TCG Infrastructure Working Group Core Integrity Schema Specification

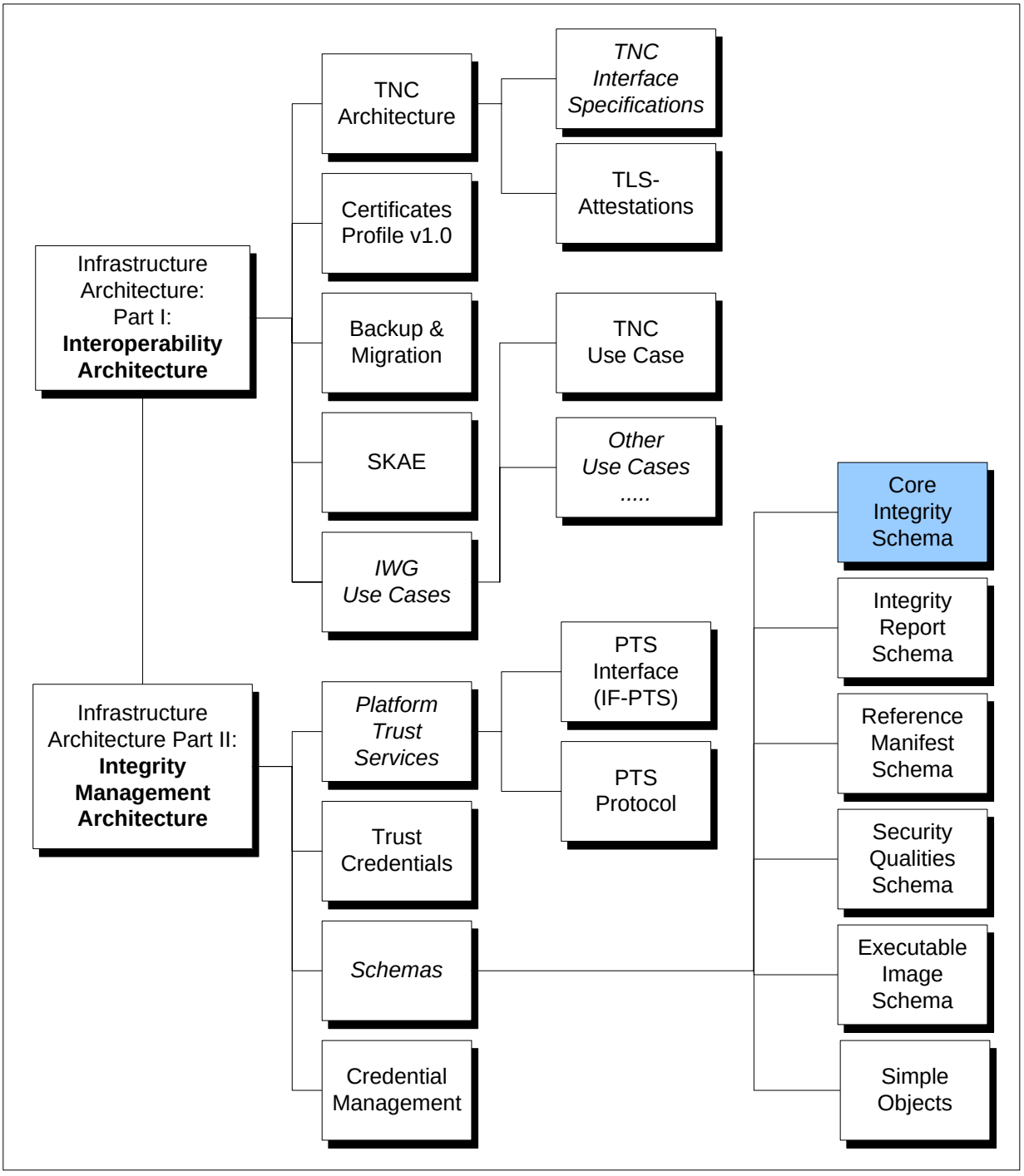
Specification Version 2.0
Revision 5
August 24, 2011
PUBLISHED

