

## Profile for comprehensive multi-signature verification reports for OASIS Digital Signature Services Version 1.0

OASIS Working Draft (WD3)

22 May 2009

**Specification URIs:**

**This Version:**

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profile-for-comprehensive-signature-verification-report-v1.0-os.html>

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profile-for-comprehensive-signature-verification-report-v1.0-os.pdf>

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profile-for-comprehensive-signature-verification-report-v1.0-os.doc>

**Latest Version:**

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profile-for-comprehensive-signature-verification-report-v1.0-os.html>

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profile-for-comprehensive-signature-verification-report-v1.0-os.pdf>

<http://docs.oasis-open.org/dss/v1.0/oasis-dss-profile-for-comprehensive-signature-verification-report-v1.0-os.doc>

**Technical Committee:**

OASIS Digital Signature Services TC

**Chair(s):**

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

**Editor(s):**

Detlef Hühnlein, Federal Ministry of the Interior, Germany (FMI)

**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.

**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 membership of OASIS on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

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

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

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

---

## Notices

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

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

Copyright © OASIS® 1993–2009. All Rights Reserved.

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 paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document 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 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

The names "OASIS" 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
2.5	Conformance Levels.....	8
2.5.1	Level “Basic”.....	9
2.5.2	Level “Comprehensive”.....	9
2.5.3	Level “Convenient”.....	9
3	Verification Reports within DSS Verifying Protocol.....	10
3.1	Element <ReturnVerificationReport>.....	10
3.2	Element <VerificationReport>.....	11
3.3	Element <IndividualReport>.....	12
3.4	VerificationResultType.....	14
3.5	Element <DetailedSignatureReport>.....	14
3.5.1	SignatureValidityType.....	15
3.5.2	AlgorithmValidityType.....	16
3.5.3	CertificatePathValidityType.....	16
3.5.3.1	CertificateValidityType.....	18
3.5.3.2	CertificateContentType.....	19
3.5.3.3	CertificateStatusType.....	21
3.5.3.4	CRLValidityType.....	22
3.5.3.5	OCSPValidityType.....	24
3.5.3.6	TrustStatusListValidityType.....	26
3.5.4	PropertiesType.....	27
3.5.4.1	Signed Properties.....	27
3.5.4.2	Unsigned Properties.....	30
3.5.4.3	AttributeCertificateValidityType.....	33
3.5.4.4	TimeStampValidityType.....	37
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
A.	Acknowledgements.....	42

---

# 1 Introduction

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

## 1.1 Terminology

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this specification are to be interpreted as described in IETF RFC 2119 [RFC2119]. These keywords are capitalized when used to unambiguously specify requirements over protocol features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

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

## 1.2 Normative References

- [CAAdES] ETSI: *Electronic Signature Formats*, Electronic Signatures and Infrastructures (ESI) – Technical Specification, ETSI TS 101 733 V1.7.4, 2008-07, <http://www.etsi.org>
- [Core-XSD] S. Drees et al.: *DSS Schema*. OASIS, February 2007, <http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-schema-v1.0-os.xsd>
- [DSSCore] S. Drees et al.: *Digital Signature Service Core Protocols and Elements*. OASIS Standard, February 2007, <http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.pdf>
- [DSSAdES] S. Drees et al.: *Advanced Electronic Signature Profiles of the OASIS Digital Signature Service Version 1.0*, OASIS Standard, April 2007, <http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-AdES-spec-v1.0-os.pdf>
- [DSSSigG] A. Kuehne: *German Signature Law Profile of the OASIS Digital Signature Service Version 1.0*, OASIS Standard, April 2007, [http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles\\_german\\_signature\\_law-spec-v1.0-os.pdf](http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles_german_signature_law-spec-v1.0-os.pdf)
- [DSSVR-XSD] D. Hühnlein: *DSS Verification Report Schema*, Working Draft (WD3), 22<sup>nd</sup> May 2009, <http://www.oasis-open.org/apps/org/workgroup/dss-x/download.php/32628/VerificationReport-WD3.xsd>
- [DSSVisSig] E. Farhi: *Visual Signature Profile of the OASIS Digital Signature Services*, 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>
- [EC/1999/93] *Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures*, [http://europa.eu.int/eurlex/pri/en/oj/dat/2000/l\\_013/l\\_01320000119en00120020.pdf](http://europa.eu.int/eurlex/pri/en/oj/dat/2000/l_013/l_01320000119en00120020.pdf)
- [ETSI102231] ETSI: *ETSI TS 102231 Electronic Signatures and Infrastructure (ESI): Provision of harmonized Trust-service status information*. Version 2.1.1 of March 2006, via <http://www.etsi.org>
- [RFC2119] S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- [RFC2560] M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams: *X.509 Internet Public Key Infrastructure - Online Certificate Status Protocol – OCSP*, IETF RFC 2560, <http://www.ietf.org/rfc/rfc3161.txt>

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

## 72 **1.3 Namespaces**

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

76 This schema is associated with the following XML namespace:

77 `urn:oasis:names:tc:dss:1.0:profiles:verificationreport:schema#`

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

79

80 Conventional XML namespace prefixes are used in this document:

- 81 • The prefix `vr:` (or no prefix) stands for this profiles namespace **[DSSVR-XSD]**.
- 82 • The prefix `ds:` stands for the W3C XML Signature namespace **[XMLSig]**.
- 83 • The prefix `dss:` stands for the DSS core namespace **[Core-XSD]**.
- 84 • The prefix `saml:` stands for the OASIS SAML Schema namespace **[SAMLCore1.1]**.
- 85 • The prefix `ts1:` stands for the ETSI Trust-service status information namespace **[ETSI102231]**.
- 86 • The prefix `xades:` stands for ETSI XML Advanced Electronic Signatures (XAdES) document  
 87 **[XAdES]**.

88

89 Applications MAY use different namespace prefixes, and MAY use whatever namespace  
90 defaulting/scoping conventions they desire, as long as they are compliant with the Namespaces in XML  
91 specification [**XML-ns**].  
92

---

## 93 2 Profile Features

### 94 2.1 Overview

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

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

103 Therefore the present profile defines how

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

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

### 112 2.2 Scope

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

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

- 116 • the type of signature object,
- 117 • the transport binding or
- 118 • the security binding.

119

### 120 2.3 Relationship To Other Profiles

121 This profile is based directly on the **[DSSCore]**. This profile is intended to be combined with other profiles  
122 freely.

### 123 2.4 Profile Identifier

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

125 `urn:oasis:names:tc:dss:1.0:profiles:verificationreport`

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

### 127 2.5 Conformance Levels

128 This profile differentiates between three conformance levels “Basic”, “Comprehensive” and “Comfortable”.



## 129 **2.5.1 Level “Basic”**

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

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

## 138 **2.5.2 Level “Comprehensive”**

139 The conformance level “Advanced” comprises all requirements of the conformance level “Basic”, as  
140 explained in Section 2.5.1. Furthermore the <Details>-element within each <IndividualReport>  
141 MUST contain exactly one object-specific element, which documents the detailed verification results for  
142 the signatures or validation data under consideration. While it is REQUIRED in this conformance level  
143 that certificate values and revocation values are included into the verification report if requested by the  
144 Include  
145 CertificateValues- and IncludeRevocationValues-element within the ReturnVerification  
146 Report-element (cf. Section 3.1), it is NOT REQUIRED in this conformance level to expand those values  
147 and other relevant validation data to XML-structures if requested by the ExpandBinaryValues-element.

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

- 149 • <DetailedSignatureReport> (cf. Section 3.5) - is used for the verification of (advanced)  
150 electronic signatures.
- 151 • <IndividualTimeStampReport> (cf. Section 3.5.5) – is used for the verification of individual time  
152 stamps according to [RFC3161], which are not included in a signature.
- 153 • <IndividualCertificateReport> (cf. Section 3.5.6) – is used for the verification of individual  
154 certificates according to [RFC5280], which are not included in a signature.
- 155 • <IndividualAttributeCertificateReport> (cf. Section 3.5.7) - is used for the verification  
156 of individual attribute certificates according to [RFC3281], which are not included in a signature.
- 157 • <IndividualCRLReport> (cf. Section 3.5.8) - is used for the verification of individual CRLs  
158 according to [RFC5280], which are not included in a signature.
- 159 • <IndividualOCSPReport> (cf. Section 3.5.9) - is used for the verification of individual OCSP-  
160 responses according to [RFC2560], which are not included in a signature.
- 161 • <EvidenceRecordReport> (cf. Section 3.5.10) – is used for the verification of evidence records  
162 according to [RFC4998].

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

## 164 **2.5.3 Level “Convenient”**

165 The conformance level “Convenient” comprises all requirements of the conformance level  
166 “Comprehensive”, as explained in Section 2.5.2. Furthermore the binary values of the validation data  
167 MUST be expanded to the corresponding XML-structures, if this is requested by the  
168 ExpandBinaryValues-element within the ReturnVerificationReport-element (cf. Section 3.1).

---

## 3 Verification Reports within DSS Verifying Protocol

169  
170

### 3.1 Element <ReturnVerificationReport>

171  
172  
173  
174

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

175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191

```
<element name="ReturnVerificationReport">
  <complexType>
    <sequence>
      <element name="IncludeVerifier" type="boolean" maxOccurs="1"
        minOccurs="0" default="true" />
      <element name="IncludeCertificateValues" type="boolean" maxOccurs="1"
        minOccurs="0" default="false" />
      <element name="IncludeRevocationValues" type="boolean" maxOccurs="1"
        minOccurs="0" default="false" />
      <element name="ExpandBinaryValues" type="boolean" maxOccurs="1"
        minOccurs="0" default="false"/>
      <element name="ReportDetailLevel" type="anyURI" maxOccurs="1"
        minOccurs="0" default="urn:oasis:names:tc:dss:1.0:profiles:
          verificationreport:reportdetail:allDetails" />
    </sequence>
  </complexType>
</element>
```

192

It contains the following elements:

193  
194

<IncludeVerifier> [Default]

195  
196

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

197

<IncludeCertificateValues> [Default]

198  
199

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

200

<IncludeRevocationValues> [Default]

201  
202

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

203

<ExpandBinaryValues> [Default]

204  
205  
206

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

207

<ReportDetailLevel> [Optional]

208

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

209  
210

- [urn:oasis:names:tc:dss:1.0:profiles:verificationreport:reportdetail:noDetails](#)  
For every signature only the final result of the verification is reported.

