

TCG Infrastructure Working Group Core Integrity Schema Specification

Specification Version 1.0.1
Revision 1.0
17 November 2006
FINAL

Contacts:

ned.Smith@intel.com (Co-Chair, Editor)

THardjono@SignaCert.com (Co-Chair)

TCG

Public

Copyright © TCG 2006

Copyright © 2006 Trusted Computing Group, Incorporated.

Disclaimer

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Without limitation, TCG disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification and to the implementation of this specification, and TCG disclaims all liability for cost of procurement of substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, direct, indirect, or special damages, whether under contract, tort, warranty or otherwise, arising in any way out of use or reliance upon this specification or any information herein.

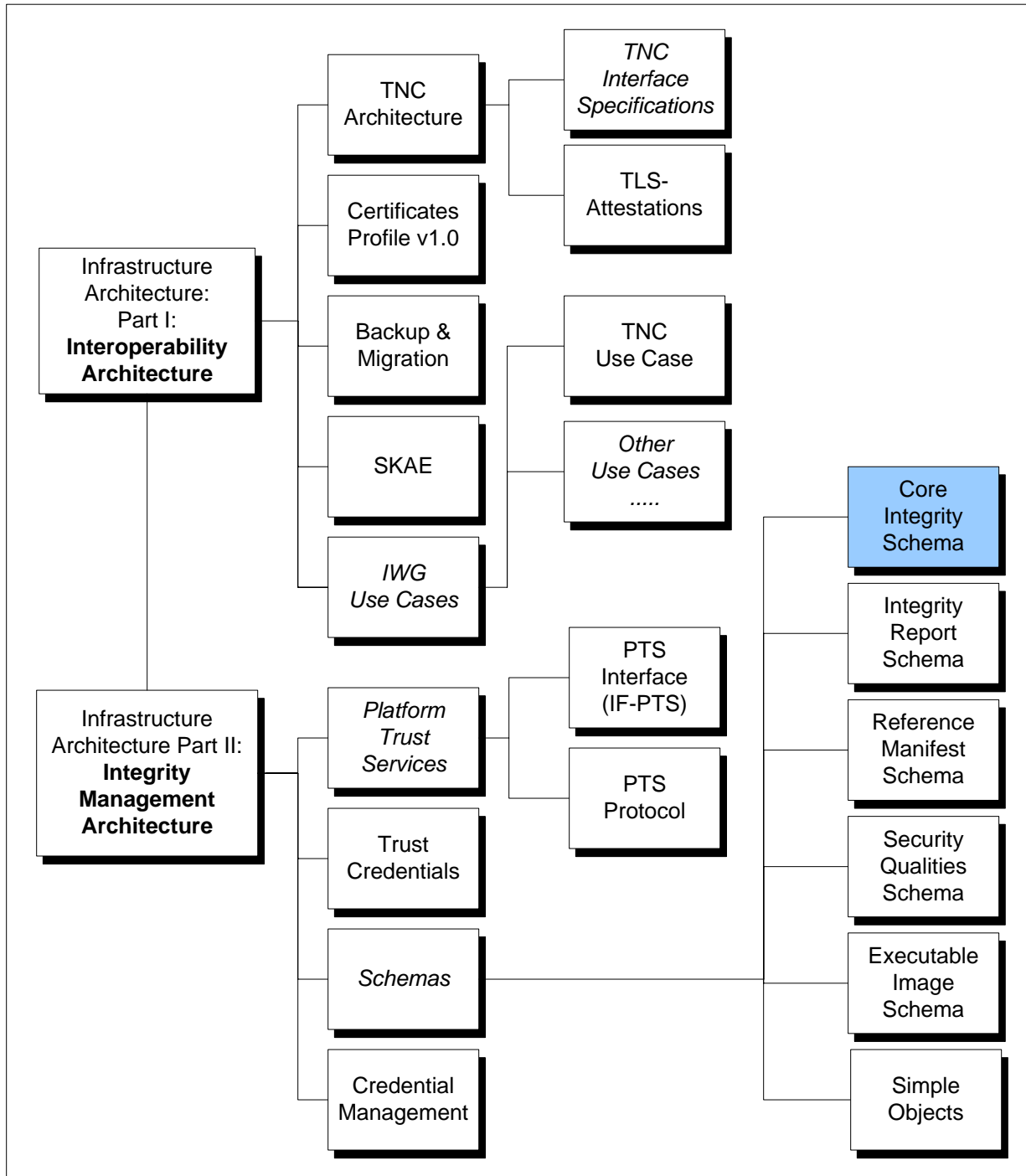
No license, express or implied, by estoppel or otherwise, to any TCG or TCG member intellectual property rights is granted herein.

Except that a license is hereby granted by TCG to copy and reproduce this specification for internal use only.

Contact the Trusted Computing Group at www.trustedcomputinggroup.org for information on specification licensing through membership agreements.

Any marks and brands contained herein are the property of their respective owners

IWG Document Roadmap



Acknowledgements

The TCG wishes to thank all those who contributed to this specification. This document builds on considerable work done in the various working groups in the TCG.

Special thanks to the members of the IWG who made significant contributions to this document:

Name	Company
Mark Redman	Freescale Semiconductor
Malcolm Duncan	CESG
Lee Terrell	IBM
Markus Gueller	Infineon
Ned Smith (editor, co-chair)	Intel Corporation
Thomas Hardjono (co-chair)	Signacert
David Bleckman	Signacert
Jeff Nisewanger	Sun
Wyllys Ingersoll	Sun
Paul Sangster	Symantec
Greg Kazmierczak	Wave Systems
Len Veil	Wave Systems

TABLE OF CONTENTS

1	SCOPE AND AUDIENCE	8
1.1	NORMATIVE AND NON-NORMATIVE	8
2	INTRODUCTION.....	9
2.1	SCHEMA VERSION	9
2.2	SCHEMA NAMESPACE.....	9
2.3	DEPENDENT SCHEMA DEFINITIONS	9
2.3.1	W3C XML Schema Syntax	9
2.3.2	W3C XML-Signature Syntax.....	10
3	CORE INTEGRITY SCHEMA	11
3.1	COMPLEX TYPES	11
3.1.1	<i>complexType AssertionType</i>	11
3.1.2	<i>complexType ComponentIDType</i>	12
3.1.3	<i>complexType ComponentRefType</i>	14
3.1.4	<i>complexType ConfidenceValueType</i>	15
3.1.5	<i>complexType DigestMethodType</i>	16
3.1.6	<i>complexType DigestValueType</i>	17
3.1.7	<i>complexType HashedURIType</i>	18
3.1.8	<i>complexType HashType</i>	19
3.1.9	<i>complexType IntegrityManifestType</i>	20
3.1.10	<i>complexType SignerInfoType</i>	22
3.1.11	<i>complexType PlatformClassType</i>	23
3.1.12	<i>complexType TransformMethodType</i>	24
3.1.13	<i>complexType ValueType</i>	25
3.1.14	<i>complexType VendorIdType</i>	26
3.2	ELEMENTS	27
3.2.1	<i>element ComponentIDType/VendorID</i>	27
3.2.2	<i>element ComponentRefType/ComponentID</i>	28
3.2.3	<i>element ComponentRefType/ComponentIDREF</i>	29
3.2.4	<i>element IntegrityManifestType/ComponentID</i>	29
3.2.5	<i>element IntegrityManifestType/SignerInfo</i>	31
3.2.6	<i>element IntegrityManifestType/ConfidenceValue</i>	31
3.2.7	<i>element IntegrityManifestType/Collector</i>	32
3.2.8	<i>element IntegrityManifestType/TransformMethod</i>	33
3.2.9	<i>element IntegrityManifestType/DigestMethod</i>	33
3.2.10	<i>element IntegrityManifestType/Values</i>	34
3.2.11	<i>element IntegrityManifestType/AssertionInfo</i>	35
3.2.12	<i>element IntegrityManifestType/PlatformClass</i>	35
3.2.13	<i>element IntegrityManifestType/SubComponents</i>	36
3.2.14	<i>element SignerInfoType/SigningComponent</i>	36
3.2.15	<i>element VendorIdType/TcgVendorId</i>	37
3.2.16	<i>element VendorIdType/SmiVendorId</i>	37
3.2.17	<i>element VendorIdType/VendorGUID</i>	38
4	REFERENCES.....	39
5	APPENDIX A: XML SIGNATURE SCHEMA	40
5.1	COMPLEX TYPES	40
5.1.1	<i>complexType ds:CanonicalizationMethodType</i>	40
5.1.2	<i>complexType ds:DigestMethodType</i>	41

5.1.3 complexType ds:DSAKeyValue	42
5.1.4 complexType ds:KeyInfo	43
5.1.5 complexType ds:KeyValue	44
5.1.6 complexType ds:Manifest	44
5.1.7 complexType ds:Object	45
5.1.8 complexType ds:PGPData	45
5.1.9 complexType ds:Reference	46
5.1.10 complexType ds:RetrievalMethod	47
5.1.11 complexType ds:RSAKeyValue	47
5.1.12 complexType ds:SignatureMethod	48
5.1.13 complexType ds:SignatureProperties	48
5.1.14 complexType ds:SignatureProperty	49
5.1.15 complexType ds:Signature	49
5.1.16 complexType ds:SignatureValue	50
5.1.17 complexType ds:SignedInfo	50
5.1.18 complexType ds:SPKIData	51
5.1.19 complexType ds:Transforms	51
5.1.20 complexType ds:Transform	52
5.1.21 complexType ds:X509Data	52
5.1.22 complexType ds:X509IssuerSerial	53
5.2 SIMPLE TYPES	53
5.2.1 simpleType ds:CryptoBinary	53
5.2.2 simpleType ds:DigestValue	53
5.2.3 simpleType ds:HMACOutputLength	54
5.3 ELEMENTS	54
5.3.1 element ds:CanonicalizationMethod	54
5.3.2 element ds:DigestMethod	55
5.3.3 element ds:DigestValue	55
5.3.4 element ds:DSAKeyValue	56
5.3.5 element ds:KeyInfo	57
5.3.6 element ds:KeyName	57
5.3.7 element ds:KeyValue	58
5.3.8 element ds:Manifest	58
5.3.9 element ds:MgmtData	59
5.3.10 element ds:Object	59
5.3.11 element ds:PGPData	60
5.3.12 element ds:Reference	61
5.3.13 element ds:RetrievalMethod	61
5.3.14 element ds:RSAKeyValue	62
5.3.15 element ds:Signature	63
5.3.16 element ds:SignatureMethod	63
5.3.17 element ds:SignatureProperties	64
5.3.18 element ds:SignatureProperty	64
5.3.19 element ds:SignatureValue	65
5.3.20 element ds:SignedInfo	65
5.3.21 element ds:SPKIData	66
5.3.22 element ds:Transform	66
5.3.23 element ds:Transforms	67
5.3.24 element ds:X509Data	67
5.3.25 element ds:DSAKeyValue/P	68
5.3.26 element ds:DSAKeyValue/Q	68
5.3.27 element ds:DSAKeyValue/G	68
5.3.28 element ds:DSAKeyValue/Y	68
5.3.29 element ds:DSAKeyValue/J	69
5.3.30 element ds:DSAKeyValue/Seed	69
5.3.31 element ds:DSAKeyValue/PgenCounter	69

5.3.32 element ds:PGPDataType/PGPKeyID	69
5.3.33 element ds:PGPDataType/PGPKeyPacket.....	70
5.3.34 element ds:PGPDataType/PGPKeyPacket.....	70
5.3.35 element ds:RSAKeyValue/Modulus.....	70
5.3.36 element ds:RSAKeyValue/Exponent.....	70
5.3.37 element ds:SignatureMethodType/HMACOutputLength	71
5.3.38 element ds:SPKIDataType/SPKISexp.....	71
5.3.39 element ds:TransformType/XPath	71
5.3.40 element ds:X509DataType/X509IssuerSerial	72
5.3.41 element ds:X509DataType/X509SKI	72
5.3.42 element ds:X509DataType/X509SubjectName	72
5.3.43 element ds:X509DataType/X509Certificate	73
5.3.44 element ds:X509DataType/X509CRL.....	73
5.3.45 element ds:X509IssuerSerialType/X509IssuerName	73
5.3.46 element ds:X509IssuerSerialType/X509SerialNumber.....	73

1 Scope and Audience

This specification is integral to the TCG Infrastructure Working Group's (IWG) reference architecture, and is directly related to the TCG's Integrity Management Model. Specifically, the core integrity metadata XML schema defines the structure with which integrity information is communicated between entities.

Architects, designers, developers, and technologists interested in the development, deployment, and interoperation of trusted systems will find this document necessary in providing a specific mechanism for communicating integrity information.

The reader is directed to *IWG Integrity Management Architecture Part II* [1] for background and glossary terms.

1.1 Normative and Non-normative

This specification defines and documents an XML schema. A companion *.xsd* file contains machine readable expression of the XML schema definition. The XML in both *.xsd* file and this document should agree. If discrepancies are found, the *.xsd* file shall be regarded as normative.

All other documentation in this specification is normative.

Non-normative text is highlighted in gray. Alternatively, a large section of non-normative comment is called out explicitly in the descriptive text and terminates at the end of the section containing explicit declaration.

2 Introduction

The purpose of this document is to provide a detailed description of the TCG Infrastructure Working Group's core integrity metadata XML schema, hereafter referred to as the *core schema*. The core schema serves the purposes of:

- Defining the basic structure of XML documents responsible for communicating integrity metadata
- Defining XML data structures applicable to dependent, derived XML schemas

The TCG Integrity Management Model (defined in the *Platform Integrity Information Architecture*) identifies five stages of integrity metadata management: production, collection, communication, storage, and evaluation. The core schema is dedicated to integrity metadata *communication*: the transfer of integrity information from entities that collect it to those responsible for integrity information collation and evaluation.

With respect to the core schema, integrity metadata schemas are intentionally undefined. It is understood that XML integrity metadata documents will be specific to a particular domain of interpretation, hence will be extended using XML Schema extensibility options. Domain specific integrity metadata will be used to communicate:

- *Integrity values* – Atomic elements of system composition, expressed as a cryptographic hash over element attributes
- *Integrity assertions* – Enumerated statements of processes followed or claims made that reflect the quality of the identified component

It is the responsibility of each integrity domain to provide a derived XML schema in which a domain-specific integrity metadata schema is defined. The TCG may define a few generic schemas that use the same extensibility feature. The core schema is primarily responsible for defining a common structure for capturing integrity metadata elements that can be controlled by a change management process; dependent, derived XML schemas are responsible for defining the structure with which domain-specific definitions of integrity metadata are communicated.

2.1 Schema Version

The core schema's version number is defined using the `version` attribute of the schema's root-level `schema` element:

```
version="version_number"
```

This document refers to version 1.0.1 of the core schema.

2.2 Schema Namespace

The core schema's namespace is defined using the `targetNamespace` attribute of the schema's root-level `schema` element:

```
targetNamespace="namespace"
```

The schema's namespace reflects the schema version, and is currently defined as follows:

```
http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#
```

2.3 Dependent Schema Definitions

2.3.1 W3C XML Schema Syntax

The core schema relies upon data structures defined by the World Wide Web Consortium's (W3C) XML-Schema syntax. Consequently, the core schema imports the W3C's XML schema with the following namespace:

`http://www.w3.org/2001/XMLSchema`

The core schema associates the abovementioned schema with the “xs” namespace prefix.

2.3.2 W3C XML-Signature Syntax

The core schema relies upon data structures defined by the World Wide Web Consortium's (W3C) XML-Signature digital signature syntax. Consequently, the core schema imports the W3C's digital signature XML schema with the following namespace:

`http://www.w3.org/2000/09/xmldsig#`

The core schema associates the abovementioned schema with the “ds” namespace prefix.

The schema location for XML-Signature schema:

<http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd>

3 Core Integrity Schema

schema location: https://trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_Manifest_v1_0_1.xsd
 attribute form default: **unqualified**
 element form default: **qualified**
 targetNamespace: http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

3.1 Complex Types

Hyperlinks to Complex type Definitions

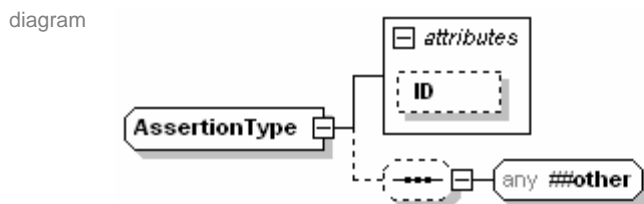
[AssertionType](#)
[ComponentIDType](#)
[ComponentRefType](#)
[ConfidenceValueType](#)
[DigestMethodType](#)
[DigestValueType](#)
[HashType](#)
[HashedURIType](#)
[IntegrityManifestType](#)
[SignerInfoType](#)
[PlatformClassType](#)
[TransformMethodType](#)
[ValueType](#)
[VendorIDType](#)

3.1.1 complexType AssertionType

3.1.1.1 Description

AssertionType consists of a record identifier and any other element containing assertions expressed in XML. Assertions are specific to a domain of interpretation, hence should be described using an applicable schema definition. AssertionType provides an extensibility feature for incorporating domain-specific assertions into integrity manifest and reporting structures. The TCG Security Qualities [5] schema is an example of an XML schema containing assertions.

3.1.1.2 Diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

properties abstract true

used by element [IntegrityManifestType/AssertionInfo](#)

attributes	Name	Type	Use	Default	Fixed
	ID	xs:ID			

3.1.1.3 Attribute Detail

Component	Description
ID	Globally unique record instance identifier. ID may be used to distinguish multiple instances of elements of type AssertionType. If the domain-specific schema defines an xs:ID identifier, it

should have the same value as ID.