Contact:
admin@trustedcomputinggroup.org

TCG Published

TCG

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
David Bleckman	Signacert
Rene Bourquin	General Dynamics C4 Systems
Mike Boyle	US Government
Carlin Covey	Freescale Semiconductor
Malcolm Duncan	CESG
Markus Gueller	Infineon
Thomas Hardjono	MIT
Wyllys Ingersoll (Editor, IWG Co-chair)	Oracle
Greg Kazmierczak	Wave Systems
Scott Kelly	Hyperthought
Carolin Latze	89grad GmbH
Jeff Nisewanger	Sun
Gilles Peskine	Gemalto SA
Mark Redman	Freescale Semiconductor
Paul Sangster (IWG Co-chair)	Symantec
Gloria Serrao	US Government
Ned Smith	Intel Corporation
Adrian Stanger	US Government
Lee Terrell	IBM
Len Veil	Wave Systems
Lee Wilson	IBM

Table of Contents

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* **Error! Reference source not found.**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.

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.

Revision 2.0 of this document contains new mandatory attributes in the ComponentIDType description to facilitate more functional-oriented attestation support. The functional information can be used by the PTS to associate attestation evidence returned for the actual components operating on the platform with the more abstract functional naming used in the attestation protocol.

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 2.0 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/2_0/core_integrity#`

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 v2_0.xsd**
attribute form default: **unqualified**
element form default: **qualified**
targetNamespace: **http://www.trustedcomputinggroup.org/XML/SCHEMA/2_0/core_integrity#**

3.1 Complex Types

Hyperlinks to Complex type Definitions

AssertionType
ComponentIDType
ComponentRefType
ConfidenceValueType
DigestMethodType
DigestValueType
HashType
HashedURIType
IntegrityManifestType
SignerInfoType
PlatformClassType
TransformMethodType
ValueType
VendorIdType

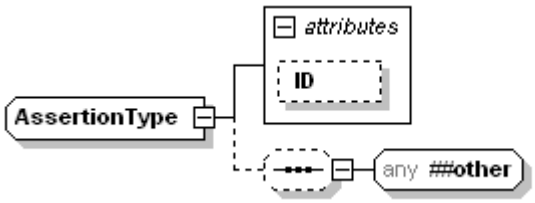
3.1.1 complexType AssertionType

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 **Error! Reference source not found.**schema is an example of an XML schema containing assertions.

3.1.1.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/2_0/core_integrity#

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 **Error! Reference source not found.**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 **Error! Reference source not found.** to capture actual measurements of components that may be extended into PCRs.

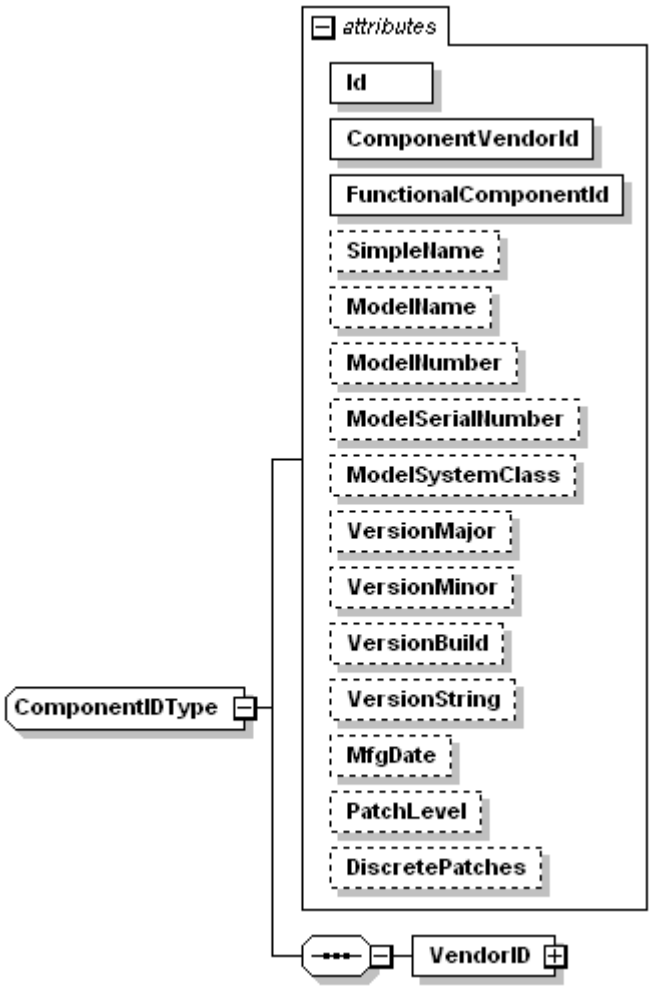
ComponentIDType is a set of attributes accommodating a wide range of change management schemes that when combined uniquely identifies a change-controlled item. The package, program code or logic under change management will have processes for ensuring integrity of its image. VendorID must uniquely identify 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.