211

- [urn:oasis:names:tc:dss:1.0:profiles:verificationreport:reportdetail:noPathDetails](#)

212  
213  
214

Additionally to the final result also the details of the signature verification including the result of the certificate path validation are reported. The details concerning the validation of individual certificates in the path are omitted however.

- 215 – [urn:oasis:names:tc:dss:1.0:profiles:verificationreport:reportdetail:allDetails](#)  
216 For every signature, the certificate path details and details on the validation of individual  
217 certificates in the path are requested. For every signature, the certificate path and each individual  
218 certificate the details are reported. If the <ReportDetailLevel>-element is missing, this  
219 option is assumed as default.

## 220 3.2 Element <VerificationReport>

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

223

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

225

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

228

```
229 <complexType name="VerificationReportType">  
230 <sequence>  
231 <element ref="dss:VerificationTimeInfo" maxOccurs="1" minOccurs="0" />  
232 <element name="VerifierIdentity" type="vr:IdentifierType"  
233 maxOccurs="1" minOccurs="0" />  
234 <element name="IndividualReport" maxOccurs="unbounded"  
235 type="vr:IndividualReportType" minOccurs="0" />  
236 </sequence>  
237 </complexType>
```

238

239 It contains the following elements:

240 <VerificationTimeInfo> [Optional]

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

243 <VerifierIdentity> [Optional]

244 This element contains the identity of the verifier, if the report option <IncludeVerifier> was set to  
245 'true'. It is of type **vr:IdentifierType**, which is defined below.

246 <IndividualReport> [Optional, Unbounded]

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

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

252

```
253 <complexType name="IdentifierType">  
254 <sequence>  
255 <element ref="ds:X509Data" maxOccurs="1" minOccurs="0">  
256 </element>  
257 <element name="SAMLv1Identifier" type="saml:NameIdentifierType"  
258 maxOccurs="1" minOccurs="0" />  
259 <element name="SAMLv2Identifier" type="saml2:NameIDType"  
260 maxOccurs="1" minOccurs="0" />  
261 <element name="Other" type="dss:AnyType" maxOccurs="1"  
262 minOccurs="0" />  
263 </sequence>
```

264 `</complexType>`

265

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

267 `<ds:X509Data>` [Optional]

268 This element contains, if present, an X.509-certificate or certificate related information. Please refer to  
269 **[RFC3275]** for further details with respect to the `ds:X509Data`-element.

270 `<SAMLv1Identifier>` [Optional]

271 This element contains, if present, an identifier of type **saml:NameIdentifierType** as defined in  
272 **[SAMLCore1.1]**.

273 `<SAMLv2Identifier>` [Optional]

274 This element contains, if present, an identifier of type **saml2:NameIDType** as defined in  
275 **[SAMLCore2.0]**.

276 `<Other>` [Optional]

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

278

### 279 **3.3 Element <IndividualReport>**

280

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

283

```
284 <complexType name="IndividualReportType">  
285 <sequence>  
286 <element name="SignedObjectIdentifier"  
287 type="vr:SignedObjectIdentifierType"/>  
288 <element ref="dss:Result"/>  
289 <element name="Details" type="dss:AnyType" maxOccurs="1" minOccurs="0" />  
290 </sequence>  
291 </complexType>
```

292

293 It contains the following elements:

294 `<SignedObjectIdentifier>` [Required]

295 This element identifies the signature or validation data under consideration. The details of the  
296 `SignedObjectIdentifierType` are specified below.

297 `<Result>` [Required]

298 The result of the signature verification as defined in section 2.6 of **[DSSCore]**.

299 `<Details>` [Optional]

300 The `<Details>` element MAY contain a detailed report for the signature or validation data under  
301 consideration or any other signature-specific optional output defined in Section 4.5 of **[DSSCore]**.  
302 The corresponding elements, which are specified in this document for this purpose are listed in  
303 Section 2.5.2.

304

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

306

```
307 <complexType name="SignedObjectIdentifierType">  
308 <sequence>
```

```

309     <element name="DigestAlgAndValue"
310           type="XAdES:DigestAlgAndValueType" maxOccurs="1" minOccurs="0" />
311     <element ref="ds:CanonicalizationMethod" maxOccurs="1" minOccurs="0" />
312     <element name="SignedProperties"
313           type="vr:SignedPropertiesType" maxOccurs="1" minOccurs="0" />
314     <element ref="ds:SignatureValue" maxOccurs="1" minOccurs="0" />
315     <element name="Other" type="dss:AnyType" maxOccurs="1" minOccurs="0" />
316 </sequence>
317 <attribute name="WhichDocument" type="IDREF" use="optional" />
318 <attribute name="XPath" type="string" use="optional" />
319 <attribute name="Offset" type="integer" use="optional" />
320 <attribute name="FieldName" type="string" use="optional" />
321 </complexType>

```

322

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

325 It contains the following attributes and elements:

326 <DigestAlgAndValue> [Optional]

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

335 <ds:CanonicalizationMethod> [Optional]

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

339 <SignedProperties> [Optional]

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

344 <ds:SignatureValue> [Optional]

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

348 <Other> [Optional]

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

351 WhichDocument [Optional]

352 This attribute MAY specify the document which contains the signature under consideration. Note that  
353 this identifier is only unique with respect to a specific request message (see **[DSSCore]**, Section  
354 2.4.1).

355 XPath [Optional]

356 This attribute MAY be used to point to a specific signature within an XML document.

357 Offset [Optional]

358 This attribute specifies the first byte of some signature and MAY be used to point to a specific  
359 signature within some binary document.

360 **FieldName** [Optional]

361 This attribute specifies the name of a signature field and MAY be used to point to a specific signature  
362 within some document format, in which there are field names such as PDF for example.

### 363 **3.4 VerificationResultType**

364 The **VerificationResultType** defined below is extensively used in the present profile to indicate the  
365 success or failure of individual verification steps.

366 This type draws from the `dss:Result`-element and the **dss:DetailType** defined in **[DSSCore]** and is  
367 defined as follows:

```
368 <complexType name="VerificationResultType">  
369 <sequence>  
370 <element name="ResultMajor" type="anyURI"/>  
371 <element name="ResultMinor" type="anyURI" minOccurs="0"/>  
372 <element name="ResultMessage" type="dss:InternationalStringType"  
373 minOccurs="0"/>  
374 <any namespace="##other" processContents="lax" minOccurs="0"  
375 maxOccurs="unbounded"/>  
376 </sequence>  
377 </complexType>
```

378

379 **<ResultMajor>** [Required]

380 This element MUST indicate whether the verification result is valid, invalid or indetermined using the  
381 URIs defined in **[DSSCore]**:

- 382 • urn:oasis:names:tc:dss:1.0:detail:valid
- 383 • urn:oasis:names:tc:dss:1.0:detail:invalid
- 384 • urn:oasis:names:tc:dss:1.0:detail:indetermined

385 **<ResultMinor>** [Optional]

386 In case of an invalid or indetermined verification step, further details MAY be provided using a specific  
387 URI defined in this document or other profiles.

388 **<ResultMessage>** [Optional]

389 Especially in case of an invalid or indetermined verification step, further details MAY be provided in  
390 textual form.

391 Furthermore an element of type **VerificationResultType** MAY contain other elements.

### 392 **3.5 Element <DetailedSignatureReport>**

393 The **<DetailedSignatureReport>**-element MAY appear in the **<Details>**-element within the  
394 **<IndividualReport>**-element, which is specified in Section 3.3 above. This element is defined as  
395 follows:

```
396 <element name="DetailedSignatureReport" type="vr:DetailedSignatureReportType"  
397 />
```

398

399 The **DetailedSignatureReportType** in turn is specified as follows:

400

```
401 <complexType name="DetailedSignatureReportType">  
402 <sequence>
```

```

403 <element name="FormatOK" type="vr:VerificationResultType" />
404 <element name="Properties" type="vr:PropertiesType"
405     maxOccurs="1" minOccurs="0" />
406 <element ref="dss:VerifyManifestResults" maxOccurs="1"
407     minOccurs="0" />
408 <element name="SignatureHasVisibleContent" type="boolean"
409     maxOccurs="1" minOccurs="0" />
410 <element name="SignatureOK"
411     type="vr:SignatureValidityType" />
412 <element name="CertificatePathValidity"
413     type="vr:CertificatePathValidityType" />
414 </sequence>
415 </complexType>

```

416  
417 It contains the following elements:

418 <FormatOK> [Required]

419 This element indicates, whether the format of the signature is ok or not. More information on the use of  
420 the **VerificationResultType** may be found in Section 3.4.

421 <Properties> [Optional]

422 This element contains information gathered during the verification of signed or unsigned properties.  
423 The structure of the **PropertiesType** is defined in Section 3.5.4.

424 <VerifyManifestResults> [Optional]

425 This element is present, if a manifest verification has been performed. The structure and the  
426 semantics of this element is described in Section 4.5.1 of [DSSCore].

427 <SignatureHasVisibleContent> [Optional]

428 This element is only present if the FieldName-attribute (cf. Section 3.3) is present and indicates  
429 whether the signature under consideration has visual signature content as explained in [DSSVisSig].

430 <SignatureOK> [Required]

431 This element contains information about the mathematical validity of the digital signature under  
432 consideration. It is of type **SignatureValidityType**, which is specified in Section 3.5.1.