3.1.1.4 XML

```
source <xs:complexType name="AssertionType" abstract="false">
  <xs:sequence minOccurs="0">
    <xs:any namespace="##other" processContents="lax"/>
  </xs:sequence>
  <xs:attribute name="ID" type="xs:ID"/>
</xs:complexType>
```

3.1.2 complexType ComponentIDType

3.1.2.1 Description

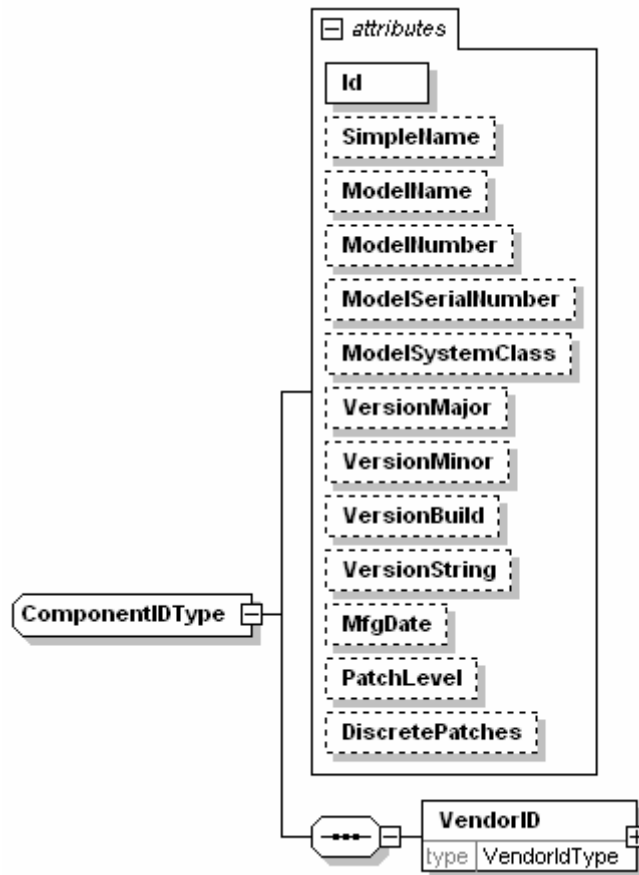
The ComponentIDType complex type represents an atomic integrity element identifying a particular program code or logic (hereafter referred to as a component). The identifier does not try to distinguish multiple instances of the same code or logic. For example, the ComponentType complex type is used within a TCG Reference Manifest Schema Specification [2] to represent application integrity values derived from a baseline build image. ComponentIDType is also used by the Snapshot complex type in the TCG Integrity Report Schema Specification [3] to capture actual measurements of components that may be extended into PCRs.

ComponentIDType is a set of attributes accommodating a wide range of change management schemes that when combined uniquely identifies a change-controlled item. The package, program code or logic under change management will have processes for ensuring integrity of its image. VendorID must uniquely identify an entity that maintains the change management process. If the VendorID is a GUID, then it is assumed the change management process owner can be obtained some other way (e.g. via database lookup using the GUID as a database key).

Most attributes are optional to ensure applicability across a variety of change management systems. However the vendorID element must be unique with respect to all possible vendors.

3.1.2.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

children [VendorID](#)

used by elements [IntegrityManifestType/Collector](#) [ComponentRefType/ComponentID](#)
[IntegrityManifestType/ComponentID](#) [SignerInfoType/SigningComponent](#)

attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	required		
	SimpleName	xs:normalizedString	optional		
	ModelName	xs:normalizedString	optional		
	ModelNumber	xs:normalizedString	optional		
	ModelSerialNumber	xs:normalizedString	optional		
	ModelSystemClass	xs:normalizedString	optional		
	VersionMajor	xs:integer	optional		
	VersionMinor	xs:integer	optional		
	VersionBuild	xs:integer	optional		
	VersionString	xs:normalizedString	optional		
	MfgDate	xs:dateTime	optional		
	PatchLevel	xs:normalizedString	optional		
	DiscretePatches	xs:NMTOKENS	optional		

3.1.2.3 Attribute Detail

Component	Description
Id	Record instance identifier – recommended globally unique
SimpleName	String-ified version information for simple compare operations
ModelName	Model name with which the component is marketed
ModelNumber	Alphanumeric model number with which the component is identified

ModelSerialNumber	Alphanumeric model serial number with which the component is identified
ModelSystemClass	Vendor-specific system type or environment with which the component is associated
VersionMajor	Major version number of the component
VersionMinor	Minor version number of the component
VersionBuild	Build number of the component
VersionString	String with which the component's version may be identified
BuildDate	Date on which the component was manufactured
PatchLevel	Patch level of the component
DiscretePatches	Token strings enumerating each discrete patch that has been applied to the component; that is not also represented by PatchLevel or other attributes in ComponentType

3.1.2.4 XML

```

source <xs:complexType name="ComponentIDType">
  <xs:sequence>
    <xs:element name="VendorID" type="VendorIDType"/>
  </xs:sequence>
  <xs:attribute name="Id" type="xs:ID" use="required"/>
  <xs:attribute name="SimpleName" type="xs:normalizedString" use="optional"/>
  </xs:attribute>
  <xs:attribute name="ModelName" type="xs:normalizedString" use="optional"/>
  </xs:attribute>
  <xs:attribute name="ModelNumber" type="xs:normalizedString" use="optional"/>
  </xs:attribute>
  <xs:attribute name="ModelSerialNumber" type="xs:normalizedString" use="optional"/>
  </xs:attribute>
  <xs:attribute name="ModelSystemClass" type="xs:normalizedString" use="optional"/>
  </xs:attribute>
  <xs:attribute name="VersionMajor" type="xs:integer" use="optional"/>
  <xs:attribute name="VersionMinor" type="xs:integer" use="optional"/>
  <xs:attribute name="VersionBuild" type="xs:integer" use="optional"/>
  <xs:attribute name="VersionString" type="xs:normalizedString" use="optional"/>
  <xs:attribute name="MfgDate" type="xs:dateTime" use="optional"/>
  </xs:attribute>
  <xs:attribute name="PatchLevel" type="xs:normalizedString" use="optional"/>
  <xs:attribute name="DiscretePatches" type="xs:NMTOKENS" use="optional"/>
</xs:complexType>

```

3.1.3 complexType ComponentRefType

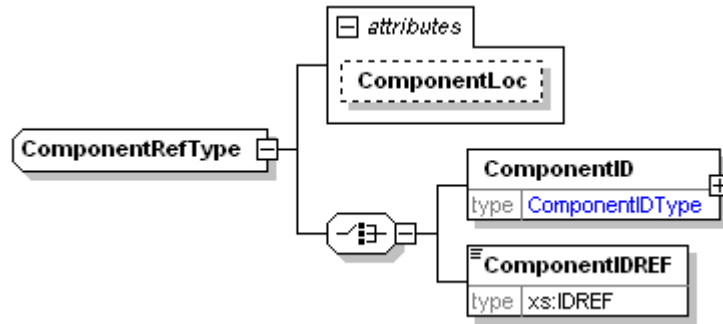
3.1.3.1 Description

The ComponentRefType complexType is used to refer to components in other locations, documents or repositories. There are three references that are useful in identifying a component.

- ComponentIDREF – a reference within an XML document.
- ComponentLoc – a reference to a web resource.
- ComponentID element – a ComponentIDType structure whose attributes may be used to perform a database query.

3.1.3.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

children [ComponentID](#) [ComponentIDREF](#)

used by elements [IntegrityManifestType/Collector](#) [SignerInfoType/SigningComponent](#)
[IntegrityManifestType/SubComponents](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	ComponentLoc	xs:anyURI	optional			

3.1.3.3 Attribute Detail

Component	Description
ComponentLoc	A URI referencing a document containing an element of type ComponentIDType

3.1.3.4 XML

```

source <xs:complexType name="ComponentRefType">
  <xs:choice>
    <xs:element name="ComponentID">
      <xs:complexType>
        <xs:complexContent>
          <xs:extension base="ComponentIDType"/>
        </xs:complexContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="ComponentIDREF" type="xs:IDREF"/>
  </xs:choice>
  <xs:attribute name="ComponentLoc" type="xs:anyURI" use="optional"/>
</xs:complexType>
  
```

3.1.4 complexType ConfidenceValueType

3.1.4.1 Description

The ConfidenceValueType complex type represents the level of confidence (hereafter referred to as a *confidence value*) with which a numerical representation of trust may be given to the assertion with which it is associated. For example, the ConfidenceValueType complex type is applied within the IntegrityMetadataType complex type to identify the level of confidence with which to trust a single collection of integrity metadata.

Further examples of assertions that may be assigned confidence values include integrity assertions and integrity values (represented using IntegrityAssertionType and IntegrityValueType complex types, respectively).

A confidence value is a rational number. Two values are integral to the calculation of a confidence value:

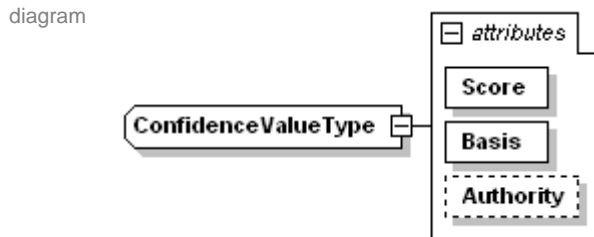
- **Score** – The confidence points given to an assertion. The score must be greater than or equal to 0, and less than or equal to the specified basis.

- *Basis* – The maximum number of confidence *points* that may be given to an assertion. The basis must be an integer greater than 0.
- *Authority* – The entity that defines criteria for establishing the Basis is optionally provided in the form of a URI.

An assertion’s confidence value is calculated by dividing its score into its basis. For example, given a basis of 100, an assertion whose score is 95 will receive a confidence value of 0.95.

Cooperation between producers and consumers of documents containing ConfidenceValue may establish scoring conventions such that all have a common frame of understanding. This specification does not define such a convention. However, a URI reference to an entity that defines such criteria can be provided.

3.1.4.2 Diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

used by element [element](#)
IntegrityManifestType/Confidence Value

attributes	Name	Type	Use	Default	Fixed	Annotation
	Score	xs:integer	required			
	Basis	xs:integer	required			
	Authority	xs:anyURI	optional			

3.1.4.3 Attribute Detail

Component	Description
Score	Confidence points given to an assertion. Greater than or equal to 0, and less than or equal to the specified basis.
Basis	Maximum number of confidence points that may be given to an assertion. Greater than 0.
Authority	Reference to an authoritative source that defines criteria for establishing the Basis value.

3.1.4.4 XML

```
source <xs:complexType name="ConfidenceValueType">
  <xs:attribute name="Score" type="xs:integer" use="required"/>
  <xs:attribute name="Basis" type="xs:integer" use="required"/>
  <xs:attribute name="Authority" type="xs:anyURI" use="optional"/>
</xs:complexType>
```

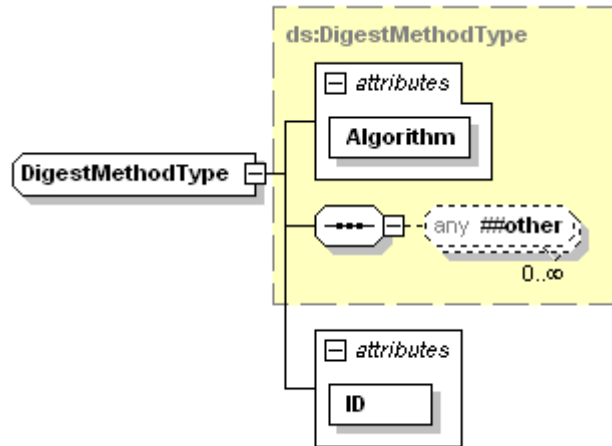
3.1.5 complexType DigestMethodType

3.1.5.1 Description

DigestMethodType identifies cryptographic hash algorithms. There may be several different digest algorithms used when generating a Reference Integrity Measurement Manifest (RIMM) structure. Instances of elements of type DigestMethodType are referenced using the ID attribute.

3.1.5.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type extension of [ds:DigestMethodType](#)

properties base [ds:DigestMethodType](#)

used by element [IntegrityManifestType/DigestMethod](#)

attributes	Name	Type	Use	Default	Fixed
	Algorithm	xs:anyURI	required		
	Id	xs:ID	required		

3.1.5.3 Attribute Detail

Component	Description
Id	Document unique record instance identifier. Id is used in other parts of the document to reference instances of hash algorithm identifiers.
Algorithm	xs:anyURI defining a well-known digest algorithm. SHA-1 must be implemented as a minimum for interoperability. (e.g. http://www.w3.org/2000/09/xmlsig#sha1)
any##other	Defines other digest algorithms not available through the Algorithm attribute.

3.1.5.4 XML

```
source <xs:complexType name="DigestMethodType">
  <xs:complexContent>
    <xs:extension base="ds:DigestMethodType">
      <xs:attribute name="Id" type="xs:ID" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

3.1.6 complexType DigestValueType

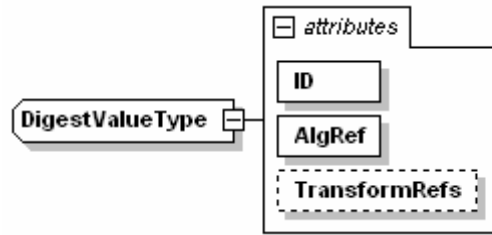
3.1.6.1 Description

DigestValueType is derived by extension from XML Signature schema. It is used as a convenience for deriving other types (such as HashType) that may be extended or restricted with other attributes.

DigestValueType is a xs:base64binary containing the result of a cryptographic digest operation.

3.1.6.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type extension of [ds:DigestValueType](#)

properties base ds:DigestValueType

used by element [HashedURIType/UriHash](#)
complexType [HashType](#)

attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	required		
	AlgRef	xs:IDREF	required		
	TransformRefs	xs:IDREFS			

3.1.6.3 Attribute Detail

Component	Description
Id	Document unique record instance identifier. Id is used to reference instances of hash algorithms that may be in use by a bounding document.
AlgRef	AlgRef refers to a hash algorithm as defined by DigestMethodType .
TransformRefs	Refers to transformation functions defined by TransformMethod elements of type TransformMethodType .

3.1.6.4 XML

```

source <xs:complexType name="DigestValueType">
  <xs:simpleContent>
    <xs:extension base="ds:DigestValueType">
      <xs:attribute name="Id" type="xs:ID" use="required"/>
      <xs:attribute name="AlgRef" type="xs:IDREF" use="required">
    </xs:attribute>
    <xs:attribute name="TransformRefs" type="xs:IDREFS">
    </xs:attribute>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
  
```

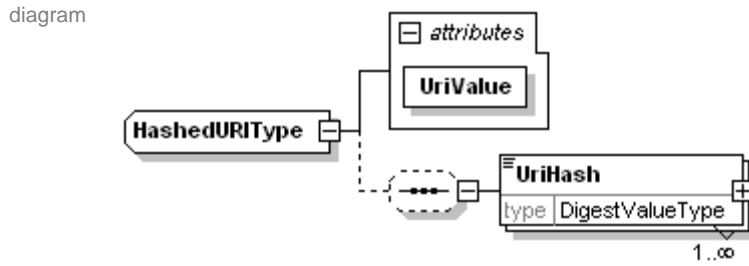
3.1.7 complexType HashedURIType

3.1.7.1 Description

The HashedURIType complex type contains a URI reference and a hash, UriHash, of the object that the URI refers to. The UriHash, if included, contains 1 or more hash values. If multiple hash algorithms are in use, it may be desirable to include multiple UriHash values. The AlgRef attribute of UriHash identifies the hash algorithms used.

The TransformRefs attributes, also in UriHash, identifies any algorithms used to measure the object referenced by UriValue.

3.1.7.2 Diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

children [UriHash](#)

attributes	Name	Type	Use	Default	Fixed
	UriValue	xs:anyURI	required		Fixed

3.1.7.3 Attribute Detail

Component	Description
UriValue	An xs:anyUri that refers to a data object whose integrity can be assessed using UriHash values.

3.1.7.4 XML

```

source <xs:complexType name="HashedURType">
  <xs:sequence minOccurs="0">
    <xs:element name="UriHash" type="DigestValueType" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="UriValue" type="xs:anyURI" use="required"/>
</xs:complexType>

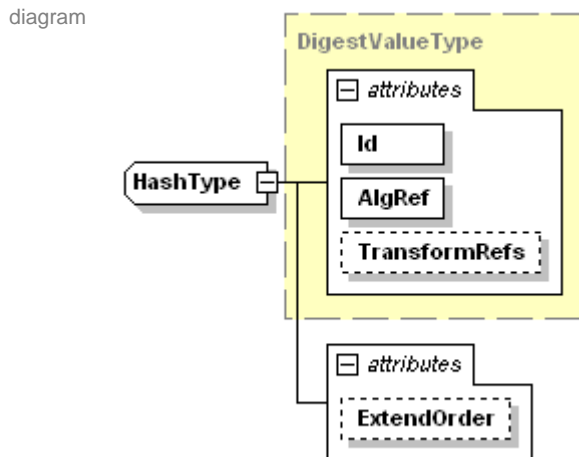
```

3.1.8 complexType HashType

3.1.8.1 Description

HashType extends DigestValueType appending the ExtendOrder attribute. ExtendOrder is used to identify the sequence in which documents are extended (hashed). AlgRef identifies the hash algorithm used. TransformRefs identifies transformation algorithms that are applied to the document prior to applying the hash algorithm.

3.1.8.2 Diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type extension of [DigestValueType](#)

properties base DigestValueType

attributes	Name	Type	Use	Default	Fixed	Annotation
	Id	xs:ID	required			
	AlgRef	xs:IDREF	required			
	TransformRefs	xs:IDREFS	optional			
	ExtendOrder	xs:IDREFS	optional			

3.1.8.3 Attribute Detail

Component	Description
ExtendOrder	ExtendOrder contains an ordered list of xs:IDREF values. Values at the beginning of the list occur before values at the end. Therefore, the first entry in the list would be the first value extended, the last entry would be the last value extended.

3.1.8.4 XML

```
source <xs:complexType name="HashType">
  <xs:simpleContent>
    <xs:extension base="DigestValueType">
      <xs:attribute name="ExtendOrder" type="xs:IDREFS"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

3.1.9 complexType IntegrityManifestType

3.1.9.1 Description

The IntegrityManifestType complex type can be used to describe integrity attributes of program code, discrete logic and packages of components. Any element that can be placed under change control is a candidate for being described using IntegrityManifestType complex type.