Version 2.0 of this specification introduced two new mandatory elements in order to allow the ComponentIDType to be more useful for attestations. The newly added FunctionalComponentID and ComponentVendorID values are required attributes so MUST be present for any Reference Manifest for use with attestation. These values help identify the specific functional component being measured and the version of the reference manifest schema that is being used. If these values are absent, then the 1.0 reference manifest schema is assumed to be in use.

3.1.2.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/2_0/core_integrity#

children **VendorID**

used by
elements **IntegrityManifestType/Collector** **ComponentRefType/ComponentID** **IntegrityManifestType/ComponentID** **SignerInfoType/SigningComponent**

attributes

Name	Type	Use	Default	Fixed
Id	**xs:ID**	required		
ComponentVendorId	**xs:integer**	required		
FunctionalComponentId	**xs:long**	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
ComponentVendorID	24-bit integer SMI value associated with the vendor who is identifying the FunctionalComponentID (functional component) described in the structure. This vendor could differ from the vendor that developed or manufactured the component.
FunctionalComponentID	64-bit Integer ID associated with the functional component (e.g. firewall) described by this

	schema instance
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="ComponentVendorId" type="xs:integer" use="required"/>
  <xs:attribute name="FunctionalComponentId" type="xs:long" 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

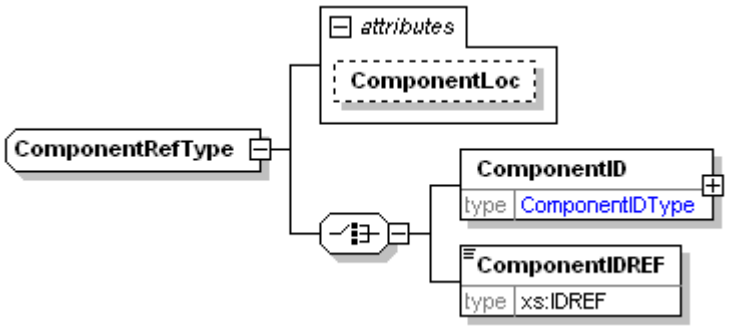
3.1.3.1 Description

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/2_0/core_integrity#

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:

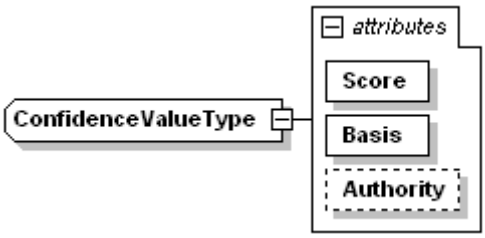
- *Score* – The confidence points given to an assertion. The score must be greater than or equal to 0, and less than or equal to the specified basis.
- *Basis* – The maximum number of confidence *points* that may be given to an assertion. The basis must be an integer greater than 0.
- *Authority* – The 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

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/2_0/core_integrity#

used by element element IntegrityManifestType/ConfidenceValue

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>
```

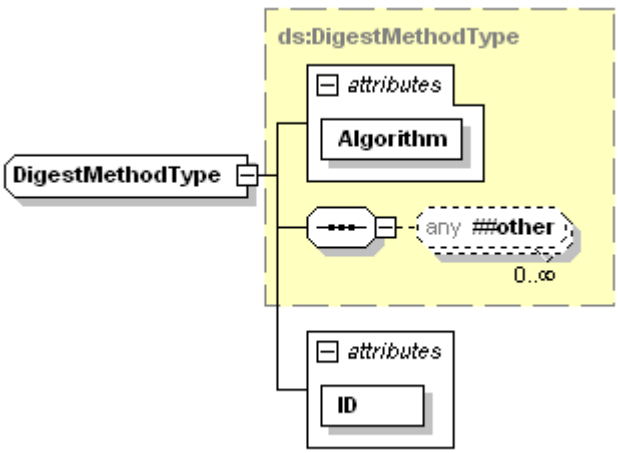
3.1.5 complexType DigestMethodType

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/2_0/core_integrity#

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/xmldsig#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

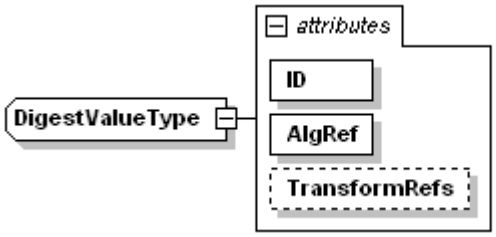
3.1.6.1 Description

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/2_0/core_integrity#

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

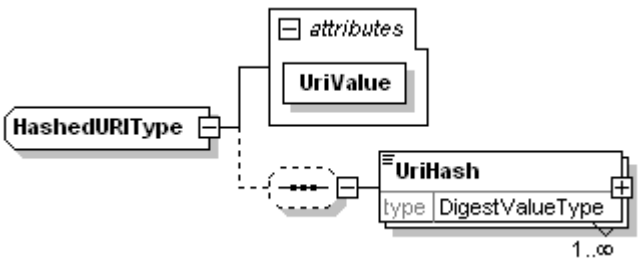
3.1.7.1 Description

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

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/2_0/core_integrity#

children **UriHash**

attributes

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

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="HashedURIType">
 <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>
```