433 <CertificatePathValidity> [Required]

434 This element contains the results of the certificate path validation. The **CertificatePathValidityType** is  
435 defined in section 3.5.3.

### 436 3.5.1 SignatureValidityType

437 The **SignatureValidityType** is used in the definition of the <DetailedSignatureReport>-element  
438 above for example and it is specified as follows:

439

```

440 <complexType name="SignatureValidityType">
441 <sequence>
442 <element name="SigMathOK" type="vr:VerificationResultType" />
443 <element name="SignatureAlgorithm" type="vr:AlgorithmValidityType"
444     maxOccurs="1" minOccurs="0" />
445 </sequence>
446 </complexType>

```

447  
448 It comprises the following elements:

449 <SigMathOK> [Required]

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

452 <SignatureAlgorithm> [Optional]

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

455

### 456 3.5.2 AlgorithmValidityType

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

459

```
460 <complexType name="AlgorithmValidityType">  
461 <sequence>  
462 <element name="Algorithm" type="anyURI" />  
463 <element name="Parameters" type="dss:AnyType" maxOccurs="1" minOccurs="0" />  
464 <element name="Suitability" type="vr:VerificationResultType" maxOccurs="1"  
465 minOccurs="0" />  
466 </sequence>  
467 </complexType>
```

468

469 <Algorithm> [Required]

470 This element contains the URI for the algorithm.

471 <Parameters> [Optional]

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

473 <Suitability> [Optional]

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

### 479 3.5.3 CertificatePathValidityType

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

- 482 • **DetailedSignatureReportType** (see above),
- 483 • **AttributeCertificateValidityType** (see Section 3.5.4.3),
- 484 • **CRLValidityType** (see Section 3.5.3.4),
- 485 • **OCSPValidityType** (see Section 3.5.3.5) and
- 486 • **TimeStampValidityType** (see Section 3.5.4.4).

487

488 It is specified as follows:

489

```
490 <complexType name="CertificatePathValidityType">  
491 <sequence>  
492 <element name="PathValiditySummary" type="vr:VerificationResultType" />  
493 <element name="CertificateIdentifier" type="ds:X509IssuerSerialType" />  
494 <element name="PathValidityDetail"  
495 type="vr:CertificatePathValidityDetailType"  
496 minOccurs="0" maxOccurs="1" />  
497 </sequence>
```



498 `</complexType>`

499

500 It contains the following elements:

501 `<PathValiditySummary>` [Required]

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

504 `<CertificateIdentifier>` [Required]

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

507 `<PathValidityDetail>` [Optional]

508 Contains detailed results of the certificate path validation, if the element `<ReportDetailLevel>` in  
509 the report options (see Section 3.1) was set to [urn:oasis:names:tc:dss:1.0:  
510 profiles:verificationreport:reportdetail:allDetails](urn:oasis:names:tc:dss:1.0:profiles:verificationreport:reportdetail:allDetails) and the detailed validity information has not been  
511 included elsewhere in the verification report.

512

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

514

```
515 <complexType name="CertificatePathValidityDetailType">  
516   <sequence>  
517     <sequence maxOccurs="unbounded" minOccurs="0">  
518       <element name="CertificateValidity" type="vr:CertificateValidityType" />  
519     </sequence>  
520     <element name="TSLValidity"  
521       type="vr:TrustStatusListValidityType" maxOccurs="1" minOccurs="0" />  
522     <element name="TrustAnchor" type="vr:VerificationResultType" />  
523   </sequence>  
524 </complexType>
```

525

526 It contains the following elements:

527 `<CertificateValidity>` [Optional, Unbounded]

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

531 `<TSLValidity>` [Optional]

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

534 `<TrustAnchor>` [Required]

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

- 537 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:trustanchor:SSCD> – indicates that the  
538 trusted root certificate is stored within a secure signature creation device according to  
539 [EC/1999/93].
- 540 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:trustanchor:otherCard> – indicates that the  
541 trusted root certificate is stored within some other hardware token.
- 542 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:trustanchor:certDataBase> – indicates that  
543 the trusted root certificate is stored within some certificate data base.
- 544 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:trustanchor:other> – indicates that the  
545 trusted root certificate is stored using other means.

546

### 547 3.5.3.1 CertificateValidityType

548

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

551

```
552 <complexType name="CertificateValidityType">  
553 <sequence>  
554 <element name="CertificateIdentifier" type="ds:X509IssuerSerialType" />  
555 <element name="Subject" type="string" />  
556 <element name="ChainingOK" type="vr:VerificationResultType"  
557 maxOccurs="1" minOccurs="0"/>  
558 <element name="ValidityPeriodOK" type="vr:VerificationResultType" />  
559 <element name="ExtensionsOK" type="vr:VerificationResultType" />  
560 <element name="CertificateValue" type="base64Binary"  
561 maxOccurs="1" minOccurs="0" />  
562 <element name="CertificateContent"  
563 type="vr:CertificateContentType" maxOccurs="1" minOccurs="0" />  
564 <element name="SignatureOK"  
565 type="vr:SignatureValidityType" />  
566 <element name="CertificateStatus" type="vr:CertificateStatusType" />  
567 </sequence>  
568 </complexType>
```

569

570 It contains the following elements:

571 <CertificateIdentifier> [Required]

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

574 <Subject> [Required]

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

578 <ChainingOK> [Optional]

579 If present, this element indicates whether the chaining to a previous certificate in the certificate path is  
580 ok or not. If the certificate under consideration is the first certificate in the certificate path, this element  
581 SHOULD be omitted. More information on the use of the **VerificationResultType** may be found in  
582 Section 3.4.

583 <ValidityPeriodOK> [Required]

584 This element indicates, whether the reference point in time is within the validity period of the  
585 certificate. More information on the use of the **VerificationResultType** may be found in Section 3.4.

586 <ExtensionsOK> [Required]

587 This element indicates, whether the certificate extensions are correct. More information on the use of  
588 the **VerificationResultType** may be found in Section 3.4.

589 <CertificateValue> [Optional]

590 If present, this element contains the certificate in binary form (coded in ASN.1), if the report option  
591 <IncludeCertificateValues> is set to 'true' and if the certificate is not already included in the  
592 verification report.

593 <CertificateContent> [Optional]

594 If present, this element contains detailed information about the content of the certificate, if the report  
595 option <ExpandBinaryValues> is set to 'true' and if the certificate content is not already included in  
596 the verification report.

597 <SignatureOK> [Required]

598 This element indicates, whether the digital signature of the certificate is mathematically correct or not.  
599 The **SignatureValidityType** is defined in section 3.5.1.

600 <CertificateStatus> [Required]

601 This element contains information about the result of the certificate revocation check. The  
602 **CertificateStatusType** is defined in Section 3.5.3.3.

603

### 604 3.5.3.2 CertificateContentType

605

606 The **CertificateContentType** is used in **CertificateValidityType** and derived from the  
607 TBSCertificate-structure defined in [RFC5280] specified as follows:

608

```
609 <complexType name="CertificateContentType">  
610   <sequence>  
611     <element name="Version" type="integer" maxOccurs="1" minOccurs="0" />  
612     <element name="SerialNumber" type="integer" />  
613     <element name="SignatureAlgorithm" type="anyURI" />  
614     <element name="Issuer" type="string" />  
615     <element name="ValidityPeriod" type="vr:ValidityPeriodType" />  
616     <element name="Subject" type="string" />  
617     <element name="Extensions" type="vr:ExtensionsType" minOccurs="0" />  
618   </sequence>  
619 </complexType>
```

620

621 It contains the following elements:

622 <Version> [Optional]

623 This element contains, if present, the version of the certificate structure.

624 <SerialNumber> [Required]

625 This element MUST contain the serial number of the certificate.

626 <SignatureAlgorithm> [Required]

627 This element MUST contain an identifier of the used signature algorithm. The  
628 vr:VerificationResultType is defined in Section 3.4.

629 <Issuer> [Required]

630 This element MUST contain the issuer of the certificate, where different relative distinguished names  
631 in a sequence MAY be separated by ":".

632 <ValidityPeriod> [Required]

633 This element MUST contain the validity period of the certificate. The **ValidityPeriodType** is defined  
634 below.

635 <Subject> [Required]

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

639

640 <Extensions> [Optional]

641 If present, this element contains information about the list of extensions present in the certificate under  
642 consideration. The **ExtensionsType** is defined below.

643

644 The **ValidityPeriodType** is specified as follows:

645

```
646 <complexType name="ValidityPeriodType">  
647   <sequence>  
648     <element name="NotBefore" type="dateTime" />  
649     <element name="NotAfter" type="dateTime" />  
650   </sequence>  
651 </complexType>
```

652

653 It contains the following elements:

654 <NotBefore> [Required]

655 The certificate is not valid before this point in time.

656 <NotAfter> [Required]

657 The certificate is not valid after this point in time.

658

659 The **ExtensionsType** is specified as follows:

660

```
661 <complexType name="ExtensionsType">  
662   <sequence minOccurs="0" maxOccurs="unbounded">  
663     <element name="Extension" type="vr:ExtensionType" />  
664   </sequence>  
665 </complexType>
```

666

667 It contains an unbounded number <Extension>-elements of type **ExtensionType**. This type is defined  
668 as follows:

669

```
670 <complexType name="ExtensionType">  
671   <sequence>  
672     <element name="ExtnId" type="XAdES:ObjectIdentifierType" />  
673     <element name="Critical" type="boolean" />  
674     <element name="ExtnValue" type="dss:AnyType" maxOccurs="1" minOccurs="0"  
675   />  
676     <element name="ExtensionOK" type="vr:VerificationResultType" />  
677   </sequence>  
678 </complexType>
```

679

680 It contains the following elements:

681 <ExtnId> [Required]

682 This element MUST contain the identifier of the extension as urn:oid: ... in the <Identifier>-  
683 element and MAY contain further information in the <Description>- and  
684 <DocumentationReferences>-elements. Please refer to **[XAdES]** for more information on the  
685 **XAdES:ObjectIdentifierType**.

