (cf. Section 3.5.10) – is used for the verification of evidence records
1757 according to [RFC4998].

1758 Other object-specific detail elements MAY be defined in other profiles.

1759 **4.3 Level 3 – “Convenient”**

1760 The conformance Level 3 (“Convenient”) comprises all requirements of the conformance Level 2
1761 (“Comprehensive”), as explained in Section 4.2. Furthermore the binary values of the validation data
1762 MUST be expanded to the corresponding XML-structures, if this is requested by the
1763 `ExpandBinaryValues`-element within the `ReturnVerificationReport`-element (cf. Section 3.1).

1764 **A. Acknowledgements**

1765 The following individuals have participated in the creation of this specification and are gratefully
1766 acknowledged:

1767 **Participants:**

- 1768 • Juan-Carlos Cruellas
- 1769 • Andreas Kühne
- 1770 • Ezer Farhi
- 1771 • Stefan Drees
- 1772 • Pim van der Eijk
- 1773 • Clemens Orthacker
- 1774 • Marta Cruellas
- 1775 • Konrad Lanz

1776

B. Revision History

1777

Revision	Date	Editor	Changes Made
R1	19.07.2009	Detlef Hühnlein	CD1 version on current OASIS template

1778

1779