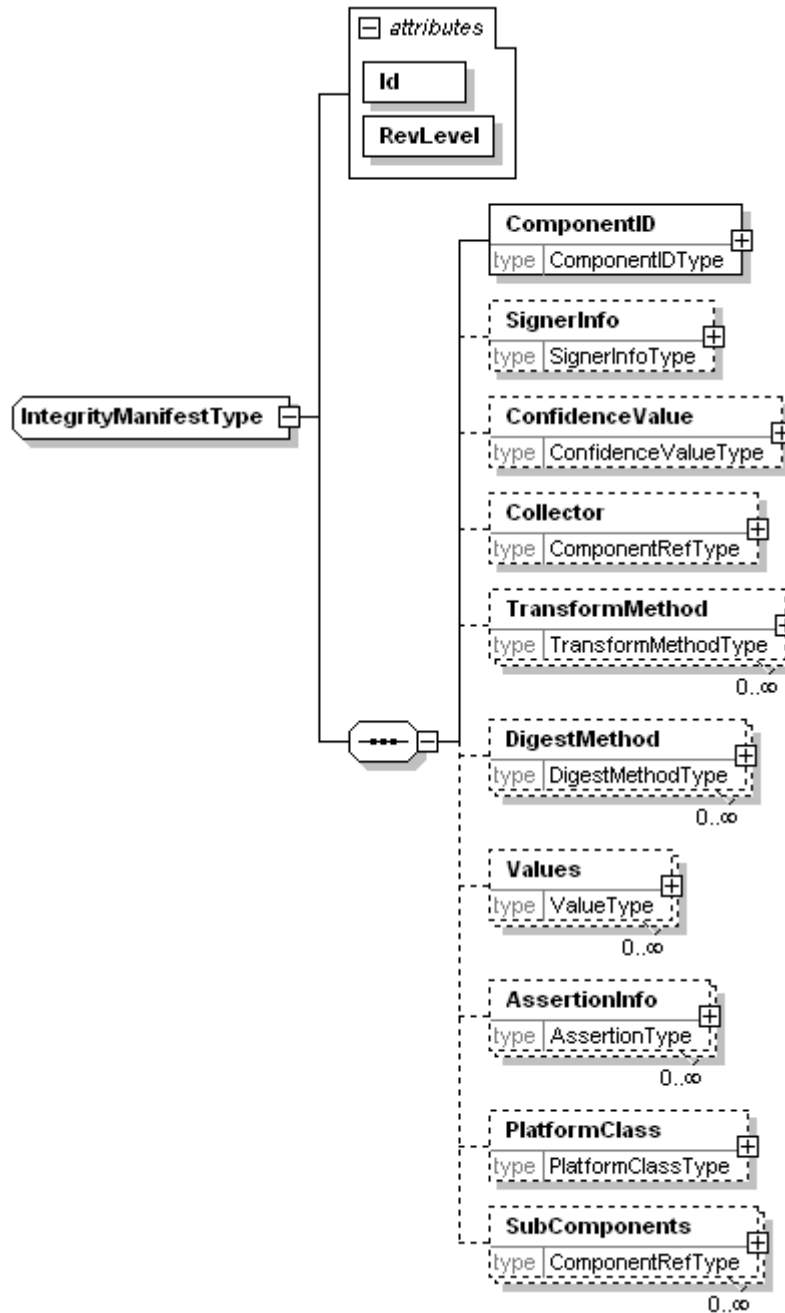
Elements of IntegrityManifestType include:

- ComponentID – is a unique complex identifier linking the component to a change management process.
- SignerInfo – is a signature over the Integrity Manifest. It includes information about the entity that produced the signature. A single signature may be applied.
- ConfidenceValue – contains a score as a single value aggregating several criteria for establishing a degree of assurance (or trust) that the values and assertions made by the manifest are correct.
- Collector – is a reference to the utility (component) used to construct an integrity manifest. A manifest for the Collector may be separately obtained for information relating to the environment that produced *this* integrity manifest. A single collector may be referenced.
- TransformMethod – contains algorithm identifiers for transforms that may have been applied prior to applying a digest method. Multiple transformation methods may be defined.
- DigestMethod – contains algorithm identifiers for hash algorithms that are used to compute message digests. Multiple digest methods may be defined.
- Values – contains integrity measurements (message digests) that pertain to *this* component. It is reasonable (even desirable) that schemas capturing domain specific structure should incorporate a composite hash structure that is incorporated into an instantiation of Integrity Manifest with an element of type HashType. Multiple instances of Values elements may be supplied.

- AssertionInfo – contains domain specific description of attributes affecting quality, assurance or reliability assessments, but where it isn't possible for measurement engines to collect *actual* values. Multiple instances of AssertionInfo elements may be supplied.
- PlatformClass – identifies the type of platform that integrity values pertain to. In particular, the methodology for PCR allocation is specified by platform specific specifications.
- SubComponents – are references to finer grain components that make up *this* component.

3.1.9.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#
 properties abstract true

children	ComponentID SignerInfo ConfidenceValue Collector TransformMethod DigestMethod Values AssertionInfo PlatformClass SubComponents															
attributes	<table border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Use</th> <th>Default</th> <th>Fixed</th> </tr> </thead> <tbody> <tr> <td>Id</td> <td>xs:ID</td> <td>required</td> <td></td> <td></td> </tr> <tr> <td>RevLevel</td> <td>xs:integer</td> <td>required</td> <td></td> <td></td> </tr> </tbody> </table>	Name	Type	Use	Default	Fixed	Id	xs:ID	required			RevLevel	xs:integer	required		
Name	Type	Use	Default	Fixed												
Id	xs:ID	required														
RevLevel	xs:integer	required														

3.1.9.3 Attribute Detail

Component	Description
Id	Globally unique record instance identifier. Id may be used by external systems, documents and <i>this</i> document to reference an instance of a component structure.
RevLevel	RevLevel is a revision number (increment for more recent revision) to distinguish revisions of an integrity manifest structure. RevLevel applies to instances of integrity manifest structures having the same Id value.

3.1.9.4 XML

```

source <xs:complexType name="IntegrityManifestType" abstract="true">
  <xs:sequence>
    <xs:element name="ComponentID" type="ComponentIDType"/>
    <xs:element name="SignerInfo" type="SignerInfoType" minOccurs="0"/>
    <xs:element name="ConfidenceValue" type="ConfidenceValueType" minOccurs="0"/>
    <xs:element name="Collector" type="ComponentRefType" minOccurs="0"/>
    <xs:element name="TransformMethod" type="TransformMethodType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="DigestMethod" type="DigestMethodType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="Values" type="ValueType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="AssertionInfo" type="AssertionType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="PlatformClass" type="PlatformClassType" minOccurs="0"/>
    <xs:element name="SubComponents" type="ComponentRefType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="Id" type="xs:ID" use="required"/>
  <xs:attribute name="RevLevel" type="xs:integer" use="required"/>
</xs:complexType>

```

3.1.10 complexType SignerInfoType

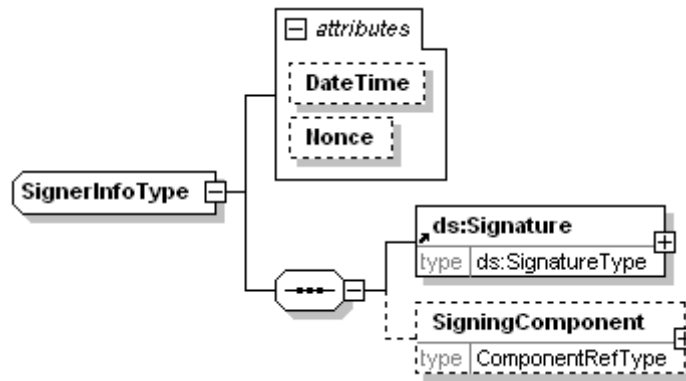
3.1.10.1 Description

Each SignerInfoType structure has the following structure:

- *Digital signature* – Contains the digital signature resulting from signing Integrity Metadata elements. Authority to sign is determined in large part by verifier policies. The structure is represented by the `ds:Signature` element.
- *Confidence value* – Identifies the level of confidence with which trust may be given to the integrity information assumed within the structure. Represented by the `ConfidenceValue` element.
- *SigningComponent* – Identifies the program code or logic responsible for compiling, measuring and formatting, the integrity information contained within the structure. Represented by a `ComponentID` element.

3.1.10.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

children [ds:Signature](#) [SigningComponent](#)

used by element [IntegrityManifestType/SignerInfo](#)

attributes	Name	Type	Use	Default	Fixed
	DateTime	xs:dateTime			
	Nonce	xs:base64Binary			

3.1.10.3 Attribute Detail

Component	Description
DateTime	The date and time that the signature was generated. This attribute, if specified, must be included in the signature calculation.
Nonce	A value obtained from a remote party that is included with a signature to guarantee freshness and to avoid replay attack. This attribute, if specified, must be included in the signature calculation.

3.1.10.4 XML

```

source <xs:complexType name="SignerInfoType">
  <xs:sequence>
    <xs:element ref="ds:Signature"/>
    <xs:element name="ConfidenceValue" type="ConfidenceValueType" minOccurs="0"/>
    <xs:element name="SigningComponent" type="ComponentIDType" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="DateTime" type="xs:dateTime">
    <xs:annotation>
      <xs:documentation>When signature was applied</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="Nonce" type="xs:base64Binary"/>
</xs:complexType>

```

3.1.11 complexType PlatformClassType

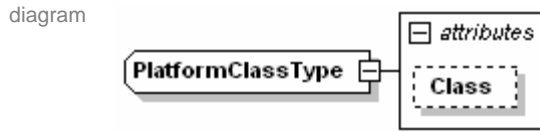
3.1.11.1 Description

PlatformClassType enumerates platform classifications as determined by the Trusted Computing Group (TCG). Platform classifications can be used to apply platform specific interpretations of integrity values and quality assertions.

PlatformClassType associates a component to a platform family or classification. The association can be used to qualify usage conventions associated with digest creation, the number of allowable digests and semantics for digest association with other components in a system.

A vendor specific classification may be provided by defining a platform identifier based on a vendor specific namespace.

3.1.11.2 Diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

used by element [IntegrityManifestType/PlatformClass](#)

attributes	Name	Type	Use	Default	Fixed
	Class	xs:anyURI	optional		

3.1.11.3 Attribute Detail

Component	Description
	A vendor specific platform classification. If the URI does not unambiguously determine the vendor, the VendorID of the ComponentID for the integrity manifest is taken to be the vendor.
Class	<p>TCG defines platform class URIs. They can be used to identify the TCG platform-specific specification that applies to the platform. In particular it can be used to distinguish how Trusted Platform Module (TPM) resources, such as PCRs can be interpreted.</p> <p>TCG defined Class Identifiers:</p> <p>http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#PC_CLIENT_X86_BIOS Signifies an x86 based system with BIOS based firmware.</p> <p>http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#PC_CLIENT_X86_EFI Signifies an x86 based system with EFI based firmware.</p>

3.1.11.4 XML

```

source <xs:complexType name="PlatformClassType">
  <xs:attribute name="Class" type="xs:anyURI" use="optional"/>
</xs:complexType>
  
```

3.1.12 complexType TransformMethodType

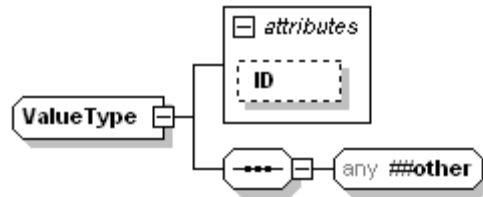
3.1.12.1 Description

The TransformMethodType is used to define an element that identifies a transformation algorithm to be applied prior to a hash computation operation.

The Id attribute is used by other elements that reference one or more transformation algorithms.

3.1.13.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

used by element [IntegrityManifestType/Values](#)

attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	optional		

3.1.13.3 Attribute Detail

Component	Description
Id	Record instance identifier of a child element whose schema definition is not in the current namespace. The Id is unique to the parent XML document.

3.1.13.4 XML

```
source <xs:complexType name="ValueType">
  <xs:sequence>
    <xs:any namespace="##other" processContents="lax"/>
  </xs:sequence>
  <xs:attribute name="Id" type="xs:ID"/>
</xs:complexType>
```

3.1.14 complexType VendorIdType

3.1.14.1 Description

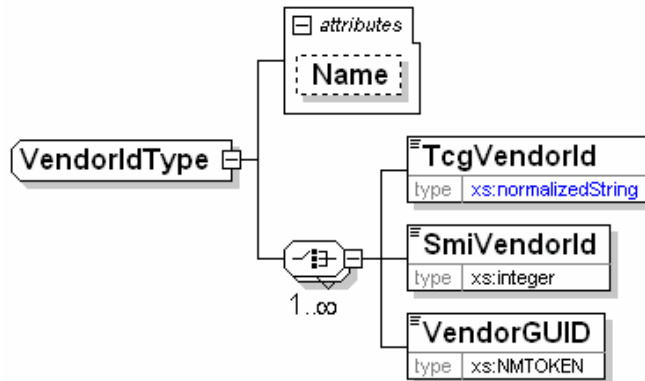
The VendorIdType is used to uniquely identify a vendor, manufacturer or other entity. There are two elements (SmiVendorId and TcgVendorId) that have managed number spaces ensuring uniqueness. VendorGUID uniqueness is derived algorithmically.

Only one form of VendorID element is required by the choice. More than one VendorID elements may be specified.

A familiar name can be specified, but should not be used to establish uniqueness properties.

3.1.14.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

children [TcgVendorId](#) [SmiVendorId](#) [VendorGUID](#)

used by element [ComponentIDType/VendorID](#)

attributes	Name	Type	Use	Default	Fixed
	Name	xs:string	optional		Fixed

3.1.14.3 Attribute Detail

Component	Description
Name	Familiar name associated with the component manufacturer or vendor

3.1.14.4 XML

```
source <xs:complexType name="VendorIdType">
  <xs:annotation>
    <xs:documentation>Identifies a vendor</xs:documentation>
  </xs:annotation>
  <xs:choice maxOccurs="unbounded">
    <xs:element name="TcgVendorId" type="xs:integer" minOccurs="0"/>
    <xs:element name="SmiVendorId" type="xs:integer" minOccurs="0"/>
    <xs:element name="VendorGUID" type="xs:NMTOKEN" minOccurs="0"/>
  </xs:choice>
  <xs:attribute name="Name" type="xs:string"/>
</xs:complexType>
```

3.2 Elements

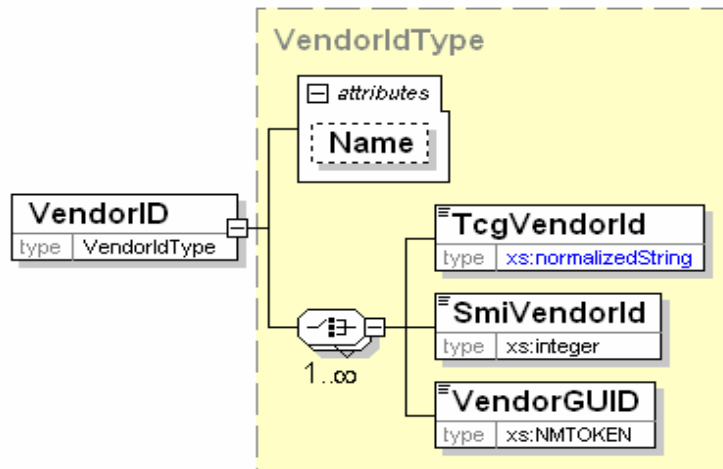
3.2.1 element ComponentIDType/VendorID

3.2.1.1 Description

The `VendorType` complex type represents a vendor, or party responsible for developing or distributing a component. For example, the `VendorType` complex type is applied within the `ComponentType` complex type (as the data type of element `Vendor`) to represent the vendor responsible for the component.

3.2.1.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type [VendorIdType](#)

properties isRef 0
content complex

children	TcgVendorId SmiVendorId VendorGUID				
attributes	Name Name	Type xs:string	Use	Default	Fixed

3.2.1.3 Attribute Detail

Component	Description
Name	Familiar name associated with the component manufacturer or vendor

3.2.1.4 XML

source `<xs:element name="VendorID" type="VendorIDType"/>`

3.2.2 element ComponentRefType/ComponentID

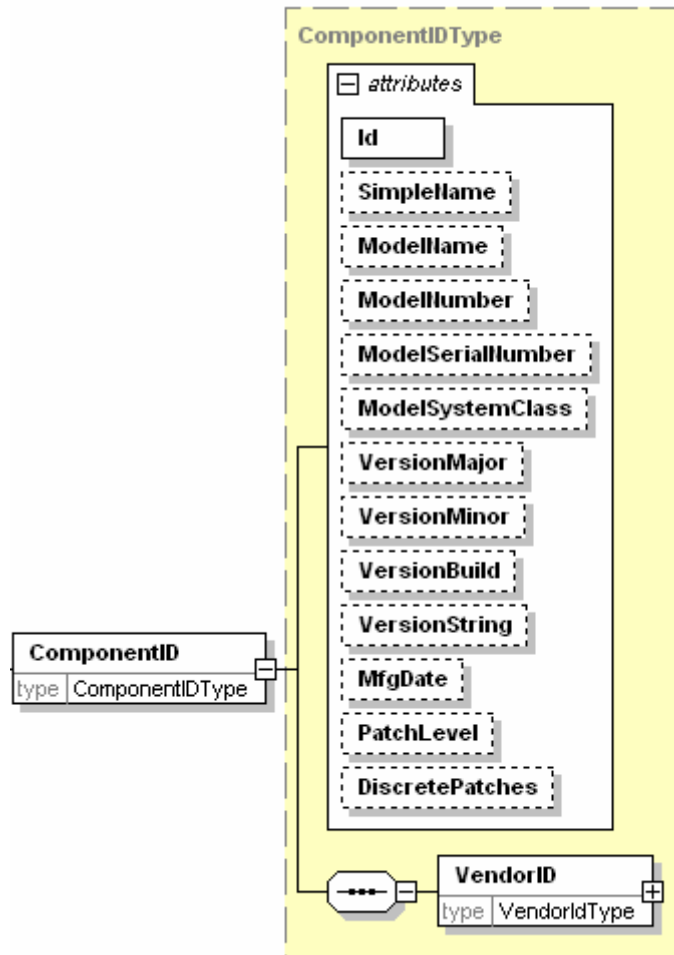
3.2.2.1 Description

The ComponentID element of a ComponentRefType contains attributes and VendorID element useful in identifying dependent or sub-components in a tree of components.

If used in conjunction with a component repository, attribute values and VendorID can be used to construct database queries that return records containing additional details relating to a component.

3.2.2.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type	ComponentIDType				
properties	isRef	0			
	content	complex			
children	VendorID				
attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	required		
	SimpleName	xs:normalizedString	optional		
	ModelName	xs:normalizedString	optional		
	ModelNumber	xs:normalizedString	optional		
	ModelSerialNumber	xs:normalizedString	optional		
	ModelSystemClass	xs:normalizedString	optional		
	VersionMajor	xs:integer	optional		
	VersionMinor	xs:integer	optional		
	VersionBuild	xs:integer	optional		
	VersionString	xs:string	optional		
	MfgDate	xs:dateTime	optional		
	PatchLevel	xs:normalizedString	optional		
	DiscretePatches	xs:NMTOKENS	optional		

3.2.2.3 XML

source `<xs:element name="ComponentID" type="ComponentIDType"/>`

3.2.3 element ComponentRefType/ComponentIDREF

3.2.3.1 Description

ComponentIDREF element is a reference to a ComponentID within the current document.