686 <Critical> [Required]

687 This element specifies, whether the extension is critical or not.

688  
689 <ExtnValue> [Optional]  
690 This element SHOULD contain the value of the extension as an XML-structure, which mirrors the  
691 original ASN.1-definition of the extension.  
692 <ExtensionOK> [Required]  
693 This element contains information about the validity of the specific extension within the given context  
694 of the certificate.  
695

### 696 3.5.3.3 CertificateStatusType

697  
698 The **CertificateStatusType** is defined as follows:  
699

```
700 <complexType name="CertificateStatusType">  
701   <sequence>  
702     <element name="CertStatusOK" type="vr:VerificationResultType" />  
703     <element name="RevocationInfo" maxOccurs="1"  
704       minOccurs="0">  
705       <complexType>  
706         <sequence>  
707           <element name="RevocationDate" type="dateTime" />  
708           <element name="RevocationReason"  
709             type="vr:VerificationResultType" />  
710         </sequence>  
711       </complexType>  
712     </element>  
713     <element name="RevocationEvidence" maxOccurs="1"  
714       minOccurs="0">  
715       <complexType>  
716         <choice>  
717           <element name="CRLValidity"  
718             type="vr:CRLValidityType" />  
719           <element name="CRLReference"  
720             type="XAdES:CRLIdentifierType" />  
721           <element name="OCSPValidity"  
722             type="vr:OCSPValidityType" />  
723           <element name="OCSPReference"  
724             type="XAdES:OCSPIdentifierType" />  
725           <element name="Other" type="dss:AnyType"/>  
726         </choice>  
727       </complexType>  
728     </element>  
729   </sequence>  
730 </complexType>
```

731  
732 It contains the following elements:

733 <CertStatusOK> [Required]  
734 This element MUST contain the status of the certificate.  
735 <RevocationInfo> [Optional]  
736 If the certificate is revoked this element will contain more information about the revocation. It is defined  
737 to be a sequence, which contains the following elements:  
738 • <RevocationDate>  
739 contains the date and time of revocation.

- 740 • `<RevocationReason>`  
741 contains the reason for revocation. Following the definition of `CRLReason` in [RFC5280] there are  
742 the following URIs to specify the revocation reason:
- 743 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:revocationreason:unspecified>
  - 744 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:revocationreason:keyCompromise>
  - 745 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:revocationreason:cACompromise>
  - 746 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:revocationreason:affiliationChanged>
  - 747 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:revocationreason:superseded>
  - 748 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:revocationreason:cessationOfOperation>
  - 749 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:revocationreason:certificateHold>
  - 750 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:revocationreason:removeFromCRL>
  - 751 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:revocationreason:privilegeWithdrawn>
  - 752 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:revocationreason:aACompromise>

753 `<RevocationEvidence>` [Optional, Choice]

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

- 756 • `<CRLValidity>`  
757 This element contains information about the used CRL and its validity. The **CRLValidityType** is  
758 defined in Section 3.5.3.4.
- 759 • `<CRLReference>`  
760 This element contains a reference to the CRL in case it is already included elsewhere in the  
761 present verification report. The **XAdES:CRLIdentifierType** is defined in [XAdES].
- 762 • `<OCSPValidity>`  
763 This element contains information about the used OCSP response and its validity. The  
764 **OCSPValidityType** is defined in Section 3.5.3.5.
- 765 • `<OCSPReference>`  
766 This element contains a reference to the used OCSP response, if it is already included elsewhere  
767 in the present verification report. The **XAdES:OCSPIdentifierType** is defined in [XAdES].
- 768 • `<Other>`  
769 This element MAY contain information about alternative sources of revocation information.

### 770 3.5.3.4 CRLValidityType

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

772

```
773 <complexType name="CRLValidityType">  
774   <sequence>  
775     <element name="CRLIdentifier" type="XAdES:CRLIdentifierType"  
776       maxOccurs="1" minOccurs="1" />  
777     <element name="CRLValue" type="base64Binary"  
778       maxOccurs="1" minOccurs="0" />  
779     <element name="CRLContent" type="vr:CRLContentType"  
780       maxOccurs="1" minOccurs="0" />  
781     <element name="SignatureOK" type="vr:SignatureValidityType" />  
782     <element name="CertificatePathValidity"  
783       type="vr:CertificatePathValidityType" />  
784   </sequence>  
785   <attribute name="Id" type="ID" use="optional" />  
786 </complexType>
```

787

788 It contains the following attributes and elements:

789 Id [Optional]

790 This attribute contains an optional identifier for the element.

791 <CRLIdentifier> [Required]

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

793 <CRLValue> [Optional]

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

796 <CRLContent> [Optional]

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

799 <SignatureOK> [Required]

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

802 <CertificatePathValidity> [Required]

803 This element contains the result of the validation of the certificate path of the certificate which has  
804 been used to sign the CRL. The **CertificatePathValidityType** is defined at the beginning of Section  
805 3.5.3.

806

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

808

```
809 <complexType name="CRLContentType">
810   <sequence>
811     <element name="Version" minOccurs="0" type="integer" />
812     <element name="Signature" type="anyURI" />
813     <element name="Issuer" type="string" />
814     <element name="ThisUpdate" type="dateTime" />
815     <element name="NextUpdate" minOccurs="0" type="dateTime" />
816     <element name="RevokedCertificates" minOccurs="0">
817       <complexType>
818         <sequence minOccurs="0" maxOccurs="unbounded">
819           <element name="UserCertificate" type="integer" />
820           <element name="RevocationDate" type="dateTime" />
821           <element name="CrlEntryExtensions" minOccurs="0"
822             type="vr:ExtensionsType" />
823         </sequence>
824       </complexType>
825     </element>
826     <element name="CrlExtensions" type="vr:ExtensionsType" minOccurs="0" />
827   </sequence>
828 </complexType>
```

829

830 It contains the following elements:

831 <Version> [Optional]

832 This element contains, if present, the version of the CRL-structure.

833 <Signature> [Required]

834 This element contains the algorithm identifier for the algorithm used to sign the CRL.

835 <Issuer> [Required]

836 This element contains the issuer of the CRL, where different relative distinguished names in a  
837 sequence MAY be separated by “.”.

838 <ThisUpdate> [Required]  
839 This element contains the issue date of the CRL.  
840 <NextUpdate> [Optional]  
841 This element contains, if present, the date by which the next CRL will be issued.  
842 <RevokedCertificates> [Optional]  
843 The revoked certificates are contained in an unbounded sequence. They are listed by their serial  
844 numbers (element <UserCertificate>). Certificates revoked by the CA are uniquely identified by  
845 their certificate serial number. The date on which the revocation occurred is contained in the element  
846 <RevocationDate>. Additional information MAY be supplied in the element  
847 <CrlEntryExtensions>.  
848 <CrlExtensions> [Optional]  
849 If present, this element contains information about the list of extensions present in the CRL under  
850 consideration. The **ExtensionType** is defined in Section 3.5.3.2.

### 851 3.5.3.5 OCSPValidityType

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

854

```
855 <complexType name="OCSPValidityType">  
856 <sequence>  
857 <element name="OCSPIdentifier" type="XAdES:OCSPIdentifierType" />  
858 <element name="OCSPValue" type="base64Binary"  
859 maxOccurs="1" minOccurs="0" />  
860 <element name="OCSPContent" type="vr:OCSPContentType"  
861 maxOccurs="1" minOccurs="0" />  
862 <element name="SignatureOK" type="vr:SignatureValidityType" />  
863 <element name="CertificatePathValidity"  
864 type="vr:CertificatePathValidityType" />  
865 </sequence>  
866 <attribute name="Id" type="ID" use="optional" />  
867 </complexType>
```

868

869 It contains the following attributes and elements:

870 Id [Optional]

871 This attribute contains an optional identifier for the element.

872 <OCSPIdentifier> [Required]

873 This element refers to an OCSP response according to **[RFC2560]**.

874 <OCSPValue> [Optional]

875 This element contains the OCSP response (encoded in ASN.1) if the report option  
876 <IncludeRevocationValues> has been set to 'true'.

877 <OCSPContent> [Optional]

878 This element contains the OCSP response in form of an equivalent XML structure if the report option  
879 <ExpandBinaryValues> has been set to 'true'. The **OCSPContentType** is defined below.

880 <SignatureOK> [Required]

881 This element indicates whether the digital signature of the OCSP-response is mathematically correct  
882 or not. The **SignatureValidityType** is defined in section 3.5.1.

883

884



885 <CertificatePathValidity> [Required]

886 This element contains the result of the validation of the certificate path of the certificate which has  
887 been used to sign the OCSP-response. The **CertificatePathValidityType** is defined at the beginning  
888 of Section 3.5.3.

889

890 The **OCSPContentType** is aligned to [RFC2560] specified as follows:

891

```
892 <complexType name="OCSPContentType">  
893   <sequence>  
894     <element name="Version" type="integer" />  
895     <element name="ResponderID" type="string" />  
896     <element name="producedAt" type="dateTime" />  
897     <element name="Responses">  
898       <complexType>  
899         <sequence maxOccurs="unbounded" minOccurs="0">  
900           <element name="SingleResponse" type="vr:SingleResponseType" />  
901         </sequence>  
902       </complexType>  
903     </element>  
904     <element name="ResponseExtensions" type="vr:ExtensionsType"  
905       maxOccurs="1" minOccurs="0" />  
906   </sequence>  
907 </complexType>
```

908

909 It contains the following elements:

910 <Version> [Required]

911 This element contains the version of the OCSP-response syntax.

912 <ResponderID> [Required]

913 This element contains the name of the OCSP-responder.

914 <producedAt> [Required]

915 This element contains the time at which the OCSP-responder produced the response.

916 <Responses> [Required]

917 This element contains an unbounded sequence of <SingleResponse> entries. The  
918 **SingleResponseType** is defined below.

