OASIS DSS v1.0 Profile for Comprehensive Multi-Signature Verification Reports Version 1.0

Committee Specification 01

12 November 2010

Specification URIs:

This Version:

http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cs01.html
http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cs01.doc
http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cs01.pdf
(Authoritative)

Previous Version:

http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cd02.html
http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cd02.doc
http://docs.oasis-open.org/dss-x/profiles/verificationreport/oasis-dssx-1.0-profiles-vr-cd02.pdf
(Authoritative)

Latest Version:

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

Technical Committee:

OASIS Digital Signature Services eXtended (DSS-X) TC

Chair(s):

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

Editor(s):

Detlef Hühnlein, *Federal Office for Information Security, Germany* <detlef.huehnlein@ecsec.de>

Related work:

This specification is based on

- oasis-dss-core-spec-v1.0-os

and may be combined with other existing profiles, such as

- oasis-dss-profiles-AdES-v1.0-os
- oasis-dss-profiles-german_signature_law-spec-v1.0-os

for example.

Declared XML Namespace(s):

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

Abstract:

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

Status:

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

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

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

Notices

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

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

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

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS" and “DSS” 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

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 document are to be interpreted as described in **[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

[CAdES] ETSI: "*Electronic Signature Formats*", Electronic Signatures and Infrastructures (ESI) – Technical Specification, ETSI TS 101 733 V1.7.4, 2008-07

[Core-XSD] S. Drees, T. Perrin, J. C. Cruellas, N. Pope, K. Lanz, et al.: "*DSS Schema*", February 2007 http://docs.oasis-open.org/dss/v1.0/DSS-XML-SCHEMAS-v1.0-os/oasis-dss-core-schema-v1.0-os.xsd

[DSSCore] OASIS Standard, *Digital Signature Service Core Protocols and Elements,* April 2007 http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.pdf

[DSSAdES] OASIS Standard, *Advanced Electronic Signature Profiles of the OASIS Digital Signature Service Version 1.0*, April 2007 http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles-AdES-spec-v1.0-os.pdf

[DSSSigG] OASIS Standard, *German Signature Law Profile of the OASIS Digital Signature Service Version 1.0*, April 2007 http://docs.oasis-open.org/dss/v1.0/oasis-dss-profiles_german_signature_law-spec-v1.0-os.pdf

[DSSVR-XSD] D. Hühnlein, I. Henkel, J. C. Cruellas, S. Drees, A. Kuehne, et. al.: "*DSS Verification Report Schema*", July 2009 http://www.oasis-open.org/committees/download.php/33059/VerificationReport-CD1.xsd

[DSSVisSig] OASIS Committee Draft 01, *Visual Signature Profile of the OASIS Digital Signature Services*, 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)

[ETSI102231-2.1.1] ETSI: *"Provision of harmonized Trust-service status information",* Electronic Signatures and Infrastructure (ESI) – Technical Specification, ETSI TS 102231 Version 2.1.1 of March 2006

[ETSI102231-3.1.2] ETSI: *"Provision of harmonized Trust-service status information",* Electronic Signatures and Infrastructure (ESI) – Technical Specification, ETSI TS 102231, Version 3.1.2 of December 2009 (http://uri.etsi.org/02231/v3.1.2/)

[RFC2119] S. Bradner: “Key words for use in RFCs to Indicate Requirement Levels”, IETF RFC 2119 (http://www.ietf.org/rfc/rfc2119.txt)

[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)

[RFC3161] C. Adams, P. Cain, D. Pinkas, R. Zuccherato: “*Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*”, IETF RFC 3161 (http://www.ietf.org/rfc/rfc3161.txt)

[RFC3275] D. Eastlage, J. Reagle, D. Solo: “*(Extensible Markup Language) XML Signature Syntax and Processing”*, IETF RFC 3275 (http://www.ietf.org/rfc/rfc3275.txt)

[RFC3281] S. Farrell, R. Housley: “*An Internet Attribute Certificate Profile for Authorization”*, IETF RFC 3281 (http://www.ietf.org/rfc/rfc3281.txt)

[RFC3852] R. Housley: “*Cryptographic Message Syntax (CMS)”*. IETF RFC 3852, (http://www.ietf.org/rfc/rfc3852.txt)

[RFC4514] K. Zeilenga, Ed.: “*Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names*”, IETF RFC 4514 (http://www.ietf.org/rfc/rfc4514.txt)

[RFC4998] T. Gondrom, R. Brandner, U. Pordesch: “*Evidence Record Syntax (ERS)”*, IETF RFC 4998 (http://www.ietf.org/rfc/rfc4998.txt)

[RFC5280] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk: “*Internet X.509 Public Key Infrastructure, Certificate and Certificate Revocation List (CRL) Profile”*, IETF RFC 5280 (http://www.ietf.org/rfc/rfc5280.txt)

[SAMLCore1.1] OASIS Standard, *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V 1.1*, September 2003 http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf

[SAMLCore2.0] OASIS Standard, *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0,* March 2005 http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[XAdES] ETSI: “*XML Advanced Electronic Signatures (XAdES)”*, ETSI TS 101 903, Version 1.3.2, March 2006

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

[XMLSig] D. Eastlake et al. “*XML-Signature Syntax and Processing”,* W3C Recommendation, June 2008 (http://www.w3.org/TR/xmldsig-core/)

1.3 Namespaces

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

This schema is associated with the following XML namespace:

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

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

Conventional XML namespace prefixes are used in this document:

- The prefix `vr:` (or no prefix) stands for this profiles namespace **[DSSVR-XSD]**.
- The prefix `ds:` stands for the W3C XML Signature namespace **[XMLSig]**.
- The prefix `dss:` stands for the DSS core namespace **[Core-XSD]**.

- The prefix `saml:` stands for the OASIS SAML Schema namespace **[SAMLCore1.1]**.
- The prefix `xades:` stands for ETSI XML Advanced Electronic Signatures (XAdES) document **[XAdES]**.

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

2 Profile Features

2.1 Overview

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

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

Therefore the present profile defines how

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

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

2.2 Scope

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

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

- the type of signature object,
- the transport binding or
- the security binding.

2.3 Relationship To Other Profiles

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

2.4 Profile Identifier

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

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

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

3 Verification Reports within DSS Verifying Protocol

3.1 Element <ReturnVerificationReport>

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:

```
   <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>
```

It contains the following elements:

<IncludeVerifier> [Default]

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’.