3.2.3.2 Diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type **xs:IDREF**

properties isRef 0
content simple

3.2.3.3 XML

source `<xs:element name="ComponentIDREF" type="xs:IDREF"/>`

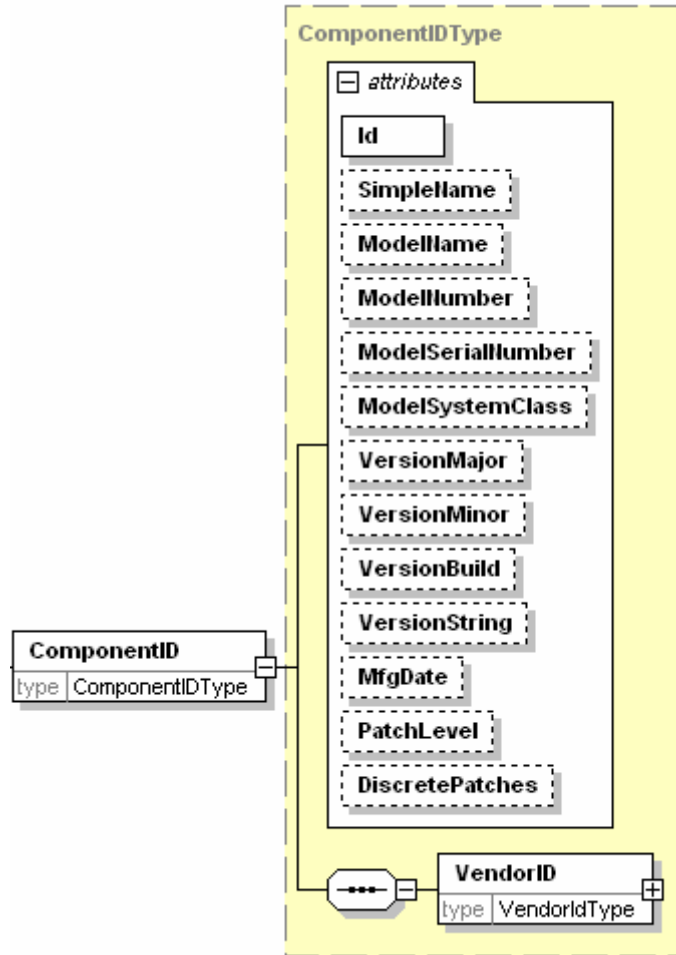
3.2.4 element IntegrityManifestType/ComponentID

3.2.4.1 Description

There is a single ComponentID element in an Integrity Manifest.

3.2.4.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type [ComponentIDType](#)

properties isRef 0
content complex

children [VendorID](#)

attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	required		
	SimpleName	xs:normalizedString	optional		
	ModelName	xs:normalizedString	optional		
	ModelNumber	xs:normalizedString	optional		
	ModelSerialNumber	xs:normalizedString	optional		
	ModelSystemClass	xs:normalizedString	optional		
	VersionMajor	xs:integer	optional		
	VersionMinor	xs:integer	optional		
	VersionBuild	xs:integer	optional		
	VersionString	xs:normalizedString	optional		
	MfgDate	xs:dateTime	optional		
	PatchLevel	xs:normalizedString	optional		
	DiscretePatches	xs:NMTOKENS	optional		

3.2.4.3 XML

source `<xs:element name="ComponentID" type="ComponentIDType"/>`

3.2.5 element IntegrityManifestType/SignerInfo

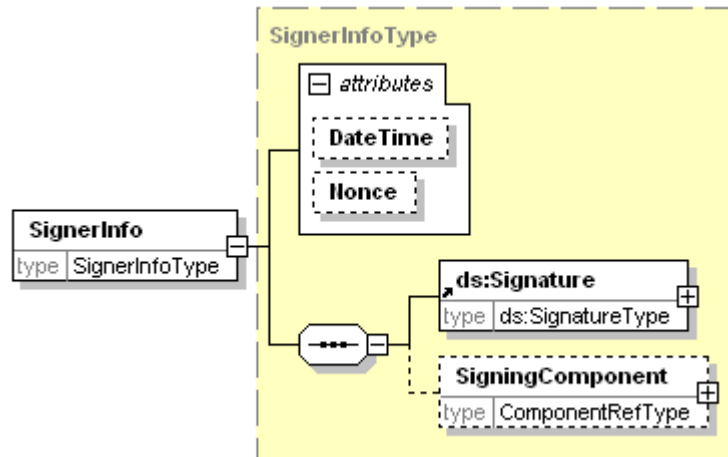
3.2.5.1 Description

The SignerInfo element contains a single signature over the Integrity Manifest. The signer may provide a confidence value and reference the component used to apply the signature.

The signature may also include a timestamp supplied by the signer or a nonce supplied by a verifier.

3.2.5.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type [SignerInfoType](#)

properties isRef 0
content complex

children [ds:Signature](#) [ConfidenceValue](#) [SigningComponent](#)

attributes	Name	Type	Use	Default	Fixed
	DateTime	xs:dateTime			
	Nonce	xs:base64Binary			

3.2.5.3 XML

source `<xs:element name="SignerInfo" type="SignerInfoType" minOccurs="0"/>`

3.2.6 element IntegrityManifestType/ConfidenceValue

3.2.6.1 Description

The ConfidenceValue element is a score given to the signed manifest describing the level of trust the signer has attributed to integrity values included in the signature.

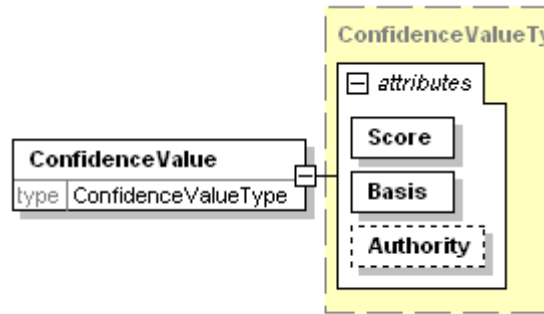
If the signer determines that confidence can be described in terms of levels and there are four possible levels then the first level could have a score of (1) with a basis of (4). Alternatively, a score of (25) would have a basis of (100).

If specified, this value must be included in the signature computation.

Basis values MUST be greater than 0.

3.2.6.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type [ConfidenceValueType](#)

properties isRef 0
content complex

attributes	Name	Type	Use	Default	Fixed
	Score	xs:integer	required		
	Basis	xs:integer	required		
	Authority	xs:anyURI	optional		

3.2.6.3 XML

source `<xs:element name="ConfidenceValue" type="ConfidenceValueType" minOccurs="0"/>`

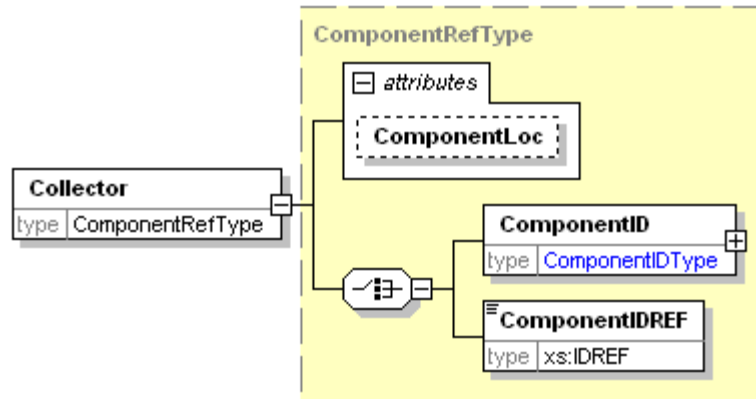
3.2.7 element IntegrityManifestType/Collector

3.2.7.1 Description

The Collector element contains information about the component used to construct the integrity manifest. If the signerInfo/SigningComponent element is the same as the Collector element, the Collector element may be omitted.

3.2.7.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type [ComponentRefType](#)

properties isRef 0
content complex

children [ComponentID](#) [ComponentIDREF](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	ComponentLoc	xs:anyURI	optional			

3.2.7.3 XML

```
source <xs:element name="Collector" type="ComponentIDType" minOccurs="0" />
```

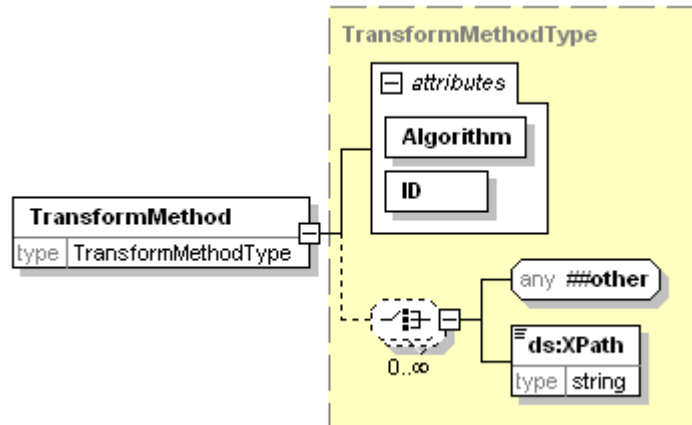
3.2.8 element IntegrityManifestType/TransformMethod

3.2.8.1 Description

The TransformMethod element identifies a filtering algorithm applied prior to generating a digest value.

3.2.8.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type [TransformMethodType](#)

properties isRef 0
content complex

children [ds:XPath](#)

attributes	Name	Type	Use	Default	Fixed
	Algorithm	xs:anyURI	required		
	Id	xs:ID	required		

3.2.8.3 XML

```
source <xs:element name="TransformMethod" type="TransformMethodType" minOccurs="0" maxOccurs="unbounded"/>
```

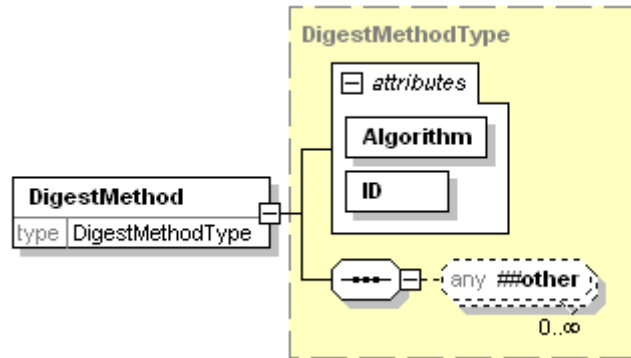
3.2.9 element IntegrityManifestType/DigestMethod

3.2.9.1 Description

The DigestMethod element is defined by the DigestMethodType complex type.

3.2.9.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type [DigestMethodType](#)

properties	isRef	0	content	complex	Type	Use	Default	Fixed
attributes	Name		Algorithm		xs:anyURI	required		
	Id				xs:ID	required		

3.2.9.3 XML

source `<xs:element name="DigestMethod" type="DigestMethodType" minOccurs="0" maxOccurs="unbounded"/>`

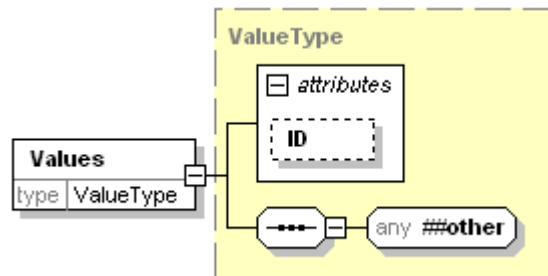
3.2.10 element IntegrityManifestType/Values

3.2.10.1 Description

The Values element in IntegrityManifestType is defined by ValueType complex type.

3.2.10.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type [ValueType](#)

properties	isRef	0	content	complex	Type	Use	Default	Fixed
attributes	Name							
	Id				xs:ID			

3.2.10.3 XML

source `<xs:element name="Values" type="ValueType" minOccurs="0" maxOccurs="unbounded"/>`

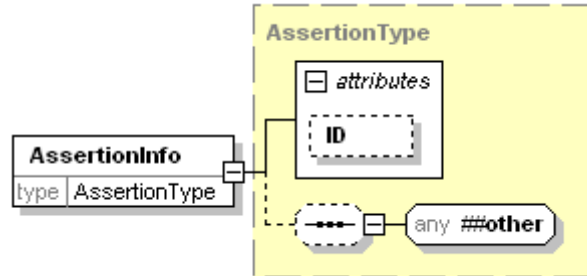
3.2.11 element IntegrityManifestType/AssertionInfo

3.2.11.1 Description

The AssertionInfo element in IntegrityManifestType is defined by AssertionInfoType complex type.

3.2.11.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type [AssertionType](#)

properties isRef 0
content complex

attributes	Name	Type	Use	Default	Fixed
Id	Id	xs:ID			

3.2.11.3 XML

source `<xs:element name="AssertionInfo" type="AssertionType" minOccurs="0" maxOccurs="unbounded"/>`

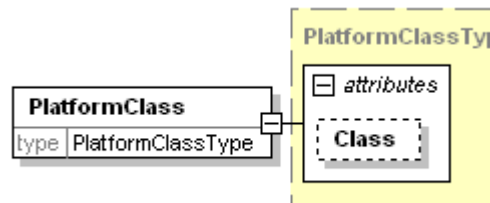
3.2.12 element IntegrityManifestType/PlatformClass

3.2.12.1 Description

The PlatformClass element in IntegrityManifestType is of type PlatformClassType.

3.2.12.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type [PlatformClassType](#)

properties isRef 0
content complex

attributes	Name	Type	Use	Default	Fixed
Class	Class	xs:anyURI	optional		

3.2.12.3 XML

source `<xs:element name="PlatformClass" type="PlatformClassType" minOccurs="0"/>`

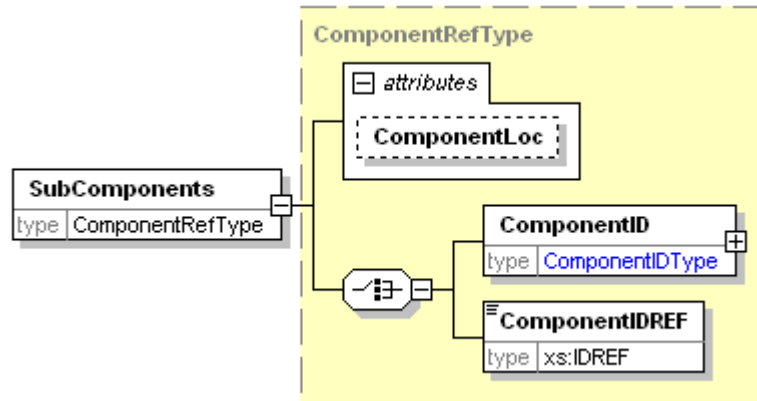
3.2.13 element IntegrityManifestType/SubComponents

3.2.13.1 Description

The SubComponents element identifies components of a system that are a decomposition of *this* component. An arbitrary nesting of subcomponents can be described if the referenced subcomponent is itself an element of type IntegrityManifestType.

3.2.13.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type [ComponentRefType](#)

properties isRef 0
content complex

children [ComponentID](#) [ComponentIDREF](#)

attributes	Name	Type	Use	Default	Fixed	Annotation
	ComponentLoc	xs:anyURI	optional			

3.2.13.3 XML

source `<xs:element name="SubComponents" type="ComponentRefType" minOccurs="0" maxOccurs="unbounded"/>`

3.2.14 element SignerInfoType/SigningComponent

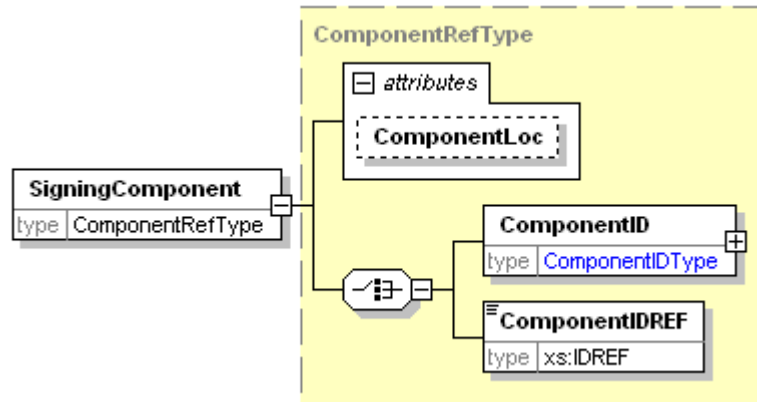
3.2.14.1 Description

The SigningComponent element identifies the tool that was used to generate the signed manifest. The signature over the manifest should include the SigningComponent element. Signing component is a reference to a document that may exist external to *this* document. The integrity values for signing component are not contained in the SigningComponent element.

If specified, this value must be included in the signature computation.

3.2.14.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type [ComponentRefType](#)

properties isRef 0
content complex

children [ComponentID](#) [ComponentIDREF](#)

attributes	Name	Type	Use	Default	Fixed
	ComponentLoc	xs:anyURI	optional		

3.2.14.3 XML

source `<xs:element name="SigningComponent" type="ComponentIDType" minOccurs="0"/>`

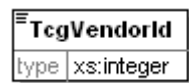
3.2.15 element VendorIdType/TcgVendorId

3.2.15.1 Description

The vendor Id issued by the TCG according to constraints defined by the TCG. It is used to uniquely identify the party responsible for applying change management to the component. Typically this is the component manufacturer or IT.

3.2.15.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type **xs:integer**

properties isRef 0
content simple

3.2.15.3 XML

source `<xs:element name="TcgVendorId" type="xs:integer" minOccurs="0"/>`

3.2.16 element VendorIdType/SmiVendorId

3.2.16.1 Description

This is a vendor Id corresponding to an SMI Network Management Private Enterprise Code issued by the Internet Assigned Number Authority (IANA). It is used to uniquely identify the party responsible for applying change management to the component. Typically this is the component manufacturer or IT.

3.2.16.2 Diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type **xs:integer**

properties isRef 0
 content simple

3.2.16.3 XML

source `<xs:element name="SmiVendorId" type="xs:integer" minOccurs="0"/>`

3.2.17 element VendorIdType/VendorGUID

3.2.17.1 Description

VendorGUID is used to uniquely identify the party responsible for applying change management to the component. Typically this is the component manufacturer or IT.