919 <ResponseExtensions> [Optional]

920 If present, this element contains information about the list of extensions present in the OCSP-response  
921 under consideration. The **ExtensionsType** is defined in Section 3.5.3.2.

922

923 The **SingleResponseType** is specified as follows:

924

```
925 <complexType name="SingleResponseType">  
926   <sequence>  
927     <element name="CertID">  
928       <complexType>  
929         <sequence>  
930           <element name="HashAlgorithm" type="anyURI" />  
931           <element name="IssuerNameHash" type="hexBinary" />  
932           <element name="IssuerKeyHash" type="hexBinary" />  
933           <element name="SerialNumber" type="integer" />  
934         </sequence>  
935       </complexType>
```

```

936     </element>
937     <element name="CertStatus" type="vr:VerificationResultType" />
938     <element name="ThisUpdate" type="dateTime" />
939     <element name="NextUpdate" type="dateTime" maxOccurs="1" minOccurs="0" />
940     <element name="SingleExtensions" type="vr:ExtensionsType"
941           maxOccurs="1" minOccurs="0" />
942   </sequence>
943 </complexType>

```

944

945 It contains the following elements:

946 <CertID> [Required]

947 This element contains a sequence of elements, which uniquely identify the certificate (cf. [RFC2560],  
 948 Section 4.1.1).

949 <CertStatus> [Required]

950 This element contains information about the status of the certificate according to [RFC2560] using the  
 951 following URI in the ResultMajor-element:

- 952 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:certstatus:good>
- 953 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:certstatus:revoked>
- 954 • <urn:oasis:names:tc:dss:1.0:profiles:verificationreport:certstatus:unknown>

955 If the certificate is revoked and the revocation reason is present, this information MUST be included in  
 956 the ResultMinor-element as a URI defined in Section 3.5.3.4. In a similar fashion the revocation  
 957 time MUST be indicated in the ResultMessage-element.

958 <ThisUpdate> [Required]

959 This element contains the time at which the status being indicated is known to be correct (cf.  
 960 [RFC2560], Section 2.4).

961 <NextUpdate> [Optional]

962 This element contains, if present, the time until more recent information about the status of the  
 963 certificate will be available (cf. [RFC2560], Section 2.4).

964 <SingleExtensions> [Optional]

965 If present, this element contains information about the list of extensions present in the  
 966 SingleResponse-element. The **ExtensionType** is defined in Section 3.5.3.2.

967

### 968 3.5.3.6 TrustStatusListValidityType

969

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

971

```

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

```

981

982 It contains the following attributes and elements:

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

### 997 3.5.4 PropertiesType

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

1000

```
1001 <complexType name="PropertiesType">  
1002   <sequence>  
1003     <element name="SignedProperties"  
1004       type="vr:SignedPropertiesType" minOccurs="0" />  
1005     <element name="UnsignedProperties"  
1006       type="vr:UnsignedPropertiesType" minOccurs="0" />  
1007   </sequence>  
1008   <attribute name="Id" type="ID" use="optional" />  
1009 </complexType>
```

1010

1011 It contains the following attributes and elements:

1012 Id [Optional]

1013 This attribute contains, if present, an optional identifier for the element.

1014 <SignedProperties> [Optional]

1015 This element contains information gathered during the verification of signed properties. Details of the  
1016 SignedPropertiesType are specified in Section 3.5.4.1.

1017 <UnsignedProperties> [Optional]

1018 This element contains information gathered during the verification of unsigned properties. Details of  
1019 the UnsignedPropertiesType are specified in Section 3.5.4.2.

#### 1020 3.5.4.1 Signed Properties

1021 The **SignedPropertiesType** is aligned to [XAdES] structured as follows:

1022

```
1023 <complexType name="SignedPropertiesType">  
1024   <sequence>  
1025     <element name="SignedSignatureProperties"  
1026       type="vr:SignedSignaturePropertiesType" maxOccurs="1" minOccurs="0" />  
1027     <element name="SignedDataObjectProperties"  
1028       type="vr:SignedDataObjectPropertiesType" minOccurs="0" />
```

```
1029     <element name="Other" type="dss:AnyType" maxOccurs="1" minOccurs="0" />
1030 </sequence>
1031 <attribute name="Id" type="ID" use="optional" />
1032 </complexType>
```

1033  
1034 It contains the following attributes and elements:

1035 Id [Optional]

1036 This attribute contains an optional identifier for the element.

1037 <SignedSignatureProperties> [Optional]

1038 This element contains information gathered during the verification of signed properties related to the  
1039 signature itself. The **SignedSignaturePropertiesType** is defined in Section 3.5.4.1.1.

1040 <SignedDataObjectProperties> [Optional]

1041 This element contains information gathered during the verification of signed properties related to the  
1042 signed data object. The **SignedDataObjectPropertiesType** is defined in Section 3.5.4.1.2.

1043 <Other> [Optional]

1044 This element contains, if present, information about other signed properties.

### 1045 3.5.4.1.1 SignedSignaturePropertiesType

1046 The **SignedSignaturePropertiesType** is aligned to [RFC3275] defined as follows:

1047

```
1048 <complexType name="SignedSignaturePropertiesType">
1049 <sequence>
1050 <element ref="XAdES:SigningTime" maxOccurs="1" minOccurs="0" />
1051 <element ref="XAdES:SigningCertificate" maxOccurs="1" minOccurs="0" />
1052 <element ref="XAdES:SignaturePolicyIdentifier" maxOccurs="1"
1053   minOccurs="0" />
1054 <choice maxOccurs="1" minOccurs="0">
1055 <element ref="XAdES:SignatureProductionPlace" />
1056 <element name="Location" type="string" />
1057 </choice>
1058 <element name="SignerRole" type="vr:SignerRoleType"
1059   minOccurs="0" />
1060 </sequence>
1061 </complexType>
```

1062  
1063 It MAY contain the following elements:

1064 <XAdES:SigningTime> [Optional]

1065 This element contains, if present, the signing time (see Section 5.2.1 of [XAdES]).

1066 <XAdES:SigningCertificate> [Optional]

1067 This element contains, if present, a reference to the certificate upon which the signature is based (see  
1068 Section 5.2.2 of [XAdES]).

1069 <XAdES:SignaturePolicyIdentifier> [Optional]

1070 This element references, if present, the policy under which the signature was produced (see Section  
1071 5.2.3 of [XAdES]).

1072 <XAdES:SignatureProductionPlace> [Optional, Choice]

1073 This element contains, if present, information about the place where the signature was generated (see  
1074 Section 5.2.7 of [XAdES]). This element SHOULD be used in case of a XAdES- or CAdES-based  
1075 signature.

1076 <Location> [Optional, Choice]

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

1079 <SignerRole> [Optional]

1080 This element contains, if present, information about the role of the signer (see Section 5.2.8 of  
1081 [XAdES]).

1082

1083 The **SignerRoleType** is specified as follows:

1084

```
1085 <complexType name="SignerRoleType">  
1086 <sequence>  
1087 <element name="ClaimedRoles"  
1088 type="XAdES:ClaimedRolesListType" minOccurs="0" />  
1089 <element name="CertifiedRoles"  
1090 type="vr:CertifiedRolesListType" minOccurs="0" />  
1091 </sequence>  
1092 </complexType>
```

1093

1094 It MAY contain the following elements:

1095 <ClaimedRoles> [Optional]

1096 This element contains information about the claimed roles of the signer. The information is directly  
1097 extracted from the signature.

1098 <CertifiedRoles> [Optional]

1099 This element contains information gathered during the verification of attribute certificates.

1100

1101 The **CertifiedRolesListType** is specified as follows:

1102

```
1103 <complexType name="CertifiedRolesListType">  
1104 <sequence>  
1105 <element name="AttributeCertificateValidity"  
1106 type="vr:AttributeCertificateValidityType" maxOccurs="unbounded" />  
1107 </sequence>  
1108 </complexType>
```

1109

1110 It contains at least one <AttributeCertificateValidity>-element, which contains information  
1111 about the content and validity of an attribute certificate according to [RFC3281]. The  
1112 **AttributeCertificateValidityType** is defined in Section 3.5.4.3.

### 1113 3.5.4.1.2 SignedDataObjectPropertiesType

1114 The **SignedDataObjectPropertiesType** is defined as follows:

1115

```
1116 <complexType name="SignedDataObjectPropertiesType">  
1117 <sequence>  
1118 <element ref="XAdES:DataObjectFormat" maxOccurs="unbounded"  
1119 minOccurs="0" />  
1120 <choice maxOccurs="1" minOccurs="0">  
1121 <element ref="XAdES:CommitmentTypeIndication"  
1122 maxOccurs="unbounded" minOccurs="1"/>  
1123 <element name="Reason" type="string" />  
1124 </choice>
```

```

1125     <element name="AllDataObjectsTimeStamp"
1126         type="vr:TimeStampValidityType" minOccurs="0" maxOccurs="unbounded" />
1127     <element name="IndividualDataObjectsTimeStamp"
1128         type="vr:TimeStampValidityType" minOccurs="0" maxOccurs="unbounded" />
1129 </sequence>
1130 <attribute name="Id" type="ID" use="optional" />
1131 </complexType>

```

1132

1133 It contains the following attributes and elements:

1134 Id [Optional]

1135 This attribute contains an optional identifier for the element.

1136 <XAdES:DataObjectFormat> [Optional, Unbounded]

1137 This element contains information about the format of the signed data object (see Section 5.2.5 of  
1138 **[XAdES]**). This information is simply extracted from the signature.

1139 <XAdES:CommitmentTypeIndication> [Choice, Unbounded]

1140 This element contains, if present, an indication of the type of commitment implied by the signature  
1141 (see Section 5.2.6 of **[XAdES]**). This element SHOULD be used in case of a XAdES- or CAdES-based  
1142 signature.