<IncludeCertificateValues> [Default]

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’.

<IncludeRevocationValues> [Default]

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’.

<ExpandBinaryValues> [Default]

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’.

<ReportDetailLevel> [Optional]

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

- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:reportdetail:noDetails
  For every signature only the final result of the verification is reported.
- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:reportdetail:noPathDetails
  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.

– urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:reportdetail:allDetails
For every signature, the certificate path details and details on the validation of individual certificates in the path are requested. For every signature, the certificate path and each individual certificate the details are reported. If the `<ReportDetailLevel>`-element is missing, this option is assumed as default.

3.2 Element <VerificationReport>

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

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

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

```
<complexType name="VerificationReportType">
	<sequence>
		<element ref="dss:VerificationTimeInfo" maxOccurs="1"
			minOccurs="0" />
		<element name="VerifierIdentity" type="vr:IdentifierType"
			maxOccurs="1" minOccurs="0" />
		<element name="IndividualReport" maxOccurs="unbounded"
			type="vr:IndividualReportType" minOccurs="0" />
	</sequence>
</complexType>
```

It contains the following elements:

`<VerificationTimeInfo>` [Optional]

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

`<VerifierIdentity>` [Optional]

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

`<IndividualReport>` [Optional, Unbounded]

For each independent[1] signed object (signature, time stamp, certificate, CRL, OCSP-response, evidence record etc.) that has been used in the signature verification process there will be one

[1] A signed object x is called independent of another signed object y, if x was produced and can be verified without y and y was produced and can be verified without x.

If a time stamp, certificate, CRL, OCSP-response etc. is included as unsigned attribute or property in an advanced electronic signature it is not independent of the signature for example.

<IndividualReport>-element in the verification report. The details of this element are specified in the following section.

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

```
  <complexType name="IdentifierType">
  <sequence>
        <element ref="ds:X509Data" maxOccurs="1" minOccurs="0" />
        <element name="SAMLv1Identifier" type="saml:NameIdentifierType"
              maxOccurs="1" minOccurs="0" />
        <element name="SAMLv2Identifier" type="saml2:NameIDType"
              maxOccurs="1" minOccurs="0" />
        <element name="Other" type="dss:AnyType" maxOccurs="1"
              minOccurs="0" />
  </sequence>
</complexType>
```

It MAY contain the following elements or other identifying information:

`<ds:X509Data>` [Optional]

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

`<SAMLv1Identifier>` [Optional]

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

`<SAMLv2Identifier>` [Optional]

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

`<Other>` [Optional]

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

3.3 Element <IndividualReport>

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

```
<complexType name="IndividualReportType">
  <sequence>
        <element name="SignedObjectIdentifier"
              type="vr:SignedObjectIdentifierType"/>
        <element ref="dss:Result"/>
        <element name="Details" type="dss:AnyType" maxOccurs="1"
              minOccurs="0" />
  </sequence>
</complexType>
```

It contains the following elements:

`<SignedObjectIdentifier>` [Required]

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

<Result> [Required]

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

<Details> [Optional]

The <Details> element MAY contain a detailed report for the signature or validation data under consideration or any other signature-specific optional output defined in Section 4.5 of **[DSSCore]**. The corresponding elements, which are specified in this document for this purpose are listed in Section 4.2.

The **SignedObjectIdentifierType** is defined as follows:

```
<complexType name="SignedObjectIdentifierType">
   <sequence>
      <element name="DigestAlgAndValue"
         type="XAdES:DigestAlgAndValueType" maxOccurs="1" minOccurs="0"/>
      <element ref="ds:CanonicalizationMethod" maxOccurs="1" minOccurs="0" />
      <element name="SignedProperties"
         type="vr:SignedPropertiesType" maxOccurs="1" minOccurs="0" />
      <element ref="ds:SignatureValue" maxOccurs="1" minOccurs="0" />
      <element name="Other" type="dss:AnyType" maxOccurs="1" minOccurs="0" />
   </sequence>
   <attribute name="WhichDocument" type="IDREF" use="optional"/>
   <attribute name="XPath" type="string" use="optional"/>
   <attribute name="Offset" type="integer" use="optional"/>
   <attribute name="FieldName" type="string" use="optional"/>
</complexType>
```

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

It contains the following attributes and elements:

<DigestAlgAndValue> [Optional]

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

<ds:CanonicalizationMethod> [Optional]

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

<SignedProperties> [Optional]

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

<ds:SignatureValue> [Optional]

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

`<Other>` [Optional]

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

`WhichDocument` [Optional]

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

`XPath` [Optional]

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

`Offset` [Optional]

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

`FieldName` [Optional]

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

3.4 VerificationResultType

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

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

```
<complexType name="VerificationResultType">
   <sequence>
         <element name="ResultMajor" type="anyURI"/>
         <element name="ResultMinor" type="anyURI" minOccurs="0"/>
         <element name="ResultMessage" type="dss:InternationalStringType"
                minOccurs="0"/>
         <any namespace="##other" processContents="lax" minOccurs="0"
                maxOccurs="unbounded"/>
   </sequence>
</complexType>
```

`<ResultMajor>` [Required]

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

- `urn:oasis:names:tc:dss:1.0:detail:valid`
- `urn:oasis:names:tc:dss:1.0:detail:invalid`
- `urn:oasis:names:tc:dss:1.0:detail:indetermined`

`<ResultMinor>` [Optional]

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

`<ResultMessage>` [Optional]

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

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

3.5 Element <DetailedSignatureReport>

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

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

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

```
<complexType name="DetailedSignatureReportType">
   <sequence>
          <element name="FormatOK" type="vr:VerificationResultType" />
          <element name="Properties" type="vr:PropertiesType"
                 maxOccurs="1" minOccurs="0" />
          <element ref="dss:VerifyManifestResults" maxOccurs="1"
                 minOccurs="0" />
          <element name="SignatureHasVisibleContent" type="boolean"
                 maxOccurs="1" minOccurs="0"/>
          <element name="SignatureOK"
                 type="vr:SignatureValidityType" />
          <element name="CertificatePathValidity"
                 type="vr:CertificatePathValidityType" />
   </sequence>
</complexType>
```

It contains the following elements:

<FormatOK> [Required]

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

<Properties> [Optional]

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

<VerifyManifestResults> [Optional]

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

<SignatureHasVisibleContent> [Optional]

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

<SignatureOK> [Required]