3.2.17.2 Diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/Core_Integrity_v1_0_1#

type **xs:NMTOKEN**

properties isRef 0
 content simple

3.2.17.3 XML

source `<xs:element name="VendorGUID" type="xs:NMTOKEN" minOccurs="0"/>`

4 References

- [1] TCG Integrity Management Model Architecture Part II
- [2] TCG Reference Integrity Measurement Manifest Schema Specification v1.0
- [3] TCG Integrity Report Schema Specification v1.0
- [4] TCG Simple Object Schema Specification v1.0
- [5] TCG Security Qualities Schema Specification v1.0

5 Appendix A: XML Signature Schema

This section contains a copy of the XML-Signature schema for reader convenience only. This section is non-normative. The reader must refer to the schema location defined in section 2 for normative reference to XML-Signature schema.

schema location: <http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd>
 attribute form default:
 element form default: **qualified**
 targetNamespace: <http://www.w3.org/2000/09/xmldsig#>

Elements

[ds:CanonicalizationMethod](#)
[ds:DigestMethod](#)
[ds:DigestValue](#)
[ds:DSAKeyValue](#)
[ds:KeyInfo](#)
[ds:KeyName](#)
[ds:KeyValue](#)
[ds:Manifest](#)
[ds:MgmtData](#)
[ds:Object](#)
[ds:PGPData](#)
[ds:Reference](#)
[ds:RetrievalMethod](#)
[ds:RSAKeyValue](#)
[ds:Signature](#)
[ds:SignatureMethod](#)
[ds:SignatureProperties](#)
[ds:SignatureProperty](#)
[ds:SignatureValue](#)
[ds:SignedInfo](#)
[ds:SPKIData](#)
[ds:Transform](#)
[ds:Transforms](#)
[ds:X509Data](#)

Complex types

[ds:CanonicalizationMethodType](#)
[ds:DigestMethodType](#)
[ds:DSAKeyValueType](#)
[ds:KeyInfoType](#)
[ds:KeyValueTypes](#)
[ds:ManifestType](#)
[ds:ObjectType](#)
[ds:PGPDataType](#)
[ds:ReferenceType](#)
[ds:RetrievalMethodType](#)
[ds:RSAKeyValueTypes](#)
[ds:SignatureMethodType](#)
[ds:SignaturePropertiesType](#)
[ds:SignaturePropertyType](#)
[ds:SignatureType](#)
[ds:SignatureValueType](#)
[ds:SignedInfoType](#)
[ds:SPKIDataType](#)
[ds:TransformsType](#)
[ds:TransformType](#)
[ds:X509DataType](#)
[ds:X509IssuerSerialType](#)

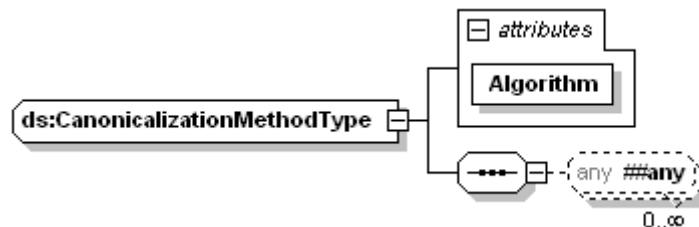
Simple types

[ds:CryptoBinary](#)
[ds:DigestValueType](#)
[ds:HMACOutputLengthType](#)

5.1 Complex Types

5.1.1 complexType ds:CanonicalizationMethodType

diagram

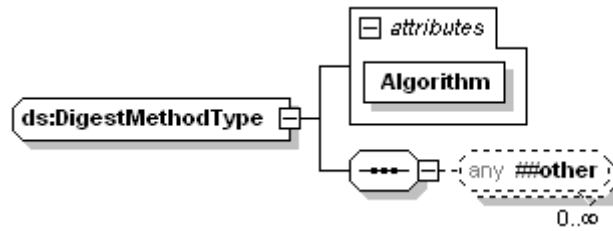


namespace <http://www.w3.org/2000/09/xmldsig#>
 properties mixed true
 used by element [ds:CanonicalizationMethod](#)

attributes	Name	Type	Use	Default	Fixed
	Algorithm	xs:anyURI	required		
source	<pre><xs:complexType name="CanonicalizationMethodType" mixed="true"> <xs:sequence> <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"/> <!-- (0,unbounded) elements from (1,1) namespace --> </xs:sequence> <xs:attribute name="Algorithm" type="anyURI" use="required"/> </xs:complexType></pre>				

5.1.2 complexType ds:DigestMethodType

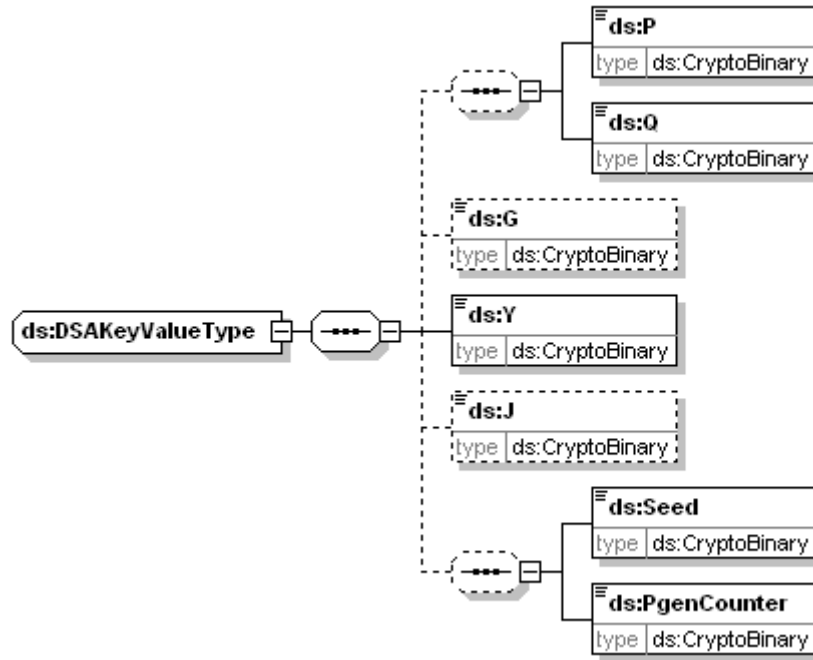
diagram



namespace	http://www.w3.org/2000/09/xmlsig#				
properties	mixed	true			
used by	element	ds:DigestMethod			
	complexType	DigestMethodType			
attributes	Name	Type	Use	Default	Fixed
	Algorithm	xs:anyURI	required		
source	<pre><xs:complexType name="DigestMethodType" mixed="true"> <xs:sequence> <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> <xs:attribute name="Algorithm" type="anyURI" use="required"/> </xs:complexType></pre>				

5.1.3 complexType ds:DSAKeyValue

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

children [ds:P](#) [ds:Q](#) [ds:G](#) [ds:Y](#) [ds:J](#) [ds:Seed](#) [ds:PgenCounter](#)

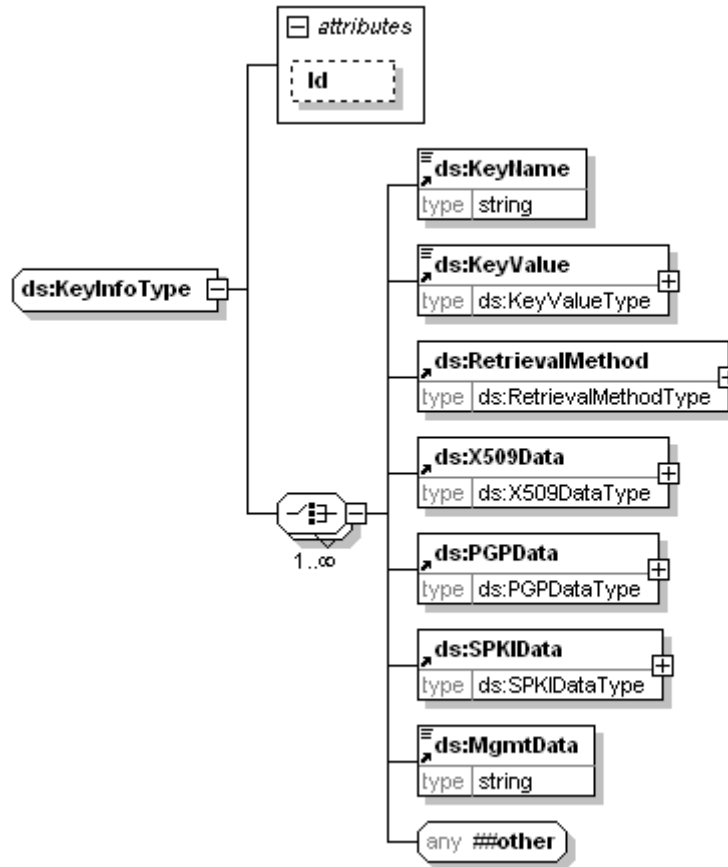
used by element [ds:DSAKeyValue](#)

```

source
<xs:complexType name="DSAKeyValue">
  <xs:sequence>
    <xs:sequence minOccurs="0">
      <xs:element name="P" type="ds:CryptBinary"/>
      <xs:element name="Q" type="ds:CryptBinary"/>
    </xs:sequence>
    <xs:element name="G" type="ds:CryptBinary" minOccurs="0"/>
    <xs:element name="Y" type="ds:CryptBinary"/>
    <xs:element name="J" type="ds:CryptBinary" minOccurs="0"/>
    <xs:sequence minOccurs="0">
      <xs:element name="Seed" type="ds:CryptBinary"/>
      <xs:element name="PgenCounter" type="ds:CryptBinary"/>
    </xs:sequence>
  </xs:sequence>
</xs:complexType>
  
```

5.1.4 complexType ds:KeyInfoType

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

properties mixed true

children [ds:KeyName](#) [ds:KeyValue](#) [ds:RetrievalMethod](#) [ds:X509Data](#) [ds:PGPData](#) [ds:SPKIData](#) [ds:MgmtData](#)

used by element [ds:KeyInfo](#)

attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	optional		

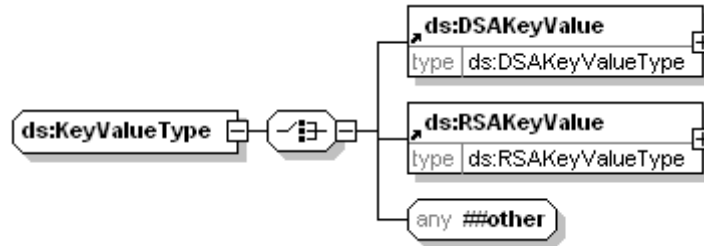
```

source <xs:complexType name="KeyInfoType" mixed="true">
  <xs:choice maxOccurs="unbounded">
    <xs:element ref="ds:KeyName"/>
    <xs:element ref="ds:KeyValue"/>
    <xs:element ref="ds:RetrievalMethod"/>
    <xs:element ref="ds:X509Data"/>
    <xs:element ref="ds:PGPData"/>
    <xs:element ref="ds:SPKIData"/>
    <xs:element ref="ds:MgmtData"/>
    <xs:any namespace="##other" processContents="lax"/>
  <!-- (1,1) elements from (0,unbounded) namespaces -->
</xs:choice>
  <xs:attribute name="Id" type="xs:ID" use="optional"/>
</xs:complexType>

```

5.1.5 complexType ds:KeyValue

diagram



namespace <http://www.w3.org/2000/09/xmlsig#>

properties mixed true

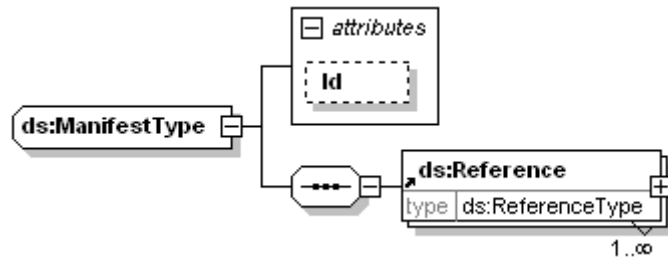
children [ds:DSAKeyValue](#) [ds:RSAKeyValue](#)

used by element [ds:KeyValue](#)

```
<xs:complexType name="KeyValue" mixed="true">
  <xs:choice>
    <xs:element ref="ds:DSAKeyValue"/>
    <xs:element ref="ds:RSAKeyValue"/>
    <xs:any namespace="##other" processContents="lax"/>
  </xs:choice>
</xs:complexType>
```

5.1.6 complexType ds:Manifest

diagram



namespace <http://www.w3.org/2000/09/xmlsig#>

children [ds:Reference](#)

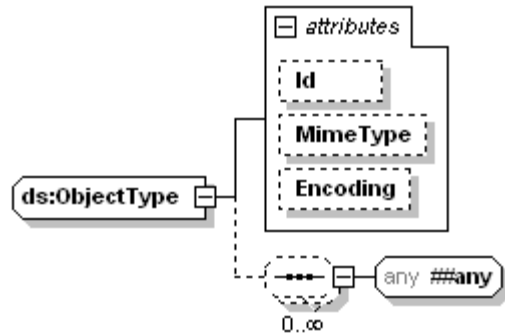
used by element [ds:Manifest](#)

attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	optional		

```
<xs:complexType name="Manifest">
  <xs:sequence>
    <xs:element ref="ds:Reference" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="Id" type="xs:ID" use="optional"/>
</xs:complexType>
```

5.1.7 complexType ds:ObjectType

diagram



namespace <http://www.w3.org/2000/09/xmlsig#>

properties mixed true

used by element [ds:Object](#)

attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	optional		
	MimeType	xs:string	optional		
	Encoding	xs:anyURI	optional		

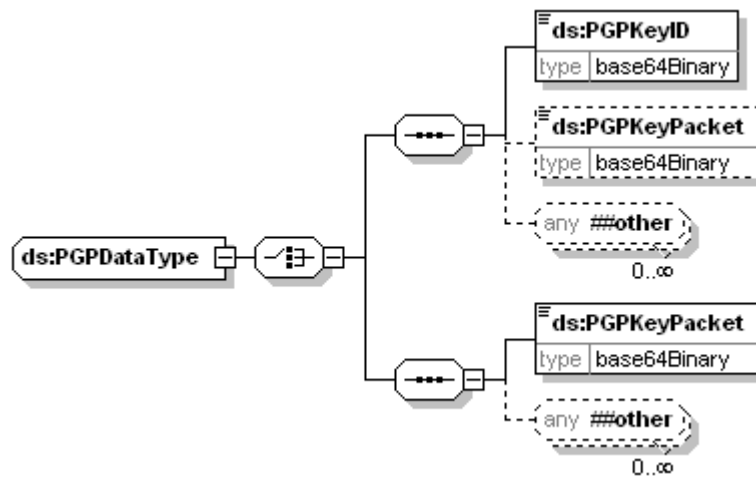
```

source <xs:complexType name="ObjectType" mixed="true">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:any namespace="##any" processContents="lax"/>
  </xs:sequence>
  <xs:attribute name="Id" type="xs:ID" use="optional"/>
  <xs:attribute name="MimeType" type="string" use="optional"/>
  <xs:attribute name="Encoding" type="anyURI" use="optional"/>
  <!-- add a grep facet -->
</xs:complexType>

```

5.1.8 complexType ds:PGPDataType

diagram



namespace <http://www.w3.org/2000/09/xmlsig#>

children [ds:PGPKeyID](#) [ds:PGPKeyPacket](#) [ds:PGPKeyPacket](#)

used by element [ds:PGPData](#)

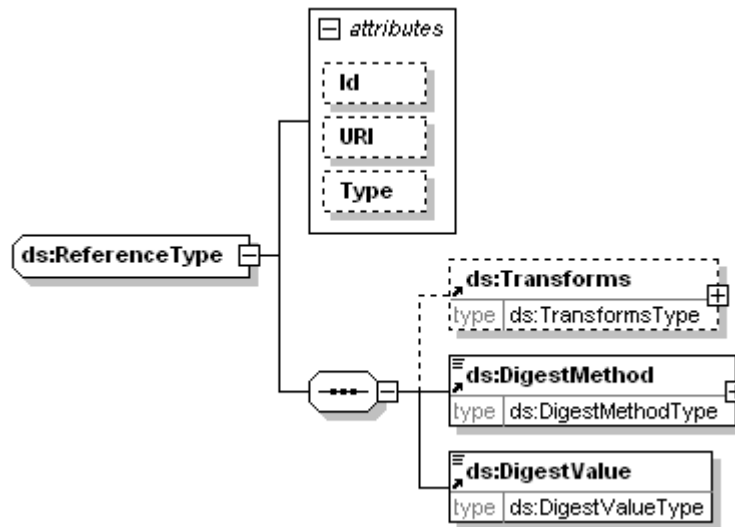
```

source <xs:complexType name="PGPDataType">
  <xs:choice>
    <xs:sequence>
      <xs:element name="PGPKeyID" type="base64Binary"/>
      <xs:element name="PGPKeyPacket" type="base64Binary" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:sequence>
      <xs:element name="PGPKeyPacket" type="base64Binary"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:choice>
</xs:complexType>

```

5.1.9 complexType ds:ReferenceType

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

children [ds:Transforms](#) [ds:DigestMethod](#) [ds:DigestValue](#)

used by element [ds:Reference](#)

attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	optional		
	URI	xs:anyURI	optional		
	Type	xs:anyURI	optional		

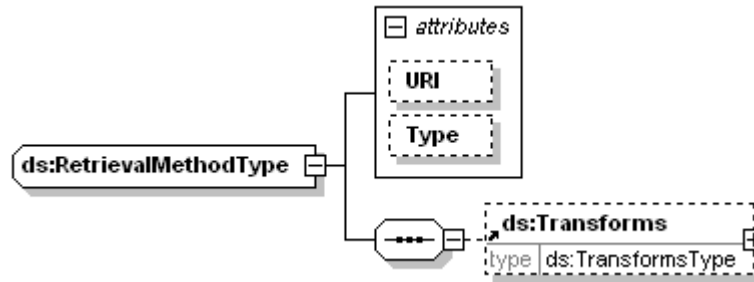
```

source <xs:complexType name="ReferenceType">
  <xs:sequence>
    <xs:element ref="ds:Transforms" minOccurs="0"/>
    <xs:element ref="ds:DigestMethod"/>
    <xs:element ref="ds:DigestValue"/>
  </xs:sequence>
  <xs:attribute name="Id" type="ID" use="optional"/>
  <xs:attribute name="URI" type="anyURI" use="optional"/>
  <xs:attribute name="Type" type="anyURI" use="optional"/>
</xs:complexType>

```

5.1.10 complexType ds:RetrievalMethodType

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

children [ds:Transforms](#)

used by element [ds:RetrievalMethod](#)