1143 <Reason> [Choice]

1144 This element contains, if present, a description of the reason of the signature generation. This element  
1145 is only relevant in case of a PDF-based signature identified by a `FieldName`-attribute (cf. Section  
1146 3.3).

1147 <AllDataObjectsTimeStamp> [Optional, Unbounded]

1148 This element contains, if present, verification results for time stamps covering all data objects (see  
1149 Section 5.2.6 of **[XAdES]**). The **TimeStampValidityType** is described in Section 3.5.4.4.

1150 <IndividualDataObjectsTimeStamp> [Optional, Unbounded]

1151 This element contains, if present, verification results for time stamps covering only certain data objects  
1152 (see Section 5.2.10 of **[XAdES]**). The **TimeStampValidityType** is described in section 3.5.4.4.

### 1153 3.5.4.2 Unsigned Properties

1154 The **UnsignedPropertiesType** is specified as follows:

1155

```

1156 <complexType name="UnsignedPropertiesType">
1157 <sequence>
1158 <element name="UnsignedSignatureProperties"
1159     type="vr:UnsignedSignaturePropertiesType" minOccurs="0" />
1160 <element ref="XAdES:UnsignedDataObjectProperties"
1161     minOccurs="1" maxOccurs="1" />
1162 <element name="Other" type="dss:AnyType" maxOccurs="1"
1163     minOccurs="0">
1164 </element>
1165 </sequence>
1166 <attribute name="Id" type="ID" use="optional" />
1167 </complexType>

```

1168

1169 It contains the following attributes and elements:

1170 Id [Optional]

1171 This attribute contains an optional identifier for the element.

1172 <UnsignedSignatureProperties> [Optional]

1173 This element contains information gathered during the verification of the unsigned properties related to  
1174 the signature itself. The **UnsignedSignaturePropertiesType** is defined below.

1175 <XAdES:UnsignedDataObjectProperties> [Optional]

1176 This element contains unsigned properties referring to the signed data objects. These properties are  
1177 directly extracted from the signature.

1178 <Other> [Optional]

1179 This element MAY contain information about other unsigned properties.

1180

1181 The **UnsignedSignaturePropertiesType** is defined as follows:

1182

```
1183 <complexType name="UnsignedSignaturePropertiesType">  
1184   <choice maxOccurs="unbounded">  
1185     <element name="CounterSignature" type="vr:SignatureValidityType" />  
1186     <element name="SignatureTimeStamp" type="vr:TimeStampValidityType" />  
1187     <element ref="XAdES:CompleteCertificateRefs" />  
1188     <element ref="XAdES:CompleteRevocationRefs" />  
1189     <element ref="XAdES:AttributeCertificateRefs" />  
1190     <element ref="XAdES:AttributeRevocationRefs" />  
1191     <element name="SigAndRefsTimeStamp" type="vr:TimeStampValidityType" />  
1192     <element name="RefsOnlyTimeStamp" type="vr:TimeStampValidityType" />  
1193     <element name="CertificateValues" type="vr:CertificateValuesType" />  
1194     <element name="RevocationValues" type="vr:RevocationValuesType" />  
1195     <element name="AttrAuthoritiesCertValues"  
1196       type="vr:CertificateValuesType" />  
1197     <element name="AttributeRevocationValues"  
1198       type="vr:RevocationValuesType" />  
1199     <element name="ArchiveTimeStamp" type="vr:TimeStampValidityType" />  
1200     <any namespace="##other" />  
1201   </choice>  
1202   <attribute name="Id" type="ID" use="optional" />  
1203 </complexType>
```

1204

1205 It contains the following attributes and elements:

1206 Id [Optional]

1207 This attribute contains an optional identifier for the element.

1208 <CounterSignature> [Choice]

1209 This element contains the results of the verification of a counter signature (see Section 7.2.4 of  
1210 **[XAdES]**). The **SignatureValidityType** is described in section 3.5.1.

1211 <SignatureTimeStamp> [Choice]

1212 This element contains verification results of a time stamp of the signature (see Section 7.3 of  
1213 **[XAdES]**). The **TimeStampValidityType** is described in section 3.5.4.4.

1214 <XAdES:CompleteCertificateRefs> [Choice]

1215 This element contains references to the certificates used during verification of the signature (see  
1216 Section 7.4.1 of **[XAdES]**). This information is simply extracted from the signature.

1217 <XAdES:CompleteRevocationRefs> [Choice]

1218 Contains references to the revocation data used for the verification of the signature (see Section 7.4.2  
1219 of **[XAdES]**). This information is simply extracted from the signature.

1220 <XAdES:AttributeCertificateRefs> [Choice]

1221 Contains the references to the full set of attribute authorities certificates that have been used to  
1222 validate the attribute certificate (see section 7.4.3 of **[XAdES]**). This information is simply extracted  
1223 from the signature.

- 1224 <XAdES:AttributeRevocationRefs> [Choice]
- 1225     Contains the references to the full set of revocation data that have been used in the validation of the
- 1226     attribute certificate(s) present in the signature (see section 7.4.4 of [XAdES]).
- 1227 <SigAndRefsTimeStamp> [Choice]
- 1228     Contains verification results for a time stamp referring to the signature and references on certificates
- 1229     and revocation data (see section 7.5.1 of [XAdES]). The **TimeStampValidityType** is described in
- 1230     section 3.5.4.4.
- 1231 <RefsOnlyTimeStamp> [Choice]
- 1232     Contains verification results for a time stamp referring only to references on certificates and revocation
- 1233     data (see section 7.5.2 of [XAdES]). The **TimeStampValidityType** is described in section 3.5.4.4.
- 1234 <CertificateValues> [Choice]
- 1235     Contains verification results for the certificates, which were used in the verification of the signature
- 1236     (see section 7.6.1 of [XAdES]). The **CertificateValuesType** is defined below.
- 1237 <RevocationValues> [Choice]
- 1238     Contains verification results of the revocation data used in the verification of the signature (see section
- 1239     7.6.2 of [XAdES]). The **RevocationValuesType** is defined below.
- 1240 <AttrAuthoritiesCertValues> [Choice]
- 1241     Contains verification results of the certificates of Attribute Authorities that have been used to validate
- 1242     the attribute certificates, which are contained in the signature (see section 7.6.3 of [XAdES]). The
- 1243     **CertificateValuesType** is defined below.
- 1244 <AttributeRevocationValues> [Choice]
- 1245     Contains verification results of the revocation data that have been used to validate the attribute
- 1246     certificate when present in the signature (see section 7.6.4 of [XAdES]). The **RevocationValuesType**
- 1247     is defined below.
- 1248 <ArchiveTimeStamp> [Choice]
- 1249     Contains verification results for a time stamp covering the complete signature including all attributes
- 1250     (see section 7.7 of [XAdES]). The **TimeStampValidityType** is described in section 3.5.4.4.

1251

1252 The **CertificateValuesType** is defined as follows:

1253

```

1254 <complexType name="CertificateValuesType">
1255   <choice minOccurs="0" maxOccurs="unbounded">
1256     <element name="EncapsulatedX509Certificate"
1257       type="vr:CertificateValidityType" />
1258     <element name="OtherCertificate" />
1259   </choice>
1260   <attribute name="Id" type="ID" use="optional" />
1261 </complexType>

```

1262

1263 It defines the following attributes and elements:

1264 Id [Optional]

1265     This attribute contains an optional identifier for the element.

1266 <EncapsulatedX509Certificate> [Optional, Unbounded, Choice]

1267     Contains verification results for an X.509 certificate included in the signature. The

1268     **CertificateValidityType** is defined in Section 3.5.3.1.

1269 <OtherCertificate> [Optional, Unbounded, Choice]



1270 This element contains verification results for other certificates included in the signature. If a certificate  
1271 with unknown format is included in the signature, a warning (error code  
1272 [urn:oasis:names:tc:dss:1.0:resultminor:certificateFormatNotCorrectWarning](#)) SHOULD be returned.

1273

1274 The **RevocationValuesType** is defined as follows:

1275

```
1276 <complexType name="RevocationValuesType">  
1277   <sequence>  
1278     <element name="CRLValues" minOccurs="0">  
1279       <complexType>  
1280         <sequence maxOccurs="unbounded" minOccurs="1">  
1281           <element name="VerifiedCRL" type="vr:CRLValidityType" />  
1282         </sequence>  
1283       </complexType>  
1284     </element>  
1285     <element name="OCSPValues" minOccurs="0">  
1286       <complexType>  
1287         <sequence maxOccurs="unbounded" minOccurs="1">  
1288           <element name="VerifiedOCSPResponse" type="vr:OCSPValidityType" />  
1289         </sequence>  
1290       </complexType>  
1291     </element>  
1292     <element name="OtherValues" type="dss:AnyType" minOccurs="0" />  
1293   </sequence>  
1294   <attribute name="Id" type="ID" use="optional" />  
1295 </complexType>
```

1296

1297 It contains the following attributes and elements:

1298 Id [Optional]

1299 This attribute contains an optional identifier for the element.

1300 <CRLValues> [Optional]

1301 Contains the verification results for all CRLs included in a signature. The **CRLValidityType** is defined  
1302 in Section 3.5.3.4.

1303 <OCSPValues> [Optional]

1304 Contains the verification results for all OCSP responses included in a signature. The  
1305 **OCSPValidityType** is defined in Section 3.5.3.5.

1306 <OtherValues> [Optional]

1307 This element MAY contain verification results for other revocation data included in the signature. If  
1308 other revocation data with unknown format is included in the signature, a warning (error  
1309 [urn:oasis:names:tc:dss:1.0:resultminor:improperRevocationInformation](#)) SHOULD be returned.