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

<CertificatePathValidity> [Required]

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

3.5.1 SignatureValidityType

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

```
  <complexType name="SignatureValidityType">
        <sequence>
              <element name="SigMathOK" type="vr:VerificationResultType" />
              <element name="SignatureAlgorithm"
                    type="vr:AlgorithmValidityType"
                    maxOccurs="1" minOccurs="0"/>
        </sequence>
  </complexType>
```

It comprises the following elements:

`<SigMathOK>` [Required]

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

`<SignatureAlgorithm>` [Optional]

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

3.5.2 AlgorithmValidityType

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

```
<complexType name="AlgorithmValidityType">
  <sequence>
        <element name="Algorithm" type="anyURI" />
        <element name="Parameters" type="dss:AnyType" maxOccurs="1"
              minOccurs="0" />
        <element name="Suitability" type="vr:VerificationResultType"
              maxOccurs="1" minOccurs="0"/>
  </sequence>
</complexType>
```

`<Algorithm>` [Required]

This element contains the URI for the algorithm.

`<Parameters>` [Optional]

This element MAY contain further parameters for the cryptographic algorithm.

`<Suitabiltity>` [Optional]

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

3.5.3 CertificatePathValidityType

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

- **DetailedSignatureReportType** (see above),
- **AttributeCertificateValidityType** (see Section 3.5.4.3),

- **CRLValidityType** (see Section 3.5.3.4),
- **OCSPValidityType** (see Section 3.5.3.5) and
- **TimeStampValidityType** (see Section 3.5.4.4).

It is specified as follows:

```
<complexType name="CertificatePathValidityType">
      <sequence>
            <element name="PathValiditySummary"
                  type="vr:VerificationResultType" />
            <element name="CertificateIdentifier"
                  type="ds:X509IssuerSerialType" />
            <element name="PathValidityDetail"
                  type="vr:CertificatePathValidityDetailType"
                  minOccurs="0" maxOccurs="1"/>
      </sequence>
</complexType>
```

It contains the following elements:

`<PathValiditySummary>` [Required]

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

`<CertificateIdentifier>` [Required]

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

`<PathValidityDetail>` [Optional]

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

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

```
<complexType name="CertificatePathValidityDetailType">
      <sequence>
         <sequence maxOccurs="unbounded" minOccurs="0">
            <element name="CertificateValidity"
                  type="vr:CertificateValidityType" />
         </sequence>
         <element name="TSLValidity"
            type="dss:AnyType" maxOccurs="1" minOccurs="0" />
         <element name="TrustAnchor" type="vr:VerificationResultType" />
      </sequence>
</complexType>
```

It contains the following elements:

`<CertificateValidity>` [Optional, Unbounded]

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

`<TSLValidity>` [Optional]

This element contains information about the validity of a Trust-service Status List (TSL) according to **[ETSI102231-2.1.1]** or **[ETSI102231-3.1.2]** for example. This element SHOULD contain information about

- the TSL-scheme under consideration, as provided by a `SchemeInformation` element,
- the Trust-service providers and their services, as provided by a `TrustServiceProviderList` element,
- the measures for protecting the integrity and authenticity of the TSL-related information and the result of the corresponding verification step. If the integrity and authenticity is protected by means of an electronic signature, it is RECOMMENDED to include a `DetailedSignatureReport` element. If the integrity is protected by a time stamp it is RECOMMENDED to include an `IndividualTimeStamp` element etc. .

`<TrustAnchor>` [Required]

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

- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:trustanchor:SSCD – indicates that the trusted root certificate is stored within a secure signature creation device according to **[EC/1999/93]**.
- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:trustanchor:otherCard – indicates that the trusted root certificate is stored within some other hardware token.
- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:trustanchor:certDataBase – indicates that the trusted root certificate is stored within some certificate data base.
- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:trustanchor:other – indicates that the trusted root certificate is stored using other means.

3.5.3.1 CertificateValidityType

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

```
<complexType name="CertificateValidityType">
   <sequence>
      <element name="CertificateIdentifier" type="ds:X509IssuerSerialType" />
      <element name="Subject" type="string" />
      <element name="ChainingOK" type="vr:VerificationResultType"
          maxOccurs="1" minOccurs="0"/>
      <element name="ValidityPeriodOK" type="vr:VerificationResultType" />
      <element name="ExtensionsOK" type="vr:VerificationResultType" />
      <element name="CertificateValue" type="base64Binary"
          maxOccurs="1" minOccurs="0" />
      <element name="CertificateContent"
          type="vr:CertificateContentType" maxOccurs="1" minOccurs="0" />
      <element name="SignatureOK"
          type="vr:SignatureValidityType" />
      <element name="CertificateStatus" type="vr:CertificateStatusType" />
   </sequence>
</complexType>
```

It contains the following elements:

`<CertificateIdentifier>` [Required]

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

`<Subject>` [Required]

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

`<ChainingOK>` [Optional]

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

`<ValidityPeriodOK>` [Required]

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

`<ExtensionsOK>` [Required]

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

`<CertificateValue>` [Optional]

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

`<CertificateContent>` [Optional]

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

`<SignatureOK>` [Required]

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

`<CertificateStatus>` [Required]

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

3.5.3.2 CertificateContentType

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

```
<complexType name="CertificateContentType">
        <sequence>
                <element name="Version" type="integer" maxOccurs="1"
                        minOccurs="0" />
                <element name="SerialNumber" type="integer" />
                <element name="SignatureAlgorithm" type="anyURI" />
                <element name="Issuer" type="string" />
                <element name="ValidityPeriod" type="vr:ValidityPeriodType" />
                <element name="Subject" type="string" />
                <element name="Extensions" type="vr:ExtensionsType"
                        minOccurs="0" />
```

```
    </sequence>
</complexType>
```

It contains the following elements:

`<Version>` [Optional]

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

`<SerialNumber>` [Required]

This element MUST contain the serial number of the certificate.

`<SignatureAlgorithm>` [Required]

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

`<Issuer>` [Required]

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

`<ValidityPeriod>` [Required]

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

`<Subject>` [Required]

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

`<Extensions>` [Optional]

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

The **ValidityPeriodType** is specified as follows:

```
<complexType name="ValidityPeriodType">
    <sequence>
        <element name="NotBefore" type="dateTime" />
        <element name="NotAfter" type="dateTime" />
    </sequence>
</complexType>
```

It contains the following elements:

`<NotBefore>` [Required]

The certificate is not valid before this point in time.