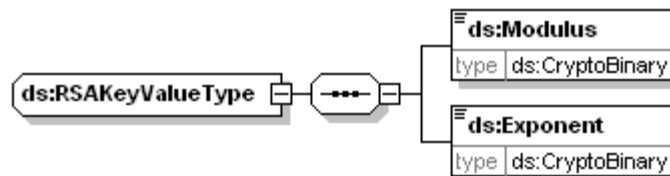
attributes	Name	Type	Use	Default	Fixed
	URI	xs:anyURI			
	Type	xs:anyURI	optional		

source

```
<xs:complexType name="RetrievalMethodType">
  <xs:sequence>
    <xs:element ref="ds:Transforms" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="URI" type="anyURI"/>
  <xs:attribute name="Type" type="anyURI" use="optional"/>
</xs:complexType>
```

5.1.11 complexType ds:RSAKeyValueType

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

children [ds:Modulus](#) [ds:Exponent](#)

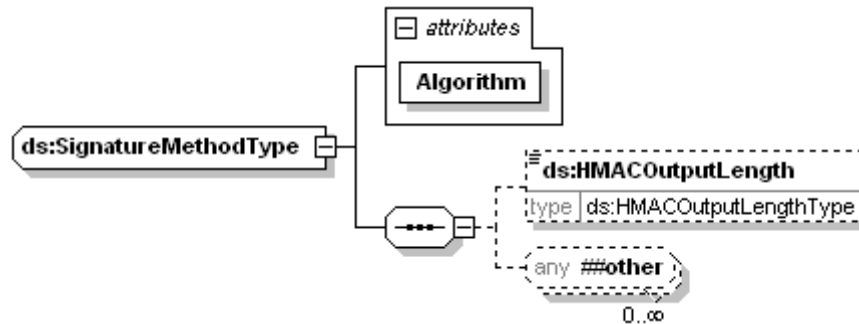
used by element [ds:RSAKeyValue](#)

source

```
<xs:complexType name="RSAKeyValueType">
  <xs:sequence>
    <xs:element name="Modulus" type="ds:CryptoBinary"/>
    <xs:element name="Exponent" type="ds:CryptoBinary"/>
  </xs:sequence>
</xs:complexType>
```

5.1.12 complexType ds:SignatureMethodType

diagram



namespace <http://www.w3.org/2000/09/xmlsig#>

properties mixed true

children [ds:HMACOutputLength](#)

used by element [ds:SignatureMethod](#)

attributes	Name	Type	Use	Default	Fixed
	Algorithm	xs:anyURI	required		

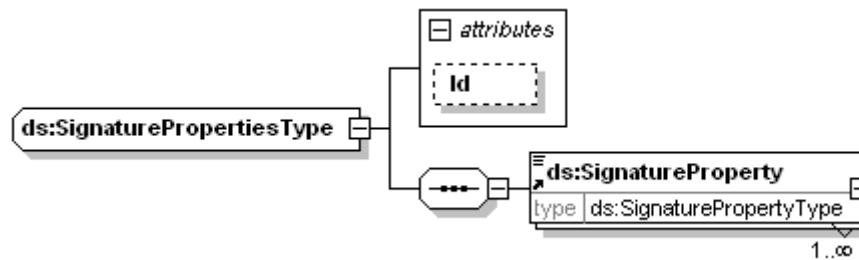
```

source <xs:complexType name="SignatureMethodType" mixed="true">
  <xs:sequence>
    <xs:element name="HMACOutputLength" type="ds:HMACOutputLengthType" minOccurs="0"/>
    <xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
    <!-- (0,unbounded) elements from (1,1) external namespace -->
  </xs:sequence>
  <xs:attribute name="Algorithm" type="anyURI" use="required"/>
</xs:complexType>

```

5.1.13 complexType ds:SignaturePropertiesType

diagram



namespace <http://www.w3.org/2000/09/xmlsig#>

children [ds:SignatureProperty](#)

used by element [ds:SignatureProperties](#)

attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	optional		

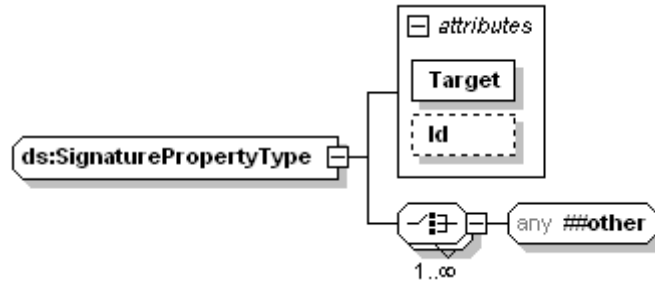
```

source <xs:complexType name="SignaturePropertiesType">
  <xs:sequence>
    <xs:element ref="ds:SignatureProperty" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="Id" type="ID" use="optional"/>
</xs:complexType>

```

5.1.14 complexType ds:SignaturePropertyType

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

properties mixed true

used by element [ds:SignatureProperty](#)

attributes	Name	Type	Use	Default	Fixed
	Target	xs:anyURI	required		
	Id	xs:ID	optional		

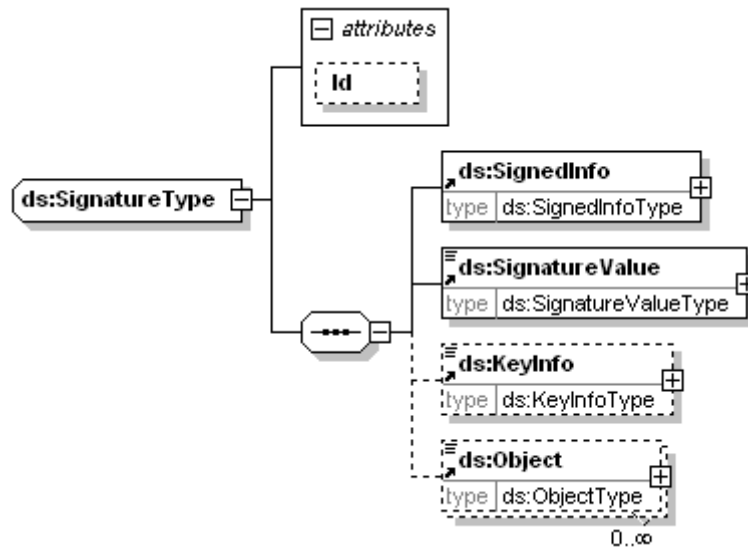
```

source <xs:complexType name="SignaturePropertyType" mixed="true">
  <xs:choice maxOccurs="unbounded">
    <xs:any namespace="##other" processContents="lax"/>
    <!-- (1,1) elements from (1,unbounded) namespaces -->
  </xs:choice>
  <xs:attribute name="Target" type="anyURI" use="required"/>
  <xs:attribute name="Id" type="ID" use="optional"/>
</xs:complexType>

```

5.1.15 complexType ds:SignatureType

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

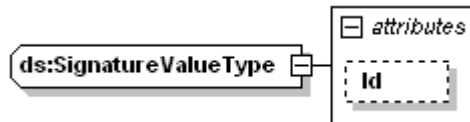
children [ds:SignedInfo](#) [ds:SignatureValue](#) [ds:KeyInfo](#) [ds:Object](#)

used by element [ds:Signature](#)

attributes	Name Id	Type xs:ID	Use optional	Default	Fixed
source	<pre><xs:complexType name="SignatureType"> <xs:sequence> <xs:element ref="ds:SignedInfo"/> <xs:element ref="ds:SignatureValue"/> <xs:element ref="ds:KeyInfo" minOccurs="0"/> <xs:element ref="ds:Object" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> <xs:attribute name="Id" type="ID" use="optional"/> </xs:complexType></pre>				

5.1.16 complexType ds:SignatureValueType

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

type extension of **xs:base64Binary**

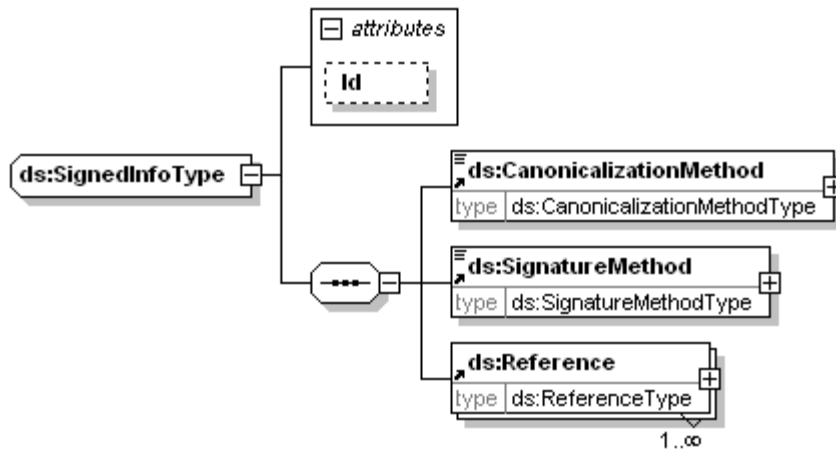
properties base base64Binary

used by element [ds:SignatureValue](#)

attributes	Name Id	Type xs:ID	Use optional	Default	Fixed
source	<pre><xs:complexType name="SignatureValueType"> <xs:simpleContent> <xs:extension base="base64Binary"> <xs:attribute name="Id" type="ID" use="optional"/> </xs:extension> </xs:simpleContent> </xs:complexType></pre>				

5.1.17 complexType ds:SignedInfoType

diagram



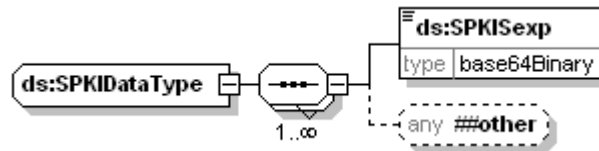
namespace <http://www.w3.org/2000/09/xmldsig#>

children [ds:CanonicalizationMethod](#) [ds:SignatureMethod](#) [ds:Reference](#)

used by	element	ds:SignedInfo			
attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	optional		
source	<pre><xs:complexType name="SignedInfoType"> <xs:sequence> <xs:element ref="ds:CanonicalizationMethod"/> <xs:element ref="ds:SignatureMethod"/> <xs:element ref="ds:Reference" maxOccurs="unbounded"/> </xs:sequence> <xs:attribute name="Id" type="ID" use="optional"/> </xs:complexType></pre>				

5.1.18 complexType ds:SPKIDataType

diagram



namespace <http://www.w3.org/2000/09/xmlsig#>

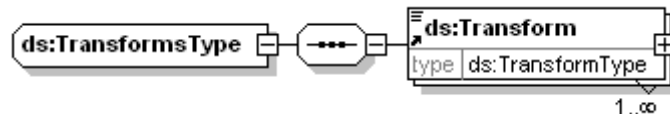
children [ds:SPKISexp](#)

used by element [ds:SPKIData](#)

```
<xs:complexType name="SPKIDataType">
  <xs:sequence maxOccurs="unbounded">
    <xs:element name="SPKISexp" type="base64Binary"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

5.1.19 complexType ds:TransformsType

diagram



namespace <http://www.w3.org/2000/09/xmlsig#>

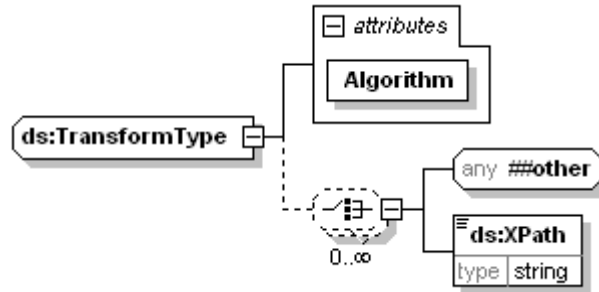
children [ds:Transform](#)

used by element [ds:Transforms](#)

```
<xs:complexType name="TransformsType">
  <xs:sequence>
    <xs:element ref="ds:Transform" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

5.1.20 complexType ds:TransformType

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

properties mixed true

children [ds:XPath](#)

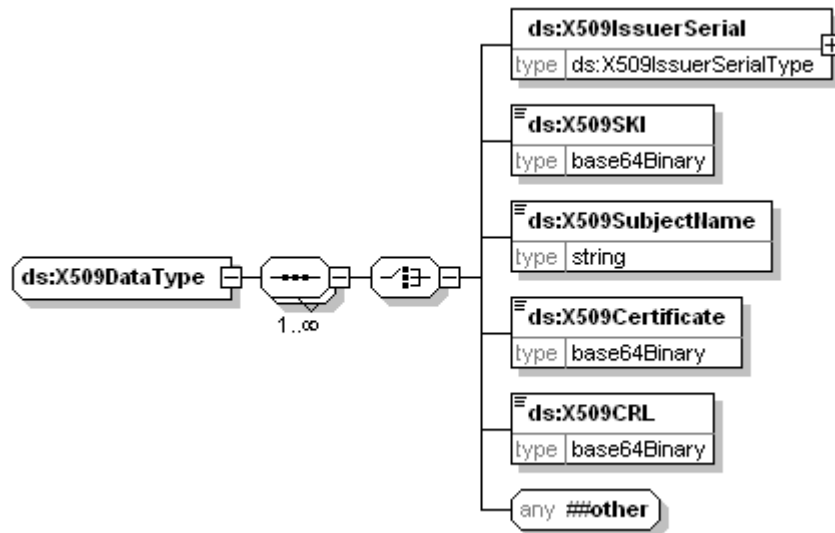
used by element [ds:Transform](#)
complexType [TransformMethodType](#)

attributes	Name	Type	Use	Default	Fixed
	Algorithm	xs:anyURI	required		

```
<xs:complexType name="TransformType" mixed="true">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:any namespace="##other" processContents="lax"/>
    <!-- (1,1) elements from (0,unbounded) namespaces -->
    <xs:element name="XPath" type="string"/>
  </xs:choice>
  <xs:attribute name="Algorithm" type="anyURI" use="required"/>
</xs:complexType>
```

5.1.21 complexType ds:X509DataType

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

children [ds:X509IssuerSerial](#) [ds:X509SKI](#) [ds:X509SubjectName](#) [ds:X509Certificate](#) [ds:X509CRL](#)

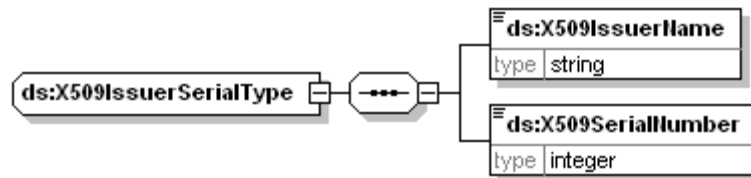
used by element [ds:X509Data](#)

source

```
<xs:complexType name="X509DataType">
  <xs:sequence maxOccurs="unbounded">
    <xs:choice>
      <xs:element name="X509IssuerSerial" type="ds:X509IssuerSerialType"/>
      <xs:element name="X509SKI" type="base64Binary"/>
      <xs:element name="X509SubjectName" type="string"/>
      <xs:element name="X509Certificate" type="base64Binary"/>
      <xs:element name="X509CRL" type="base64Binary"/>
      <xs:any namespace="##other" processContents="lax"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

5.1.22 complexType ds:X509IssuerSerialType

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

children [ds:X509IssuerName](#) [ds:X509SerialNumber](#)

used by element [ds:X509DataType/X509IssuerSerial](#)

source

```
<xs:complexType name="X509IssuerSerialType">
  <xs:sequence>
    <xs:element name="X509IssuerName" type="string"/>
    <xs:element name="X509SerialNumber" type="integer"/>
  </xs:sequence>
</xs:complexType>
```

5.2 Simple Types

5.2.1 simpleType ds:CryptoBinary

namespace <http://www.w3.org/2000/09/xmldsig#>

type **xs:base64Binary**

used by elements [ds:RSAKeyValue/Exponent](#) [ds:DSAKeyValue/G](#) [ds:DSAKeyValue/J](#)
[ds:RSAKeyValue/Modulus](#) [ds:DSAKeyValue/P](#) [ds:DSAKeyValue/PgenCounter](#)
[ds:DSAKeyValue/Q](#) [ds:DSAKeyValue/Seed](#) [ds:DSAKeyValue/Y](#)

source