1310

### 1311 3.5.4.3 AttributeCertificateValidityType

1312 The **AttributeCertificateValidityType** is defined as follows:

1313

```
1314 <complexType name="AttributeCertificateValidityType">  
1315   <sequence>  
1316     <element name="AttributeCertificateIdentifier"  
1317       type="vr:AttrCertIDType" maxOccurs="1" minOccurs="0" />  
1318     <element name="AttributeCertificateValue" type="base64Binary"  
1319       maxOccurs="1" minOccurs="0" />  
1320     <element name="AttributeCertificateContent"
```

```
1321     type="vr:AttributeCertificateContentType" maxOccurs="1" minOccurs="0" />
1322     <element name="SignatureOK" type="vr:SignatureValidityType" />
1323     <element name="CertificatePathValidity"
1324         type="vr:CertificatePathValidityType" />
1325     </sequence>
1326 </complexType>
```

1327  
1328 It contains the following elements:

1329 <AttributeCertificateIdentifier> [Optional]

1330 This element MAY refer to an X.509v3 attribute certificate according to [RFC3281]. The structure of  
1331 the **AttrCertIDType** is defined below.

1332 <AttributeCertificateValue> [Optional]

1333 This element MAY contain the certificate in binary form (coded in ASN.1), if the report option  
1334 <IncludeCertificateValues> is set to 'true'.

1335 <AttributeCertificateContent> [Optional]

1336 This element MAY contain an XML-based analogue of the content of the certificate, if the report option  
1337 <ExpandBinaryValues> is set to 'true'. The structure of the  
1338 **AttributeCertificateContentType** is defined below.

1339 <SignatureOK> [Required]

1340 This element indicates, whether the digital signature is mathematically valid or not. The  
1341 **SignatureValidityType** is defined in section 3.5.1.

1342 <CertificatePathValidity> [Required]

1343 This element contains the result of the validation of the certificate path of the certificate which has  
1344 been used to sign the attribute certificate. The **CertificatePathValidityType** is defined at the  
1345 beginning of Section 3.5.3.

1346

1347 The **AttrCertIDType** is structured as follows:

1348

```
1349 <complexType name="AttrCertIDType">
1350     <sequence>
1351         <element name="Holder" type="vr:EntityType" maxOccurs="1" minOccurs="0"/>
1352         <element name="Issuer" type="vr:EntityType" />
1353         <element name="SerialNumber" type="integer"></element>
1354     </sequence>
1355 </complexType>
```

1356  
1357 It contains the following elements:

1358 <Holder> [Optional]

1359 This element contains, if present, information about the holder of the certificate. The structure of the  
1360 **EntityType** is defined below.

1361 <Issuer> [Required]

1362 This element contains information about the issuer of the attribute certificate. The structure of the  
1363 **EntityType** is defined below.

1364 <SerialNumber> [Required]

1365 This element contains the serial number of the attribute certificate, which (together with the information  
1366 provided in the <Issuer>-element) uniquely identifies the attribute certificate.

1367

1368 The **EntityType** is aligned to the structure of `Holder` and `V2Form` in **[RFC3281]** and is defined as  
1369 follows:

1370

```
1371 <complexType name="EntityType">  
1372   <sequence>  
1373     <element name="BaseCertificateID"  
1374       type="ds:X509IssuerSerialType" maxOccurs="1" minOccurs="0" />  
1375     <element name="Name" type="string" maxOccurs="1" minOccurs="0" />  
1376     <element name="Other" type="dss:AnyType" maxOccurs="1"  
1377   minOccurs="0" /></element>  
1378   </sequence>  
1379 </complexType>
```

1380

1381 It SHOULD contain sufficient information to identify the entity uniquely and MAY contain the following  
1382 optional elements:

1383 <BaseCertificateID> [Optional]

1384 This element identifies, if present, the public-key certificate of the entity. The structure of the  
1385 `ds:X509IssuerSerialType` is defined in **[RFC3275]**.

1386 <Name> [Optional]

1387 This element contains, if present, the name of the entity.

1388 <Other> [Optional]

1389 This element MAY contain other information, which is used to identify the entity.

1390

1391 The **AttributeCertificateContentType** contains the content of an attribute certificate according to  
1392 **[RFC3281]** as XML structure and is structured as follows:

1393

```
1394 <complexType name="AttributeCertificateContentType">  
1395   <sequence>  
1396     <element name="Version" minOccurs="0" type="integer" />  
1397     <element name="Holder" type="vr:EntityType" />  
1398     <element name="Issuer" type="vr:EntityType" />  
1399     <element name="SignatureAlgorithm" type="anyURI" />  
1400     <element name="SerialNumber" type="integer" />  
1401     <element name="AttCertValidityPeriod"  
1402       type="vr:ValidityType" />  
1403     <element name="Attributes">  
1404       <complexType>  
1405         <sequence minOccurs="0" maxOccurs="unbounded">  
1406           <element name="Attribute"  
1407             type="vr:AttributeType" />  
1408         </sequence>  
1409       </complexType>  
1410     </element>  
1411     <element name="IssuerUniqueID" type="hexBinary" maxOccurs="1"  
1412   minOccurs="0" />  
1413     <element name="Extensions" minOccurs="0"  
1414       type="vr:ExtensionsType" />  
1415   </sequence>  
1416 </complexType>
```

1417

1418 It contains the following elements:

1419 <Version> [Optional]

1420 This element contains, if present, the version of the attribute certificate.

1421 <Holder> [Required]  
 1422 This element contains information about the holder of the certificate. The structure of the **EntityType**  
 1423 is defined above.  
 1424 <Issuer> [Required]  
 1425 This element contains the issuer of the attribute certificate. The structure of the **EntityType** is defined  
 1426 above.  
 1427 <SignatureAlgorithm> [Required]  
 1428 This element contains an identifier of the used signature algorithm.  
 1429 <SerialNumber> [Required]  
 1430 This element contains the serial number of the attribute certificate.  
 1431 <AttCertValidityPeriod> [Required]  
 1432 This element contains the validity period of the attribute certificate. The **ValidityType** is defined in  
 1433 section 3.5.3.2.  
 1434 <Attributes> [Optional, Unbounded]  
 1435 This element contains, if present, a list of attributes. The **AttributeType** is defined below.  
 1436 <IssuerUniqueID> [Optional]  
 1437 This element contains, if present, a unique identifier of the issuer of the attribute certificate.  
 1438 <Extensions> [Optional]  
 1439 If present, this element contains information about the list of extensions present in the attribute  
 1440 certificate. The **ExtensionType** is defined in Section 3.5.3.2.

1441  
 1442 The **AttributeType** is defined as follows:  
 1443

```

1444 <complexType name="AttributeType">
1445   <sequence>
1446     <element name="Type" type="anyURI" />
1447     <element name="Value" type="dss:AnyType" maxOccurs="unbounded"
1448 minOccurs="0"></element>
1449   </sequence>
1450 </complexType>
  
```

1451  
 1452 It contains the following elements:  
 1453 <Type> [Required]  
 1454 This element MUST contain an identifier for the type of the attribute in the <Code>-element and MAY  
 1455 contain further information.  
 1456 <Value> [Optional, Unbounded]  
 1457 This element MAY contain any number of attribute values.  
 1458

### 3.5.4.4 TimeStampValidityType

1459 The **TimeStampValidityType** is structured as follows:  
 1460  
 1461

```

1462 <complexType name="TimeStampValidityType">
1463   <sequence>
1464     <element name="FormatOK" type="vr:VerificationResultType" />
1465     <element name="TimeStampContent" type="vr:TstContentType"
  
```

```

1466     maxOccurs="1" minOccurs="0" />
1467     <element name="MessageHashAlgorithm" type="vr:AlgorithmValidityType"
1468     maxOccurs="1" minOccurs="0" />
1469     <element name="SignatureOK"
1470     type="vr:SignatureValidityType" />
1471     <element name="CertificatePathValidity"
1472     type="vr:CertificatePathValidityType" />
1473 </sequence>
1474 <attribute name="Id" type="ID" use="optional" />
1475 </complexType>

```

1476  
1477 It contains the following elements and attributes:

1478 Id [Optional]

1479 This attribute contains an optional identifier for the element.

1480 <FormatOK> [Required]

1481 This element indicates, whether the format of the time stamp is ok or not. More information on the use  
1482 of the **VerificationResultType** may be found in Section 3.4.

1483 <TimeStampContent> [Optional]

1484 This element contains the content of time stamp in form of an XML structure, if the report option  
1485 <ExpandBinaryValues> is set to 'true'. The **TstContentType** is specified below.

1486 <MessageHashAlgorithm> [Optional]

1487 This element contains, if present, information about the message hash algorithm and its suitability.  
1488 The **AlgorithmValidityType** is defined in Section 3.5.2.

1489 <SignatureOK> [Required]

1490 This element indicates, whether the digital signature is mathematically valid or not. The  
1491 **SignatureValidityType** is defined in Section 3.5.1.

1492 <CertificatePathValidity> [Required]

1493 This element contains the result of the validity check of the certificate. The  
1494 **CertificatePathValidityType** is defined in Section 3.5.3.

1495

1496 The **TstContentType** complex type is defined as follows:

1497

```

1498 <complexType name="TstContentType">
1499   <sequence>
1500     <element ref="dss:TstInfo" maxOccurs="1" minOccurs="0"/>
1501     <element name="Other" type="dss:AnyType" maxOccurs="1" minOccurs="0"/>
1502   </sequence>
1503 </complexType>

```

1504 It contains the following elements:

1505 <dss:TstInfo> [Optional]

1506 This element MAY contain the standard content of a time stamp as defined in Section 5.1.2 of  
1507 **[DSSCore]**. Note that there is a straightforward mapping from the TSTInfo-Element according to  
1508 **[RFC3161]** to the present structure.