`<NotAfter>` [Required]

The certificate is not valid after this point in time.

The **ExtensionsType** is specified as follows:

```
<complexType name="ExtensionsType">
    <sequence minOccurs="0" maxOccurs="unbounded">
```

```
                <element name="Extension" type="vr:ExtensionType" />
          </sequence>
    </complexType>
```

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

```
    <complexType name="ExtensionType">
          <sequence>
             <element name="ExtnId" type="XAdES:ObjectIdentifierType" />
             <element name="Critical" type="boolean" />
             <element name="ExtnValue" type="dss:AnyType" maxOccurs="1"
                minOccurs="0" />
             <element name="ExtensionOK" type="vr:VerificationResultType" />
          </sequence>
    </complexType>
```

It contains the following elements:

`<ExtnId>` [Required]

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

`<Critical>` [Required]

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

`<ExtnValue>` [Optional]

This element SHOULD contain the value of the extension as an XML-structure, which mirrors the original ASN.1-definition of the extension.

`<ExtensionOK>` [Required]

This element contains information about the validity of the specific extension within the given context of the certificate.

3.5.3.3 CertificateStatusType

The **CertificateStatusType** is defined as follows:

```
    <complexType name="CertificateStatusType">
          <sequence>
             <element name="CertStatusOK" type="vr:VerificationResultType" />
             <element name="RevocationInfo" maxOccurs="1"
                minOccurs="0">
                <complexType>
                      <sequence>
                            <element name="RevocationDate" type="dateTime" />
                            <element name="RevocationReason"
                                         type="vr:VerificationResultType" />
                      </sequence>
                </complexType>
          </element>
          <element name="RevocationEvidence" maxOccurs="1" minOccurs="0">
              <complexType>
                     <choice>
                            <element name="CRLValidity"
                                   type="vr:CRLValidityType" />
                            <element name="CRLReference"
                                   type="XAdES:CRLIdentifierType" />
                            <element name="OCSPValidity"
                                   type="vr:OCSPValidityType" />
                            <element name="OCSPReference"
                                   type="XAdES:OCSPIdentifierType" />
                            <element name="Other" type="dss:AnyType"/>
                     </choice>
              </complexType>
          </element>
       </sequence>
</complexType>
```

It contains the following elements:

<CertStatusOK> [Required]

This element MUST contain the status of the certificate.

<RevocationInfo> [Optional]

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

- <RevocationDate>
  contains the date and time of revocation.
- <RevocationReason>
  contains the reason for revocation. Following the definition of CRLReason in **[RFC5280]** there are the following URIs to specify the revocation reason:
  - urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:unspecified
  - urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:keyCompromise
  - urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:cACompromise
  - urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:affiliationChanged
  - urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:superseded
  - urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:cessationOfOperation
  - urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:certificateHold
  - urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:removeFromCRL
  - urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:privilegeWithdrawn
  - urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:revocationreason:aACompromise

<RevocationEvidence> [Optional, Choice]

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

- <CRLValidity>
  This element contains information about the used CRL and its validity. The **CRLValidityType** is defined in Section 3.5.3.4.

- `<CRLReference>`
  This element contains a reference to the CRL in case it is already included elsewhere in the present verification report. The **XAdES:CRLIdentifierType** is defined in **[XAdES]**.
- `<OCSPValidity>`
  This element contains information about the used OCSP response and its validity. The **OCSPValidityType** is defined in Section 3.5.3.5.
- `<OCSPReference>`
  This element contains a reference to the used OCSP response, if it is already included elsewhere in the present verification report. The **XAdES:OCSPIdentifierType** is defined in **[XAdES]**.
- `<Other>`
  This element MAY contain information about alternative sources of revocation information.

3.5.3.4 CRLValidityType

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

```
<complexType name="CRLValidityType">
       <sequence>
              <element name="CRLIdentifier" type="XAdES:CRLIdentifierType"
                     maxOccurs="1" minOccurs="1" />
              <element name="CRLValue" type="base64Binary"
                     maxOccurs="1" minOccurs="0" />
              <element name="CRLContent" type="vr:CRLContentType"
                     maxOccurs="1" minOccurs="0" />
              <element name="SignatureOK" type="vr:SignatureValidityType" />
              <element name="CertificatePathValidity"
                     type="vr:CertificatePathValidityType" />
       </sequence>
       <attribute name="Id" type="ID" use="optional" />
</complexType>
```

It contains the following attributes and elements:

`Id` [Optional]

This attribute contains an optional identifier for the element.

`<CRLIdentifier>` [Required]

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

`<CRLValue>` [Optional]

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

`<CRLContent>` [Optional]

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

`<SignatureOK>` [Required]

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

`<CertificatePathValidity>` [Required]

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

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

```
<complexType name="CRLContentType">
      <sequence>
            <element name="Version" minOccurs="0" type="integer" />
            <element name="Signature" type="anyURI" />
            <element name="Issuer" type="string" />
            <element name="ThisUpdate" type="dateTime" />
            <element name="NextUpdate" minOccurs="0" type="dateTime" />
            <element name="RevokedCertificates" minOccurs="0">
               <complexType>
                   <sequence minOccurs="0" maxOccurs="unbounded">
                         <element name="UserCertificate" type="integer" />
                         <element name="RevocationDate" type="dateTime" />
                         <element name="CrlEntryExtensions" minOccurs="0"
                               type="vr:ExtensionsType" />
                   </sequence>
               </complexType>
            </element>
            <element name="CrlExtensions" type="vr:ExtensionsType"
                  minOccurs="0" />
      </sequence>
</complexType>
```

It contains the following elements:

`<Version>` [Optional]

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

`<Signature>` [Required]

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

`<Issuer>` [Required]

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

`<ThisUpdate>` [Required]

This element contains the issue date of the CRL.

`<NextUpdate>` [Optional]

This element contains, if present, the date by which the next CRL will be issued.

`<RevokedCertificates>` [Optional]

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

`<CrlExtensions>` [Optional]

If present, this element contains information about the list of extensions present in the CRL under consideration. The **ExtensionType** is defined in Section 3.5.3.2.