```
<xs:simpleType name="CryptoBinary">
  <xs:restriction base="base64Binary"/>
</xs:simpleType>
```

5.2.2 simpleType ds:DigestValueType

namespace <http://www.w3.org/2000/09/xmldsig#>

type **xs:base64Binary**

used by element [ds:DigestValue](#)
 complexType [DigestValueType](#)
 attributes [DigestType/@Hash](#) [DigestType/@StartHash](#)

source `<xs:simpleType name="DigestValueType">
 <xs:restriction base="base64Binary"/>
 </xs:simpleType>`

5.2.3 simpleType ds:HMACOutputLengthType

namespace <http://www.w3.org/2000/09/xmlsig#>

type **xs:integer**

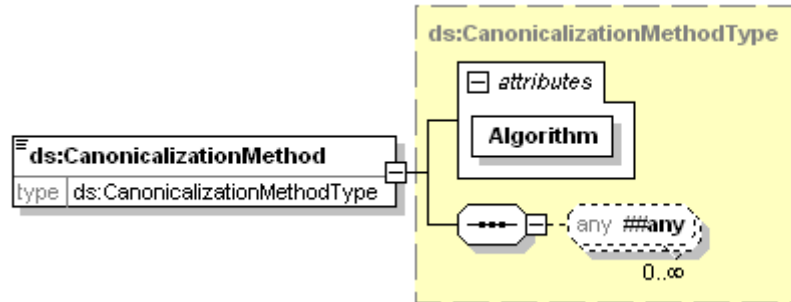
used by element [ds:SignatureMethodType/HMACOutputLength](#)

source `<xs:simpleType name="HMACOutputLengthType">
 <xs:restriction base="integer"/>
 </xs:simpleType>`

5.3 Elements

5.3.1 element ds:CanonicalizationMethod

diagram



namespace <http://www.w3.org/2000/09/xmlsig#>

type [ds:CanonicalizationMethodType](#)

properties content complex
 mixed true

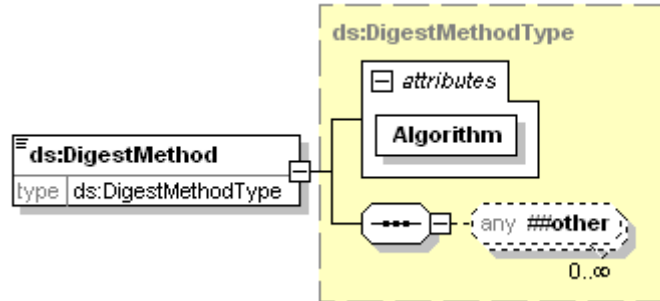
used by complexType [ds:SignedInfoType](#)

attributes	Name	Type	Use	Default	Fixed
	Algorithm	xs:anyURI	required		

source `<xs:element name="CanonicalizationMethod" type="ds:CanonicalizationMethodType"/>`

5.3.2 element ds:DigestMethod

diagram



namespace <http://www.w3.org/2000/09/xmlsig#>

type [ds:DigestMethodType](#)

properties content complex
mixed true
used by complexType [ds:ReferenceType](#)

attributes	Name	Type	Use	Default	Fixed
	Algorithm	xs:anyURI	required		
source	<code><xs:element name="DigestMethod" type="ds:DigestMethodType"/></code>				

5.3.3 element ds:DigestValue

diagram



namespace <http://www.w3.org/2000/09/xmlsig#>

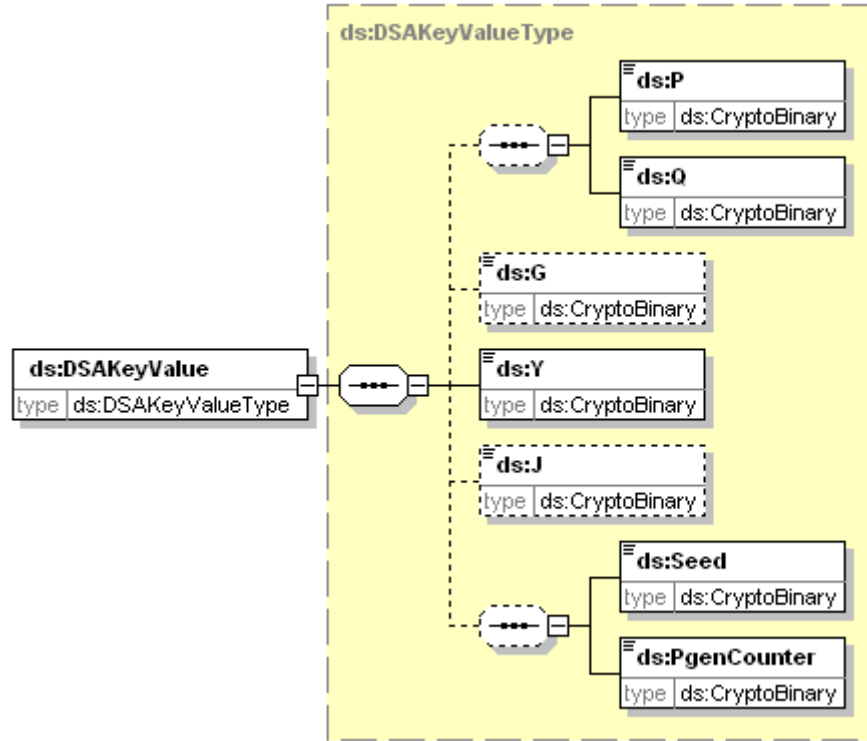
type [ds:DigestValueType](#)

properties content simple
used by complexType [ds:ReferenceType](#)

source `<xs:element name="DigestValue" type="ds:DigestValueType"/>`

5.3.4 element ds:DSAKeyValue

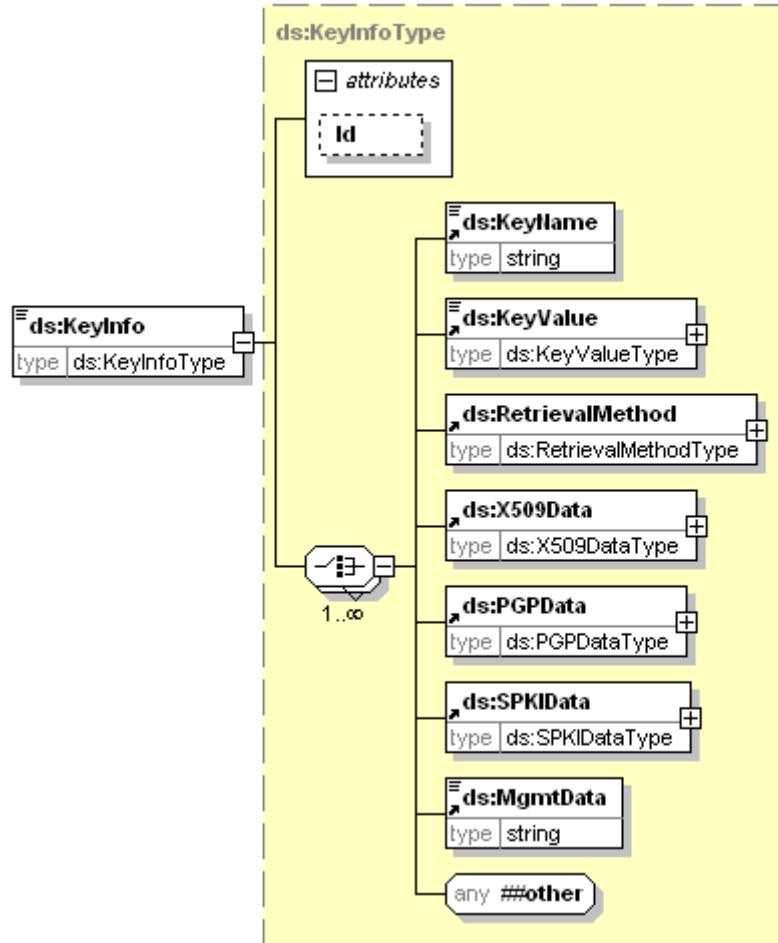
diagram



namespace <http://www.w3.org/2000/09/xmldsig#>
 type [ds:DSAKeyValue Type](#)
 properties content complex
 children [ds:P](#) [ds:Q](#) [ds:G](#) [ds:Y](#) [ds:J](#) [ds:Seed](#) [ds:PgenCounter](#)
 used by complexType [ds:KeyValue Type](#)
 source `<xs:element name="DSAKeyValue" type="ds:DSAKeyValue Type"/>`

5.3.5 element ds:KeyInfo

diagram



namespace <http://www.w3.org/2000/09/xmlsig#>

type [ds:KeyInfoType](#)

properties content complex
mixed true

children [ds:KeyName](#) [ds:KeyValue](#) [ds:RetrievalMethod](#) [ds:X509Data](#) [ds:PGPData](#) [ds:SPKIDData](#) [ds:MgmtData](#)

used by complexType [ds:SignatureType](#)

attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	optional		

source `<xs:element name="KeyInfo" type="ds:KeyInfoType"/>`

5.3.6 element ds:KeyName

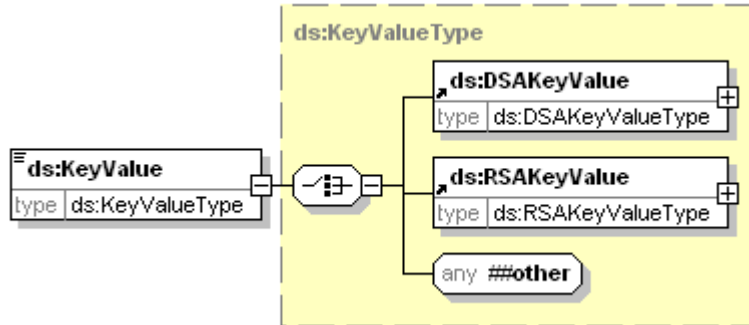
diagram



namespace <http://www.w3.org/2000/09/xmldsig#>
 type **xs:string**
 properties content simple
 used by complexType [ds:KeyInfoType](#)
 source `<xs:element name="KeyName" type="string"/>`

5.3.7 element ds:KeyValue

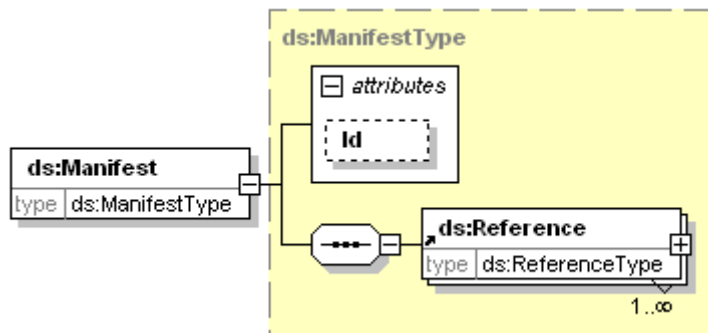
diagram



namespace <http://www.w3.org/2000/09/xmldsig#>
 type [ds:KeyValue](#)
 properties content complex
 mixed true
 children [ds:DSAKeyValue](#) [ds:RSAKeyValue](#)
 used by complexType [ds:KeyInfoType](#)
 source `<xs:element name="KeyValue" type="ds:KeyValue"/>`

5.3.8 element ds:Manifest

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>
 type [ds:Manifest](#)
 properties content complex
 children [ds:Reference](#)
 attributes

Name	Type	Use	Default	Fixed
Id	xs:ID	optional		

source `<xs:element name="Manifest" type="ds:ManifestType"/>`

5.3.9 element ds:MgmtData

diagram



namespace `http://www.w3.org/2000/09/xmldsig#`

type **xs:string**

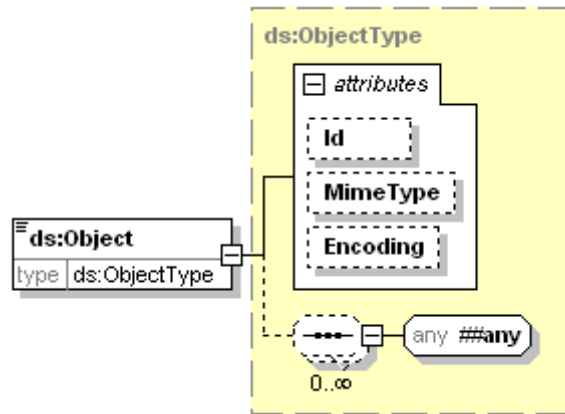
properties content simple

used by complexType [ds:KeyInfoType](#)

source `<xs:element name="MgmtData" type="string"/>`

5.3.10 element ds:Object

diagram



namespace `http://www.w3.org/2000/09/xmldsig#`

type [ds:ObjectType](#)

properties content complex
mixed true

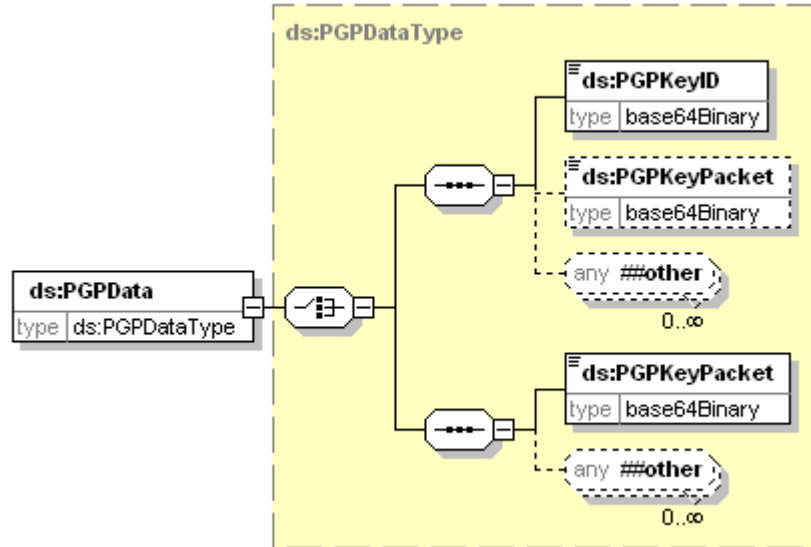
used by complexType [ds:SignatureType](#)

attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	optional		
	MimeType	xs:string	optional		
	Encoding	xs:anyURI	optional		

source `<xs:element name="Object" type="ds:ObjectType"/>`

5.3.11 element ds:PGPData

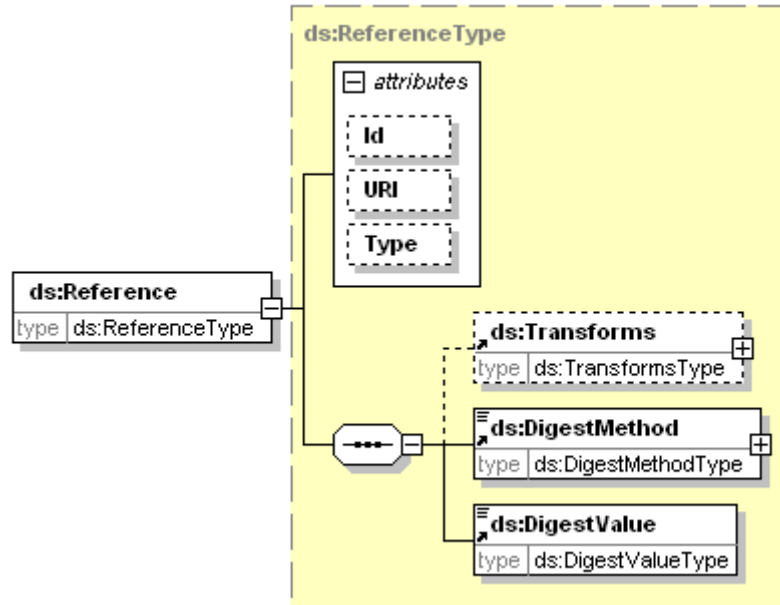
diagram



namespace <http://www.w3.org/2000/09/xmldsig#>
 type [ds:PGPDataType](#)
 properties content complex
 children [ds:PGPKeyID](#) [ds:PGPKeyPacket](#) [ds:PGPKeyPacket](#)
 used by complexType [ds:KeyInfoType](#)
 source `<xs:element name="PGPData" type="ds:PGPDataType"/>`

5.3.12 element ds:Reference

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

type [ds:ReferenceType](#)

properties content complex

children [ds:Transforms](#) [ds:DigestMethod](#) [ds:DigestValue](#)

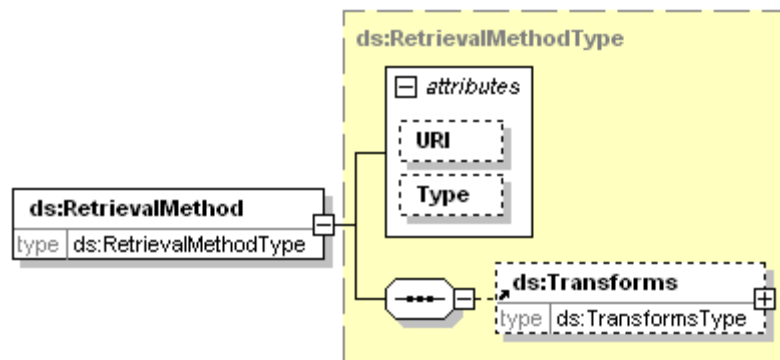
used by complexTypes [ds:ManifestType](#) [ds:SignedInfoType](#)

attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	optional		
	URI	xs:anyURI	optional		
	Type	xs:anyURI	optional		

source `<xs:element name="Reference" type="ds:ReferenceType"/>`

5.3.13 element ds:RetrievalMethod

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

type [ds:RetrievalMethodType](#)

properties content complex

children [ds:Transforms](#)

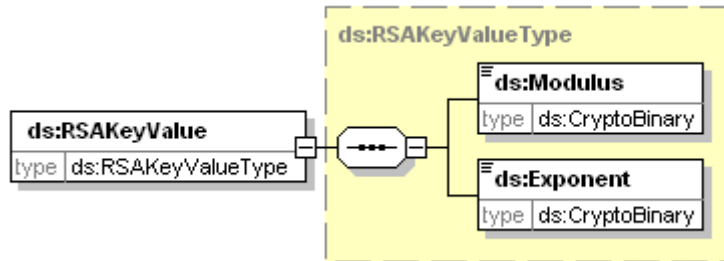
used by complexType [ds:KeyInfoType](#)

attributes	Name	Type	Use	Default	Fixed
	URI	xs:anyURI			
	Type	xs:anyURI	optional		

source `<xs:element name="RetrievalMethod" type="ds:RetrievalMethodType"/>`

5.3.14 element ds:RSAKeyValue

diagram



namespace `http://www.w3.org/2000/09/xmldsig#`

type [ds:RSAKeyValue](#)

properties content complex

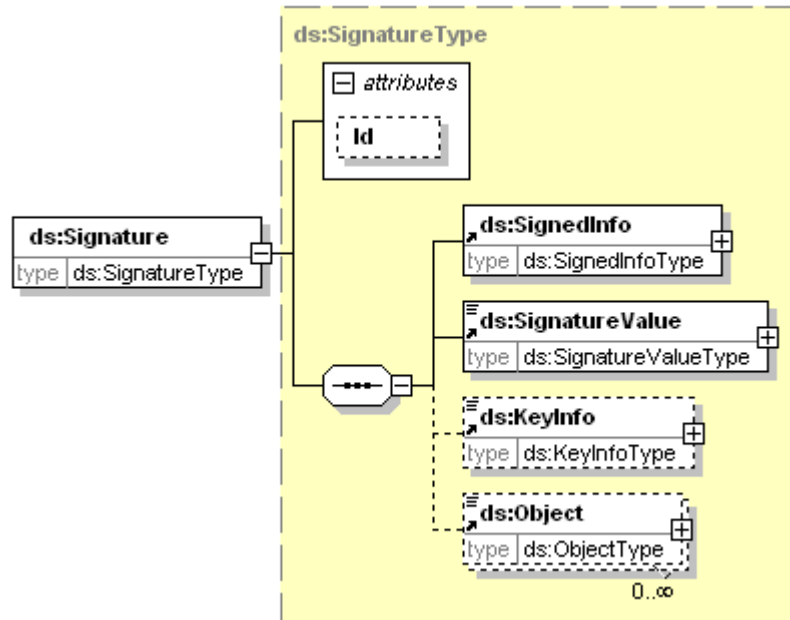
children [ds:Modulus](#) [ds:Exponent](#)

used by complexType [ds:KeyValue](#)

source `<xs:element name="RSAKeyValue" type="ds:RSAKeyValue"/>`

5.3.15 element ds:Signature

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

type [ds:SignatureType](#)

properties content complex

children [ds:SignedInfo](#) [ds:SignatureValue](#) [ds:KeyInfo](#) [ds:Object](#)

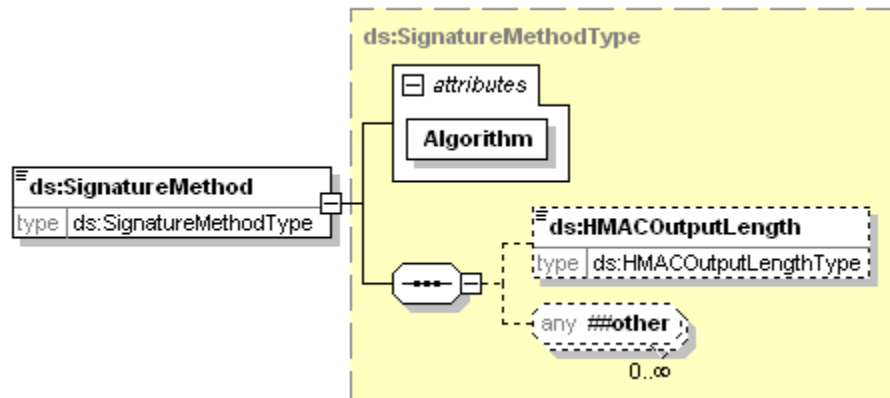
used by complexType [SignerInfoType](#)

attributes	Name	Type	Use	optional	Default	Fixed
	Id	xs:ID				

source `<xs:element name="Signature" type="ds:SignatureType"/>`

5.3.16 element ds:SignatureMethod

diagram

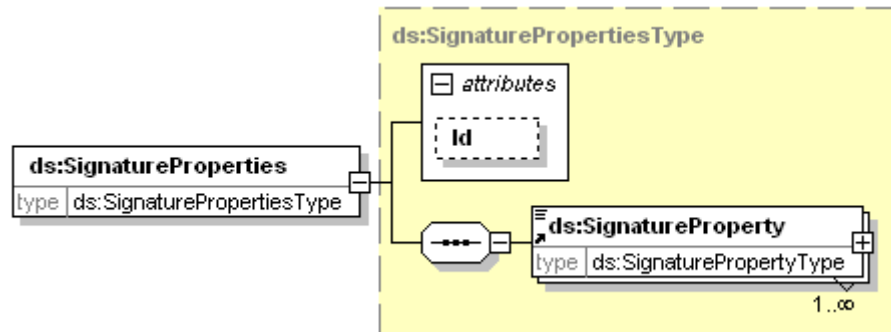


namespace <http://www.w3.org/2000/09/xmldsig#>

type	ds:SignatureMethodType				
properties	content	complex			
	mixed	true			
children	ds:HMACOutputLength				
used by	complexType	ds:SignedInfoType			
attributes	Name	Type	Use	Default	Fixed
	Algorithm	xs:anyURI	required		
source	<code><xs:element name="SignatureMethod" type="ds:SignatureMethodType"/></code>				

5.3.17 element ds:SignatureProperties

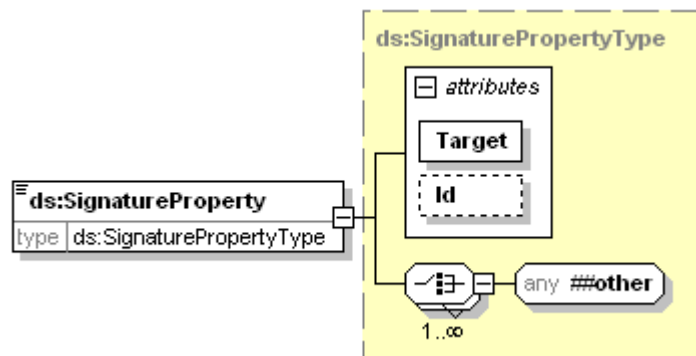
diagram



namespace	http://www.w3.org/2000/09/xmlsig#				
type	ds:SignaturePropertiesType				
properties	content	complex			
children	ds:SignatureProperty				
attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	optional		
source	<code><xs:element name="SignatureProperties" type="ds:SignaturePropertiesType"/></code>				

5.3.18 element ds:SignatureProperty

diagram

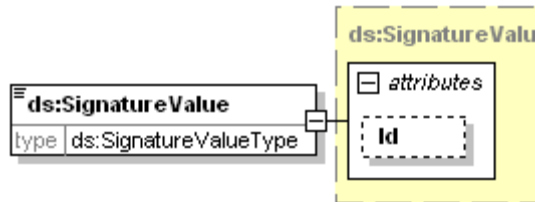


namespace	http://www.w3.org/2000/09/xmlsig#				
type	ds:SignaturePropertyType				
properties	content	complex			
	mixed	true			

used by	complexType	ds:SignaturePropertiesType			
attributes	Name	Type	Use	Default	Fixed
	Target	xs:anyURI	required		