1509 <Other> [Optional]

1510 This element MAY contain other information included in the time stamp.

### 1511 3.5.5 Element <IndividualTimeStampReport>

1512 The <IndividualTimeStampReport>-element MAY appear in the <Details>-element within the  
1513 <IndividualReport>-element defined in Section 3.3. This element is defined as follows:

```
1514 <element name="IndividualTimeStampReport" type="vr:TimeStampValidityType" />
```

1515 The **TimeStampValidityType** is defined in Section 3.5.4.4.

### 1516 3.5.6 Element <IndividualCertificateReport>

1517 The <IndividualCertificateReport>-element MAY appear in the <Details>-element within the  
1518 <IndividualReport>-element defined in Section 3.3. This element is defined as follows:

```
1519 <element name="IndividualCertificateReport"  
1520 type="vr:CertificateValidityType" />
```

1521 The **CertificateValidityType** is defined in Section 3.5.3.1.

### 1522 3.5.7 Element <IndividualAttributeCertificateReport>

1523 The <IndividualAttributeCertificateReport>-element MAY appear in the <Details>-  
1524 element within the <IndividualReport>-element defined in Section 3.3. This element is defined as  
1525 follows:

```
1526 <element name="IndividualAttributeCertificateReport"  
1527 type="vr:AttributeCertificateValidityType" />
```

1528 The **AttributeCertificateValidityType** is defined in Section 3.5.4.3.

### 1529 3.5.8 Element <IndividualCRLReport>

1530 The <IndividualCRLReport>-element MAY appear in the <Details>-element within the  
1531 <IndividualReport>-element defined in Section 3.3. This element is defined as follows:

```
1532 <element name="IndividualCRLReport" type="vr:CRLValidityType" />
```

1533 The **CRLValidityType** is defined in Section 3.5.3.4.

### 1534 3.5.9 Element <IndividualOCSPReport>

1535 The <IndividualOCSPReport>-element MAY appear in the <Details>-element within the  
1536 <IndividualReport>-element defined in Section 3.3. This element is defined as follows:

```
1537 <element name="IndividualOCSPReport" type="vr:OCSPValidityType" />
```

1538 The **OCSPValidityType** is defined in Section 3.5.3.5.

### 1539 3.5.10 Element <EvidenceRecordReport>

1540 The <EvidenceRecordReport>-element MAY appear in the <Details>-element within the  
1541 <IndividualReport>-element defined in Section 3.3. This element is defined as follows:

```
1542 <element name="EvidenceRecordReport" type="vr:EvidenceRecordValidityType" />
```

1543 The **EvidenceRecordValidityType** is based on the definition of the EvidenceRecord-element in  
1544 **[RFC4998]** defined as follows:

```
1545 <complexType name="EvidenceRecordValidityType">  
1546 <sequence>  
1547 <element name="FormatOK" type="vr:VerificationResultType" />
```

```

1548 <element name="Version" type="integer"
1549     maxOccurs="1" minOccurs="0">
1550 </element>
1551 <element name="DigestAlgorithm"
1552     type="vr:AlgorithmValidityType" maxOccurs="unbounded" minOccurs="0">
1553 </element>
1554 <element name="CryptoInfos" maxOccurs="1" minOccurs="0">
1555     <complexType>
1556         <sequence>
1557             <element name="Attribute"
1558                 type="vr:AttributeType" maxOccurs="unbounded" minOccurs="1">
1559             </element>
1560         </sequence>
1561     </complexType>
1562 </element>
1563 <element name="EncryptionInfo" maxOccurs="1" minOccurs="0">
1564     <complexType>
1565         <sequence>
1566             <element name="EncryptionInfoType"
1567                 type="vr:AlgorithmValidityType">
1568             </element>
1569             <element name="EncryptionInfoValue"
1570                 type="dss:AnyType">
1571             </element>
1572         </sequence>
1573     </complexType>
1574 </element>
1575 <element name="ArchiveTimeStampSequence" maxOccurs="1"
1576     minOccurs="1">
1577     <complexType>
1578         <sequence maxOccurs="unbounded" minOccurs="0">
1579             <element name="ArchiveTimeStampChain">
1580                 <complexType>
1581                     <sequence maxOccurs="unbounded"
1582                         minOccurs="0">
1583                         <element name="ArchiveTimeStamp"
1584                             type="vr:ArchiveTimeStampValidityType">
1585                         </element>
1586                     </sequence>
1587                 </complexType>
1588             </element>
1589         </sequence>
1590     </complexType>
1591 </element>
1592 </sequence>
1593 <attribute name="Id" type="ID" use="optional" />
1594 </complexType>

```

1595

1596 It contains the following elements and attributes:

1597 Id [Optional]

1598 This attribute contains an optional identifier for the element.

1599 <FormatOK> [Required]

1600 This element indicates, whether the format of the evidence record according to **[RFC4998]** is ok or  
1601 not. More information on the use of the **VerificationResultType** may be found in Section **Fehler!**  
1602 **Verweisquelle konnte nicht gefunden werden..**

1603 <Version> [Optional]

1604 This element contains, if present, the version of the Evidence Record Syntax.

1605 <DigestAlgorithm> [Optional, unbounded]

1606 This element appears for each hash algorithm used to produce the evidence record and contains  
1607 information about the hash algorithm and possibly its suitability. The **AlgorithmValidityType** is  
1608 defined in Section 3.5.2.

1609 <CryptoInfos> [Optional]

1610 This element MAY contain further data useful in the validation of the <ArchiveTimeStampSequence>-  
1611 element. As explained in [RFC4998] this MAY include possible Trust Anchors, certificates, revocation  
1612 information, or the information concerning the suitability of cryptographic algorithms.

1613 <EncryptionInfo> [Optional]

1614 This element MAY contain the necessary information to support encrypted content (cf. [RFC4998],  
1615 Section 6.1).

1616 <ArchiveTimeStampSequence> [Required]

1617 This element is required and MAY contain a sequence of <ArchiveTimeStampChain>-elements (cf.  
1618 [RFC4998], Section 5), which in turn MAY contain a sequence of <ArchiveTimeStamp>-elements,  
1619 which are of type **ArchiveTimeStampValidityType** defined below.

1620

1621 The **ArchiveTimeStampValidityType** is based on the definition of the ArchiveTimeStamp-element in  
1622 [RFC4998] defined as follows:

1623

```
1624 <complexType name="ArchiveTimeStampValidityType">  
1625   <sequence>  
1626     <element name="FormatOK" type="vr:VerificationResultType" />  
1627     <element name="DigestAlgorithm" type="vr:AlgorithmValidityType"  
1628       maxOccurs="1" minOccurs="0" />  
1629     <element name="Attributes" maxOccurs="1" minOccurs="0">  
1630       <complexType>  
1631         <sequence>  
1632           <element name="Attribute" type="vr:AttributeType"  
1633             maxOccurs="unbounded" minOccurs="1"/>  
1634         </sequence>  
1635       </complexType>  
1636     </element>  
1637     <element name="ReducedHashTree" maxOccurs="1" minOccurs="0">  
1638       <complexType>  
1639         <sequence maxOccurs="unbounded" minOccurs="1">  
1640           <element name="PartialHashTree">  
1641             <complexType>  
1642               <sequence maxOccurs="unbounded" minOccurs="1">  
1643                 <element name="HashValue"  
1644                   type="vr:HashValueType">  
1645                 </element>  
1646               </sequence>  
1647             </complexType>  
1648           </element>  
1649         </sequence>  
1650       </complexType>  
1651     </element>  
1652     <element name="TimeStamp"  
1653       type="vr:TimeStampValidityType" />  
1654   </sequence>  
1655   <attribute name="Id" type="ID" use="optional" />  
1656 </complexType>
```

1657

1658 It contains the following elements and attributes:

1659 Id [Optional]

1660 This attribute contains an optional identifier for the element.



1661 <FormatOK> [Required]  
1662 This element indicates, whether the format of the evidence record according to [RFC4998] is ok or  
1663 not. More information on the use of the **VerificationResultType** may be found in Section Fehler!  
1664 **Verweisquelle konnte nicht gefunden werden..**

1665 <DigestAlgorithm> [Optional]  
1666 This element contains, if present, information about the hash algorithm and possibly its suitability. The  
1667 **AlgorithmValidityType** is defined in Section 3.5.2.

1668 <Attributes> [Optional]  
1669 This element contains, if present, information about further attributes related to the archive time  
1670 stamp.

1671 <ReducedHashTree> [Optional]  
1672 This element MAY contain a sequence of <PartialHashTree>-elements, which in turn contain a  
1673 list of <HashValue>-elements of type **HashValueType** defined below.

1674 <TimeStamp> [Required]  
1675 This element is of type **TimeStampValidityType** (cf. Section 3.5.4.4) and contains information about  
1676 the validity of the conventional time stamp, which is included in the present archive time stamp.

1677  
1678 The **HashValueType** is used for the <HashValue>-element within the <PartialHashTree>-element  
1679 above and is defined as follows:

```
1680 <complexType name="HashValueType">  
1681   <sequence>  
1682     <element name="HashValue" type="hexBinary" />  
1683   </sequence>  
1684   <attribute name="HashedObject" type="IDREF" use="optional" />  
1685 </complexType>
```

1686 It contains the following elements and attributes:

1687 HashedObject [Optional]  
1688 This attribute MAY be used to point to the object, which served as pre-image of the hash value.

1689 <HashValue> [Required]  
1690 This element contains the hash value produced by applying the hash algorithm specified by the  
1691 <DigestAlgorithm>- or <TimeStamp>-element to the data specified by the HashedObject  
1692 attribute.

---

1693

## A. Acknowledgements

1694 The following individuals have participated in the creation of this specification and are gratefully  
1695 acknowledged:

1696 **Participants:**

- 1697 • Juan-Carlos Cruellas
  - 1698 • Andreas Kühne
  - 1699 • Ingo Henkel
  - 1700 • Ezer Farhi
  - 1701 • Stefan Drees
  - 1702 • Pim van der Eijk
  - 1703 • Clemens Orthacker
  - 1704 • Marta Cruellas
  - 1705 • Konrad Lanz
- 

1706

1707