3.5.3.5 OCSPValidityType

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

```
   <complexType name="OCSPValidityType">
          <sequence>
             <element name="OCSPIdentifier" type="XAdES:OCSPIdentifierType" />
             <element name="OCSPValue" type="base64Binary"
                maxOccurs="1" minOccurs="0" />
             <element name="OCSPContent" type="vr:OCSPContentType"
                maxOccurs="1" minOccurs="0" />
             <element name="SignatureOK" type="vr:SignatureValidityType" />
             <element name="CertificatePathValidity"
                type="vr:CertificatePathValidityType" />
          </sequence>
          <attribute name="Id" type="ID" use="optional" />
   </complexType>
```

It contains the following attributes and elements:

`Id` [Optional]

This attribute contains an optional identifier for the element.

`<OCSPIdentifier>` [Required]

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

`<OCSPValue>` [Optional]

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

`<OCSPContent>` [Optional]

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

`<SignatureOK>` [Required]

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

`<CertificatePathValidity>` [Required]

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

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

```
<complexType name="OCSPContentType">
         <sequence>
            <element name="Version" type="integer" />
            <element name="ResponderID" type="string" />
            <element name="producedAt" type="dateTime" />
            <element name="Responses">
                <complexType>
                   <sequence maxOccurs="unbounded" minOccurs="0">
                       <element name="SingleResponse"
                              type="vr:SingleResponseType" />
                   </sequence>
                </complexType>
            </element>
            <element name="ResponseExtensions" type="vr:ExtensionsType"
                maxOccurs="1" minOccurs="0"/>
```

```
        </sequence>
</complexType>
```

It contains the following elements:

`<Version>` [Required]

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

`<ResponderID>` [Required]

This element contains the name of the OCSP-responder.

`<producedAt>` [Required]

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

`<Responses>` [Required]

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

`<ResponseExtensions>` [Optional]

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

The **SingleResponseType** is specified as follows:

```
<complexType name="SingleResponseType">
        <sequence>
           <element name="CertID">
               <complexType>
                       <sequence>
                          <element name="HashAlgorithm" type="anyURI" />
                          <element name="IssuerNameHash" type="hexBinary" />
                          <element name="IssuerKeyHash" type="hexBinary" />
                          <element name="SerialNumber" type="integer" />
                       </sequence>
               </complexType>
           </element>
           <element name="CertStatus" type="vr:VerificationResultType" />
           <element name="ThisUpdate" type="dateTime" />
           <element name="NextUpdate" type="dateTime" maxOccurs="1"
               minOccurs="0" />
           <element name="SingleExtensions" type="vr:ExtensionsType"
               maxOccurs="1" minOccurs="0" />
        </sequence>
</complexType>
```

It contains the following elements:

`<CertID>` [Required]

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

`<CertStatus>` [Required]

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

- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:certstatus:good
- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:certstatus:revoked

- urn:oasis:names:tc:dss-x:1.0:profiles:verificationreport:certstatus:unknown

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

<ThisUpdate> [Required]

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

<NextUpdate> [Optional]

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

<SingleExtensions> [Optional]

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

3.5.4 PropertiesType

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

```
<complexType name="PropertiesType">
      <sequence>
            <element name="SignedProperties"
                  type="vr:SignedPropertiesType" minOccurs="0" />
            <element name="UnsignedProperties"
                  type="vr:UnsignedPropertiesType" minOccurs="0" />
      </sequence>
      <attribute name="Id" type="ID" use="optional" />
</complexType>
```

It contains the following attributes and elements:

Id [Optional]

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

<SignedProperties> [Optional]

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

<UnsignedProperties> [Optional]

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

3.5.4.1 Signed Properties

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

```
<complexType name="SignedPropertiesType">
      <sequence>
            <element name="SignedSignatureProperties"
                  type="vr:SignedSignaturePropertiesType" maxOccurs="1"
                  minOccurs="0" />
            <element name="SignedDataObjectProperties"
                  type="vr:SignedDataObjectPropertiesType"
                  minOccurs="0" />
```

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

It contains the following attributes and elements:

`Id` [Optional]

This attribute contains an optional identifier for the element.

`<SignedSignatureProperties>` [Optional]

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

`<SignedDataObjectProperties>` [Optional]

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

`<Other>` [Optional]

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

3.5.4.1.1 SignedSignaturePropertiesType

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

```
  <complexType name="SignedSignaturePropertiesType">
  <sequence>
        <element ref="XAdES:SigningTime" maxOccurs="1" minOccurs="0" />
        <element ref="XAdES:SigningCertificate" maxOccurs="1"
              minOccurs="0" />
        <element ref="XAdES:SignaturePolicyIdentifier" maxOccurs="1"
              minOccurs="0" />
        <choice maxOccurs="1" minOccurs="0">
              <element ref="XAdES:SignatureProductionPlace" />
              <element name="Location" type="string" />
        </choice>
        <element name="SignerRole" type="vr:SignerRoleType" minOccurs="0" />
  </sequence>
</complexType>
```

It MAY contain the following elements:

`<XAdES:SigningTime>` [Optional]

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

`<XAdES:SigningCertificate>` [Optional]

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

`<XAdES:SignaturePolicyIdentifier>` [Optional]

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

`<XAdES:SignatureProductionPlace>` [Optional, Choice]

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

`<Location>` [Optional, Choice]

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

`<SignerRole>` [Optional]

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

The **SignerRoleType** is specified as follows:

```
<complexType name="SignerRoleType">
      <sequence>
            <element name="ClaimedRoles"
                  type="XAdES:ClaimedRolesListType" minOccurs="0" />
            <element name="CertifiedRoles"
                  type="vr:CertifiedRolesListType" minOccurs="0" />
      </sequence>
</complexType>
```

It MAY contain the following elements:

`<ClaimedRoles>` [Optional]

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

`<CertifiedRoles>` [Optional]

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

The **CertifiedRolesListType** is specified as follows:

```
<complexType name="CertifiedRolesListType">
      <sequence>
      <element name="AttributeCertificateValidity"
                  type="vr:AttributeCertificateValidityType"
                  maxOccurs="unbounded" />
      </sequence>
</complexType>
```

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

3.5.4.1.2 SignedDataObjectPropertiesType

The **SignedDataObjectPropertiesType** is defined as follows:

```
<complexType name="SignedDataObjectPropertiesType">
<sequence>
      <element ref="XAdES:DataObjectFormat" maxOccurs="unbounded"
            minOccurs="0" />
      <choice maxOccurs="1" minOccurs="0">
            <element ref="XAdES:CommitmentTypeIndication"
                  maxOccurs="unbounded" minOccurs="1"/>
      <element name="Reason" type="string" />
```

```
         </choice>
         <element name="AllDataObjectsTimeStamp"
                type="vr:TimeStampValidityType" minOccurs="0"
                maxOccurs="unbounded" />
         <element name="IndividualDataObjectsTimeStamp"
                type="vr:TimeStampValidityType" minOccurs="0"
                maxOccurs="unbounded" />
   </sequence>
   <attribute name="Id" type="ID" use="optional" />
</complexType>
```

It contains the following attributes and elements:

`Id` [Optional]

This attribute contains an optional identifier for the element.