	Id	xs:ID	optional		
source	<code><xs:element name="SignatureProperty" type="ds:SignaturePropertyType"/></code>				

5.3.19 element ds:SignatureValue

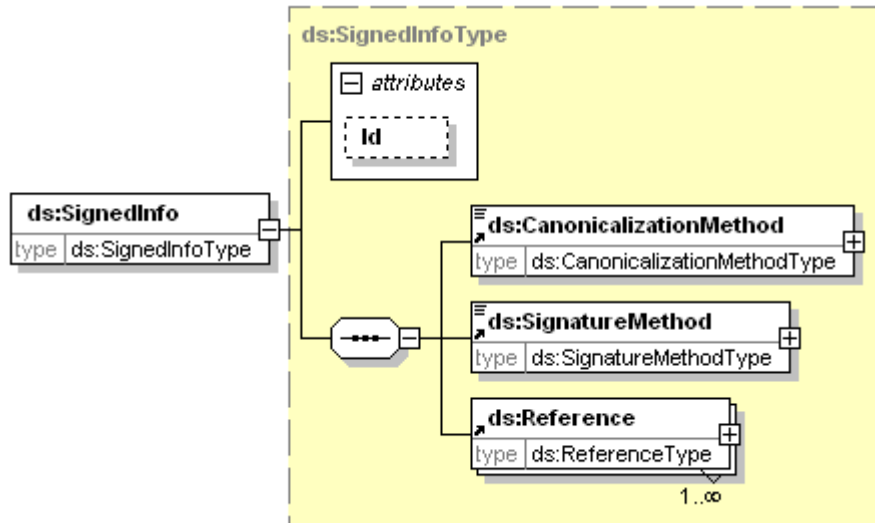
diagram



namespace	http://www.w3.org/2000/09/xmlsig#				
type	ds:SignatureValueType				
properties	content	complex			
used by	complexType	ds:SignatureType			
attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	optional		
source	<code><xs:element name="SignatureValue" type="ds:SignatureValueType"/></code>				

5.3.20 element ds:SignedInfo

diagram

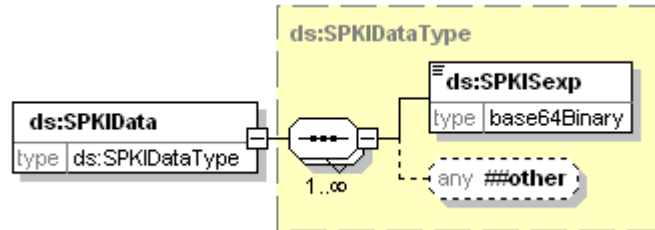


namespace	http://www.w3.org/2000/09/xmlsig#				
type	ds:SignedInfoType				
properties	content	complex			
children	ds:CanonicalizationMethod ds:SignatureMethod ds:Reference				

used by	complexType	ds:SignatureType			
attributes	Name	Type	Use	Default	Fixed
	Id	xs:ID	optional		
source	<code><xs:element name="SignedInfo" type="ds:SignedInfoType"/></code>				

5.3.21 element ds:SPKIData

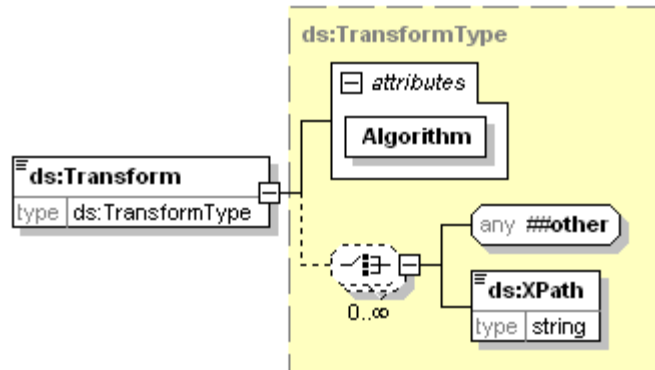
diagram



namespace	http://www.w3.org/2000/09/xmlsig#				
type	ds:SPKIDataType				
properties	content	complex			
children	ds:SPKISexp				
used by	complexType	ds:KeyInfoType			
source	<code><xs:element name="SPKIData" type="ds:SPKIDataType"/></code>				

5.3.22 element ds:Transform

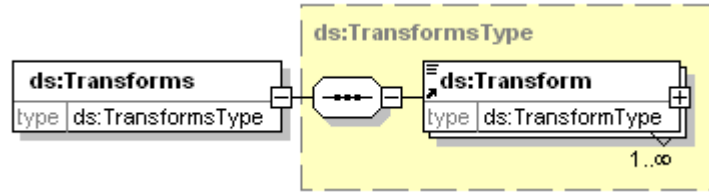
diagram



namespace	http://www.w3.org/2000/09/xmlsig#				
type	ds:TransformType				
properties	content	complex			
	mixed	true			
children	ds:XPath				
used by	complexType	ds:TransformsType			
attributes	Name	Type	Use	Default	Fixed
	Algorithm	xs:anyURI	required		
source	<code><xs:element name="Transform" type="ds:TransformType"/></code>				

5.3.23 element ds:Transforms

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

type [ds:TransformsType](#)

properties content complex

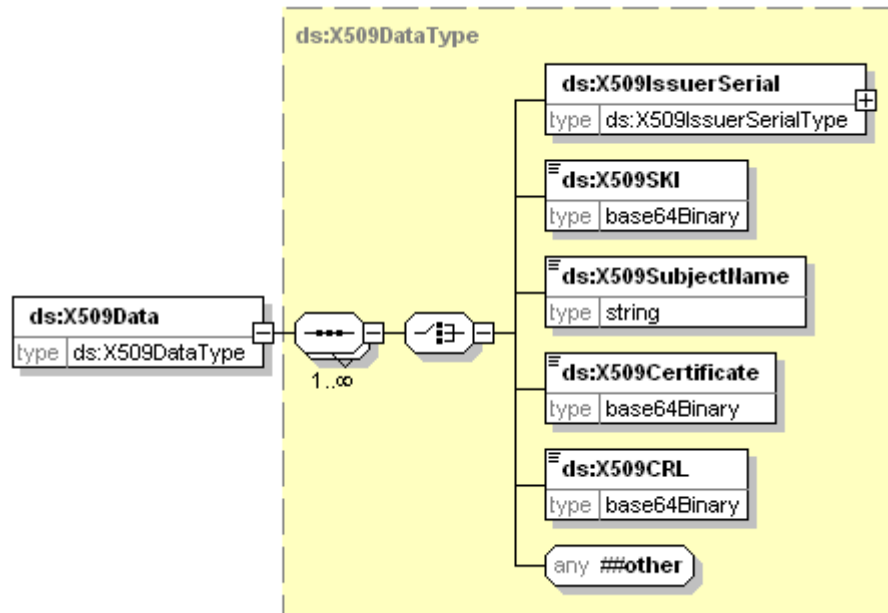
children [ds:Transform](#)

used by complexTypes [ds:ReferenceType](#) [ds:RetrievalMethodType](#)

source `<xs:element name="Transforms" type="ds:TransformsType"/>`

5.3.24 element ds:X509Data

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

type [ds:X509DataType](#)

properties content complex

children [ds:X509IssuerSerial](#) [ds:X509SKI](#) [ds:X509SubjectName](#) [ds:X509Certificate](#) [ds:X509CRL](#)

used by complexType [ds:KeyInfoType](#)

source `<xs:element name="X509Data" type="ds:X509DataType"/>`

5.3.25 element ds:DSAKeyValue/P



namespace <http://www.w3.org/2000/09/xmldsig#>

type [ds:CryptoBinary](#)

properties isRef 0
content simple

source `<xs:element name="P" type="ds:CryptoBinary"/>`

5.3.26 element ds:DSAKeyValue/Q



namespace <http://www.w3.org/2000/09/xmldsig#>

type [ds:CryptoBinary](#)

properties isRef 0
content simple

source `<xs:element name="Q" type="ds:CryptoBinary"/>`

5.3.27 element ds:DSAKeyValue/G



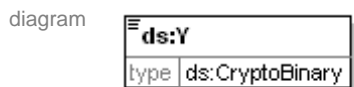
namespace <http://www.w3.org/2000/09/xmldsig#>

type [ds:CryptoBinary](#)

properties isRef 0
content simple

source `<xs:element name="G" type="ds:CryptoBinary" minOccurs="0"/>`

5.3.28 element ds:DSAKeyValue/Y



namespace <http://www.w3.org/2000/09/xmldsig#>

type [ds:CryptoBinary](#)

properties isRef 0
content simple

source `<xs:element name="Y" type="ds:CryptoBinary"/>`

5.3.29 element ds:DSAKeyValue/J



namespace `http://www.w3.org/2000/09/xmldsig#`

type [ds:CryptoBinary](#)

properties
isRef 0
content simple

source `<xs:element name="J" type="ds:CryptoBinary" minOccurs="0"/>`

5.3.30 element ds:DSAKeyValue/Seed



namespace `http://www.w3.org/2000/09/xmldsig#`

type [ds:CryptoBinary](#)

properties
isRef 0
content simple

source `<xs:element name="Seed" type="ds:CryptoBinary"/>`

5.3.31 element ds:DSAKeyValue/PgenCounter



namespace `http://www.w3.org/2000/09/xmldsig#`

type [ds:CryptoBinary](#)

properties
isRef 0
content simple

source `<xs:element name="PgenCounter" type="ds:CryptoBinary"/>`

5.3.32 element ds:PGPDataType/PGPKeyID



namespace `http://www.w3.org/2000/09/xmldsig#`

type `xs:base64Binary`

properties isRef 0
 content simple
source `<xs:element name="PGPKeyID" type="base64Binary"/>`

5.3.33 element ds:PGPDataType/PGPKeyPacket



namespace `http://www.w3.org/2000/09/xmldsig#`
type **xs:base64Binary**
properties isRef 0
 content simple
source `<xs:element name="PGPKeyPacket" type="base64Binary" minOccurs="0"/>`

5.3.34 element ds:PGPDataType/PGPKeyPacket



namespace `http://www.w3.org/2000/09/xmldsig#`
type **xs:base64Binary**
properties isRef 0
 content simple
source `<xs:element name="PGPKeyPacket" type="base64Binary"/>`

5.3.35 element ds:RSAKeyValue/Modulus



namespace `http://www.w3.org/2000/09/xmldsig#`
type [ds:CryptoBinary](#)
properties isRef 0
 content simple
source `<xs:element name="Modulus" type="ds:CryptoBinary"/>`

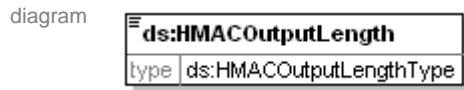
5.3.36 element ds:RSAKeyValue/Exponent



namespace `http://www.w3.org/2000/09/xmldsig#`

type [ds:CryptoBinary](#)
properties isRef 0
content simple
source `<xs:element name="Exponent" type="ds:CryptoBinary"/>`

5.3.37 element ds:SignatureMethodType/HMACOutputLength



namespace `http://www.w3.org/2000/09/xmldsig#`

type [ds:HMACOutputLengthType](#)
properties isRef 0
content simple
source `<xs:element name="HMACOutputLength" type="ds:HMACOutputLengthType" minOccurs="0"/>`

5.3.38 element ds:SPKIDataType/SPKISexp



namespace `http://www.w3.org/2000/09/xmldsig#`

type **xs:base64Binary**
properties isRef 0
content simple
source `<xs:element name="SPKISexp" type="base64Binary"/>`

5.3.39 element ds:TransformType/XPath

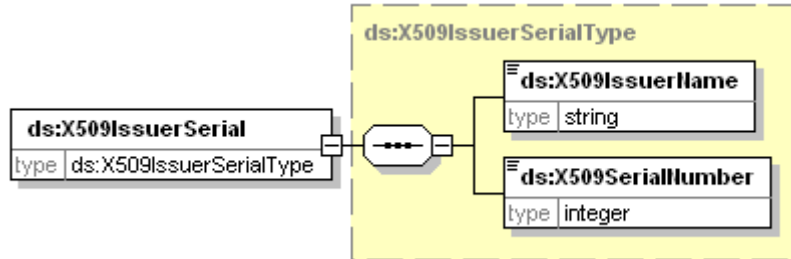


namespace `http://www.w3.org/2000/09/xmldsig#`

type **xs:string**
properties isRef 0
content simple
source `<xs:element name="XPath" type="string"/>`

5.3.40 element ds:X509DataType/X509IssuerSerial

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

type [ds:X509IssuerSerialType](#)

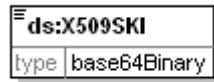
properties isRef 0
content complex

children [ds:X509IssuerName](#) [ds:X509SerialNumber](#)

source `<xs:element name="X509IssuerSerial" type="ds:X509IssuerSerialType"/>`

5.3.41 element ds:X509DataType/X509SKI

diagram



namespace <http://www.w3.org/2000/09/xmldsig#>

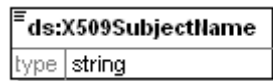
type **xs:base64Binary**

properties isRef 0
content simple

source `<xs:element name="X509SKI" type="base64Binary"/>`

5.3.42 element ds:X509DataType/X509SubjectName

diagram



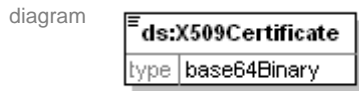
namespace <http://www.w3.org/2000/09/xmldsig#>

type **xs:string**

properties isRef 0
content simple

source `<xs:element name="X509SubjectName" type="string"/>`

5.3.43 element ds:X509DataType/X509Certificate



namespace <http://www.w3.org/2000/09/xmldsig#>

type **xs:base64Binary**

properties isRef 0
content simple

source `<xs:element name="X509Certificate" type="base64Binary"/>`

5.3.44 element ds:X509DataType/X509CRL



namespace <http://www.w3.org/2000/09/xmldsig#>

type **xs:base64Binary**

properties isRef 0
content simple

source `<xs:element name="X509CRL" type="base64Binary"/>`

5.3.45 element ds:X509IssuerSerialType/X509IssuerName



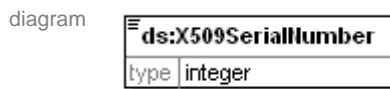
namespace <http://www.w3.org/2000/09/xmldsig#>

type **xs:string**

properties isRef 0
content simple

source `<xs:element name="X509IssuerName" type="string"/>`

5.3.46 element ds:X509IssuerSerialType/X509SerialNumber



namespace <http://www.w3.org/2000/09/xmldsig#>

type **xs:integer**

properties isRef 0
content simple

```
source <xs:element name="X509SerialNumber" type="integer"/>
```