`<XAdES:DataObjectFormat>` [Optional, Unbounded]

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

`<XAdES:CommitmentTypeIndication>` [Choice, Unbounded]

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

`<Reason>` [Choice]

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

`<AllDataObjectsTimeStamp>` [Optional, Unbounded]

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

`<IndividualDataObjectsTimeStamp>` [Optional, Unbounded]

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

3.5.4.2 Unsigned Properties

The **UnsignedPropertiesType** is specified as follows:

```
<complexType name="UnsignedPropertiesType">
   <sequence>
         <element name="UnsignedSignatureProperties"
                type="vr:UnsignedSignaturePropertiesType" minOccurs="0" />
         <element ref="XAdES:UnsignedDataObjectProperties"
                maxOccurs="1" minOccurs="0" />
         <element name="Other" type="dss:AnyType" maxOccurs="1"
                minOccurs="0">
         </element>
   </sequence>
   <attribute name="Id" type="ID" use="optional" />
</complexType>
```

It contains the following attributes and elements:

`Id` [Optional]

This attribute contains an optional identifier for the element.

<UnsignedSignatureProperties> [Optional]

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

<XAdES:UnsignedDataObjectProperties> [Optional]

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

<Other> [Optional]

This element MAY contain information about other unsigned properties.

The **UnsignedSignaturePropertiesType** is defined as follows:

```
<complexType name="UnsignedSignaturePropertiesType">
   <choice maxOccurs="unbounded">
         <element name="CounterSignature" type="vr:SignatureValidityType" />
         <element name="SignatureTimeStamp" type="vr:TimeStampValidityType" />
         <element ref="XAdES:CompleteCertificateRefs" />
         <element ref="XAdES:CompleteRevocationRefs" />
         <element ref="XAdES:AttributeCertificateRefs" />
         <element ref="XAdES:AttributeRevocationRefs" />
         <element name="SigAndRefsTimeStamp"
                type="vr:TimeStampValidityType" />
         <element name="RefsOnlyTimeStamp" type="vr:TimeStampValidityType" />
         <element name="CertificateValues" type="vr:CertificateValuesType" />
         <element name="RevocationValues" type="vr:RevocationValuesType" />
         <element name="AttrAuthoritiesCertValues"
                type="vr:CertificateValuesType" />
         <element name="AttributeRevocationValues"
                type="vr:RevocationValuesType" />
         <element name="ArchiveTimeStamp" type="vr:TimeStampValidityType" />
   </choice>
   <attribute name="Id" type="ID" use="optional" />
</complexType>
```

It contains the following attributes and elements:

Id [Optional]

This attribute contains an optional identifier for the element.

<CounterSignature> [Choice]

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

<SignatureTimeStamp> [Choice]

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

<XAdES:CompleteCertificateRefs> [Choice]

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

<XAdES:CompleteRevocationRefs> [Choice]

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

<XAdES:AttributeCertificateRefs> [Choice]

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

<XAdES:AttributeRevocationRefs> [Choice]

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

<SigAndRefsTimeStamp> [Choice]

Contains verification results for a time stamp referring to the signature and references on certificates and revocation data (see section 7.5.1 of **[XAdES]**). The **TimeStampValidityType** is described in section 3.5.4.4.

<RefsOnlyTimeStamp> [Choice]

Contains verification results for a time stamp referring only to references on certificates and revocation data (see section 7.5.2 of **[XAdES]**). The **TimeStampValidityType** is described in section 3.5.4.4.

<CertificateValues> [Choice]

Contains verification results for the certificates, which were used in the verification of the signature (see section 7.6.1 of **[XAdES]**). The **CertificateValuesType** is defined below.

<RevocationValues> [Choice]

Contains verification results of the revocation data used in the verification of the signature (see section 7.6.2 of **[XAdES]**). The **RevocationValuesType** is defined below.

<AttrAuthoritiesCertValues> [Choice]

Contains verification results of the certificates of Attribute Authorities that have been used to validate the attribute certificates, which are contained in the signature (see section 7.6.3 of **[XAdES]**). The **CertificateValuesType** is defined below.

<AttributeRevocationValues> [Choice]

Contains verification results of the revocation data that have been used to validate the attribute certificate when present in the signature (see section 7.6.4 of **[XAdES]**). The **RevocationValuesType** is defined below.

<ArchiveTimeStamp> [Choice]

Contains verification results for a time stamp covering the complete signature including all attributes (see section 7.7 of **[XAdES]**). The **TimeStampValidityType** is described in section 3.5.4.4.

The **CertificateValuesType** is defined as follows:

```
<complexType name="CertificateValuesType">
       <choice minOccurs="0" maxOccurs="unbounded">
              <element name="EncapsulatedX509Certificate"
                     type="vr:CertificateValidityType" />
              <element name="OtherCertificate" />
       </choice>
       <attribute name="Id" type="ID" use="optional" />
</complexType>
```

It defines the following attributes and elements:

Id [Optional]

This attribute contains an optional identifier for the element.

<EncapsulatedX509Certificate> [Optional, Unbounded, Choice]

Contains verification results for an X.509 certificate included in the signature. The **CertificateValidityType** is defined in Section 3.5.3.1.

`<OtherCertificate>` [Optional, Unbounded, Choice]

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

The **RevocationValuesType** is defined as follows:

```
<complexType name="RevocationValuesType">
       <sequence>
          <element name="CRLValues" minOccurs="0">
             <complexType>
                    <sequence maxOccurs="unbounded" minOccurs="1">
                          <element name="VerifiedCRL"
                                 type="vr:CRLValidityType" />
                    </sequence>
             </complexType>
          </element>
          <element name="OCSPValues" minOccurs="0">
             <complexType>
                    <sequence maxOccurs="unbounded" minOccurs="1">
                          <element name="VerifiedOCSPResponse"
                                 type="vr:OCSPValidityType" />
                    </sequence>
             </complexType>
          </element>
          <element name="OtherValues" type="dss:AnyType" minOccurs="0" />
       </sequence>
       <attribute name="Id" type="ID" use="optional" />
</complexType>
```

It contains the following attributes and elements:

`Id` [Optional]

This attribute contains an optional identifier for the element.

`<CRLValues>` [Optional]

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

`<OCSPValues>` [Optional]

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

`<OtherValues>` [Optional]

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

3.5.4.3 AttributeCertificateValidityType

The **AttributeCertificateValidityType** is defined as follows:

```
<complexType name="AttributeCertificateValidityType">
       <sequence>
             <element name="AttributeCertificateIdentifier"
                    type="vr:AttrCertIDType" maxOccurs="1" minOccurs="0" />
```

```
                <element name="AttributeCertificateValue" type="base64Binary"
                        maxOccurs="1" minOccurs="0" />
                <element name="AttributeCertificateContent"
                        type="vr:AttributeCertificateContentType" maxOccurs="1"
                        minOccurs="0" />
                <element name="SignatureOK" type="vr:SignatureValidityType" />
                <element name="CertificatePathValidity"
                        type="vr:CertificatePathValidityType" />
        </sequence>
</complexType>
```

It contains the following elements:

`<AttributeCertificateIdentifier>` [Optional]

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

`<AttributeCertificateValue>` [Optional]

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

`<AttributeCertificateContent>` [Optional]

This element MAY contain an XML-based analogue of the content of the certificate, if the report option `<ExpandBinaryValues>` is set to ‘true’. The structure of the `AttributeCertificateContentType` is defined below.

`<SignatureOK>` [Required]

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

`<CertificatePathValidity>` [Required]

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

The **AttrCertIDType** is structured as follows:

```
<complexType name="AttrCertIDType">
        <sequence>
                <element name="Holder" type="vr:EntityType" maxOccurs="1"
                        minOccurs="0"/>
                <element name="Issuer" type="vr:EntityType" />
                <element name="SerialNumber" type="integer" />
        </sequence>
</complexType>
```

It contains the following elements:

`<Holder>` [Optional]

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

<Issuer> [Required]

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

<SerialNumber> [Required]

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

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

```
<complexType name="EntityType">
      <sequence>
            <element name="BaseCertificateID"
                  type="ds:X509IssuerSerialType" maxOccurs="1"
                  minOccurs="0"/>
            <element name="Name" type="string" maxOccurs="1"
                  minOccurs="0"/>
            <element name="Other" type="dss:AnyType" maxOccurs="1"
                  minOccurs="0"/>
      </sequence>
</complexType>
```

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

<BaseCertificateID> [Optional]

This element identifies, if present, the public-key certificate of the entity. The structure of the ds:X509IssuerSerielType is defined in **[RFC3275]**.

<Name> [Optional]

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

<Other> [Optional]

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

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

```
<complexType name="AttributeCertificateContentType">
      <sequence>
            <element name="Version" minOccurs="0" type="integer" />
            <element name="Holder" type="vr:EntityType" />
            <element name="Issuer" type="vr:EntityType" />
            <element name="SignatureAlgorithm" type="anyURI" />
            <element name="SerialNumber" type="integer" />
            <element name="AttCertValidityPeriod"
                  type="vr:ValidityType" />
            <element name="Attributes">
                  <complexType>
                        <sequence minOccurs="0" maxOccurs="unbounded">
                              <element name="Attribute"
                                    type="vr:AttributeType" />
                        </sequence>
                  </complexType>
            </element>
```

```
                <element name="IssuerUniqueID" type="hexBinary" maxOccurs="1"
                      minOccurs="0"/>
                <element name="Extensions" minOccurs="0"
                      type="vr:ExtensionsType" />
          </sequence>
   </complexType>
```

It contains the following elements:

`<Version>` [Optional]

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

`<Holder>` [Required]

This element contains information about the holder of the certificate. The structure of the **EntityType** is defined above.

`<Issuer>` [Required]

This element contains the issuer of the attribute certificate. The structure of the **EntityType** is defined above.

`<SignatureAlgorithm>` [Required]

This element contains an identifier of the used signature algorithm.

`<SerialNumber>` [Required]

This element contains the serial number of the attribute certificate.

`<AttCertValidityPeriod>` [Required]

This element contains the validity period of the attribute certificate. The **ValidityType** is defined in section 3.5.3.2.

`<Attributes>` [Optional, Unbounded]

This element contains, if present, a list of attributes. The **AttributeType** is defined below.

`<IssuerUniqueID>` [Optional]

This element contains, if present, a unique identifier of the issuer of the attribute certificate.

`<Extensions>` [Optional]

If present, this element contains information about the list of extensions present in the attribute certificate. The **ExtensionType** is defined in Section 3.5.3.2.

The **AttributeType** is defined as follows:

```
   <complexType name="AttributeType">
          <sequence>
                <element name="Type" type="anyURI" />
                <element name="Value" type="dss:AnyType" maxOccurs="unbounded"
                      minOccurs="0" />
          </sequence>
   </complexType>
```

It contains the following elements:

`<Type>` [Required]

This element MUST contain an identifier for the type of the attribute in the `<Code>`-element and MAY contain further information.

`<Value>` [Optional, Unbounded]

This element MAY contain any number of attribute values.

3.5.4.4 TimeStampValidityType

The **TimeStampValidityType** is structured as follows:

```
<complexType name="TimeStampValidityType">
        <sequence>
                <element name="FormatOK" type="vr:VerificationResultType" />
                <element name="TimeStampContent" type="vr:TstContentType"
                        maxOccurs="1" minOccurs="0" />
                <element name="MessageHashAlgorithm"
                        type="vr:AlgorithmValidityType"
                        maxOccurs="1" minOccurs="0" />
                <element name="SignatureOK"
                        type="vr:SignatureValidityType" />
                <element name="CertificatePathValidity"
                        type="vr:CertificatePathValidityType" />
        </sequence>
        <attribute name="Id" type="ID" use="optional" />
</complexType>
```

It contains the following elements and attributes:

`Id` [Optional]

This attribute contains an optional identifier for the element.

`<FormatOK>` [Required]

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

`<TimeStampContent>` [Optional]

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

`<MessageHashAlgorithm>` [Optional]

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

`<SignatureOK>` [Required]

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

`<CertificatePathValidity>` [Required]

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

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

```
<complexType name="TstContentType">
        <sequence>
                <element ref="dss:TstInfo" maxOccurs="1" minOccurs="0"/>
                <element name="Other" type="dss:AnyType" maxOccurs="1"
                        minOccurs="0"/>
        </sequence>
</complexType>
```

It contains the following elements:

<dss:TstInfo> [Optional]

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

<Other> [Optional]

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

3.5.5 Element <IndividualTimeStampReport>

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

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

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

3.5.6 Element <IndividualCertificateReport>

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

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

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

3.5.7 Element <IndividualAttributeCertificateReport>

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

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

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

3.5.8 Element <IndividualCRLReport>

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

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

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

3.5.9 Element <IndividualOCSPReport>

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

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

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

3.5.10 Element <EvidenceRecordReport>

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

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

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

```
<complexType name="EvidenceRecordValidityType">
       <sequence>
              <element name="FormatOK" type="vr:VerificationResultType" />
              <element name="Version" type="integer"
                     maxOccurs="1" minOccurs="0" />
              <element name="DigestAlgorithm"
                     type="vr:AlgorithmValidityType" maxOccurs="unbounded"
                     minOccurs="0">
              </element>
              <element name="CryptoInfos" maxOccurs="1" minOccurs="0">
                     <complexType>
                            <sequence>
                               <element name="Attribute" type="vr:AttributeType"
                                   maxOccurs="unbounded" minOccurs="1" />
                            </sequence>
                     </complexType>
              </element>
              <element name="EncryptionInfo" maxOccurs="1" minOccurs="0">
                     <complexType>
                            <sequence>
                               <element name="EncryptionInfoType"
                                   type="vr:AlgorithmValidityType" />
                               <element name="EncryptionInfoValue"
                                   type="dss:AnyType" />
                            </sequence>
                     </complexType>
              </element>
              <element name="ArchiveTimeStampSequence" maxOccurs="1"
                     minOccurs="1">
                     <complexType>
                            <sequence maxOccurs="unbounded" minOccurs="0">
                               <element name="ArchiveTimeStampChain">
                                   <complexType>
                                          <sequence maxOccurs="unbounded"
                                                 minOccurs="0">
                                            <element name="ArchiveTimeStamp"
                                            type="vr:ArchiveTimeStampValidityType"/>
                                          </sequence>
                                   </complexType>
                               </element>
                            </sequence>
                     </complexType>
              </element>
       </sequence>
       <attribute name="Id" type="ID" use="optional" />
</complexType>
```

It contains the following elements and attributes:

`Id` [Optional]

This attribute contains an optional identifier for the element.

`<FormatOK>` [Required]

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

`<Version>` [Optional]

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

`<DigestAlgorithm>` [Optional, unbounded]

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

`<CryptoInfos>` [Optional]

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

`<EncryptionInfo>` [Optional]

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

`<ArchiveTimeStampSequence>` [Required]

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

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

```
<complexType name="ArchiveTimeStampValidityType">
      <sequence>
             <element name="FormatOK" type="vr:VerificationResultType" />
             <element name="DigestAlgorithm" type="vr:AlgorithmValidityType"
                   maxOccurs="1" minOccurs="0" />
             <element name="Attributes" maxOccurs="1" minOccurs="0">
                   <complexType>
                          <sequence>
                                 <element name="Attribute" type="vr:AttributeType"
                                       maxOccurs="unbounded" minOccurs="1"/>
                          </sequence>
                   </complexType>
             </element>
             <element name="ReducedHashTree" maxOccurs="1" minOccurs="0">
                   <complexType>
                          <sequence maxOccurs="unbounded" minOccurs="1">
                                 <element name="PartialHashTree">
                                    <complexType>
                                        <sequence maxOccurs="unbounded"
                                              minOccurs="1">
                                              <element name="HashValue"
                                              type="vr:HashValueType"/>
                                        </sequence>
                                    </complexType>
                                 </element>
                          </sequence>
                   </complexType>
             </element>
             <element name="TimeStamp"
                   type="vr:TimeStampValidityType" />
      </sequence>
      <attribute name="Id" type="ID" use="optional" />
</complexType>
```

It contains the following elements and attributes:

Id [Optional]

This attribute contains an optional identifier for the element.

<FormatOK> [Required]

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

<DigestAlgorithm> [Optional]

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

<Attributes> [Optional]

This element contains, if present, information about further attributes related to the archive time stamp.

<ReducedHashTree> [Optional]

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

<TimeStamp> [Required]

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

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

```
<complexType name="HashValueType">
        <sequence>
                <element name="HashValue" type="hexBinary" />
        </sequence>
        <attribute name="HashedObject" type="IDREF" use="optional"/>
</complexType>
```

It contains the following elements and attributes:

HashedObject [Optional]

This attribute MAY be used to point to the object, which served as pre-image of the hash value.

<HashValue> [Required]

This element contains the hash value produced by applying the hash algorithm specified by the <DigestAlgorithm>- or <TimeStamp>-element to the data specified by the HashedObject attribute.

4 Conformance

This profile defines three conformance levels:

- Level 1 - “Basic”,
- Level 2 - “Comprehensive” and
- Level 3 - “Comfortable”.

4.1 Level 1 – “Basic”

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

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

4.2 Level 2 – “Comprehensive”

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

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

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

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

4.3 Level 3 – “Convenient”

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

A. Acknowledgements

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

Participants:

- Juan-Carlos Cruellas
- Andreas Kühne
- Ingo Henkel
- Ezer Farhi
- Stefan Drees
- Pim van der Eijk
- Clemens Orthacker
- Marta Cruellas
- Konrad Lanz

B. Revision History

Revision	Date	Editor	Changes Made
R1	19.07.2009	Detlef Hühnlein	CD1 version on current OASIS template
R2	15.03.2010	Detlef Hühnlein	Draft of CS1 version, which includes a clarifying footnote and minor editing
R3	16.06.2010	Detlef Hühnlein	Potential CS1 version, which uses `TSLValidity`-element of **dss:AnyType** and drops the previously used **TrustStatusListValidityType** in order to support different TSL-versions.
R4	15.07.2010	Detlef Hühnlein	Potential CS1 version, which provides textual recommendations for filling the `TSLValidity`-element.
R5	27.09.2010	Detlef Hühnlein	Editorial correction of references section