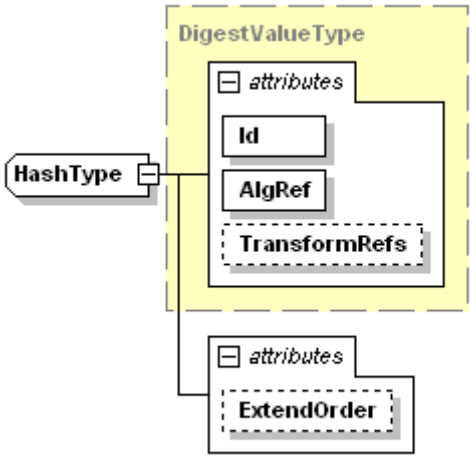
3.1.8 complexType HashType

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

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/2_0/core_integrity#

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.

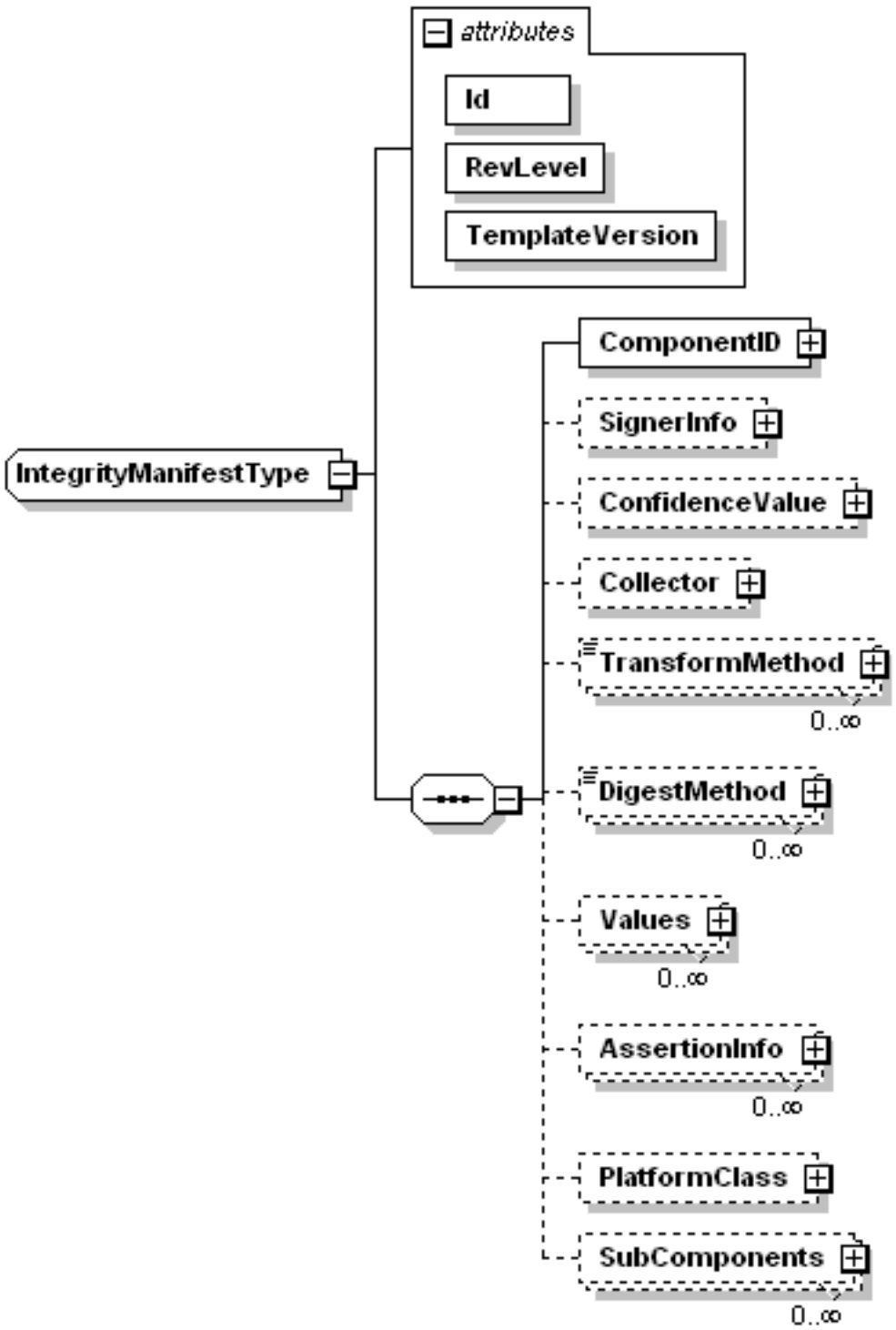
Elements of IntegrityManifestType include:

- 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 the 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/2_0/core_integrity#

properties abstract true

children ComponentID SignerInfo ConfidenceValue Collector TransformMethod DigestMethod Values AssertionInfo PlatformClass SubComponents

attributes

Name	Type	Use	Default	Fixed
Id	**xs:ID**	required		
RevLevel	**xs:integer**	required		
TemplateVersion	**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 *this* 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.
TemplateVersion	TemplateVersion indicates the revision of the structure of this Integrity Manifest. All new Integrity Manifests should use a 1 in this field and each time the structure of the Integrity Manifest changes, this field should be incremented. The structure indicates the organization of component/sub-component hierarchy described by the document. If a new component is added or the sub-component set changes this reflects a different organization of the document. This field is used during attestation to detect when the two parties have the same Integrity Manifest so it can be used as the basis for organizing the resulting Integrity Report.

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:attribute name="TemplateVersion" type="xs:integer" use="required"/>
</xs:complexType>
```

3.1.10 complexType SignerInfoType

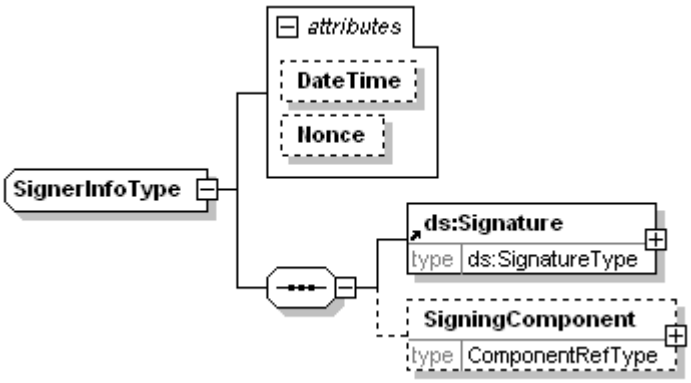
3.1.10.1 Description

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/2_0/core_integrity#

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

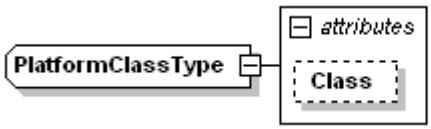
PlatformClassType enumerates platform classifications as determined by the Trusted Computing Group (TCG). Platform classifications can be used to apply platform specific interpretations of integrity values and quality assertions.

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

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/2_0/core_integrity#

used by element **IntegrityManifestType/PlatformClass**

attributes

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

3.1.11.3 Attribute Detail

Component	Description
Class	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. 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. TCG defined Class Identifiers: http://www.trustedcomputinggroup.org/XML/SCHEMA/2_0/core_integrity#PC_CLIENT_X86_BIOS Signifyies an x86 based system with BIOS based firmware. http://www.trustedcomputinggroup.org/XML/SCHEMA/2_0_1/core_integrity#PC_CLIENT_X86_EFI Signifyies an x86 based system with EFI based firmware.

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

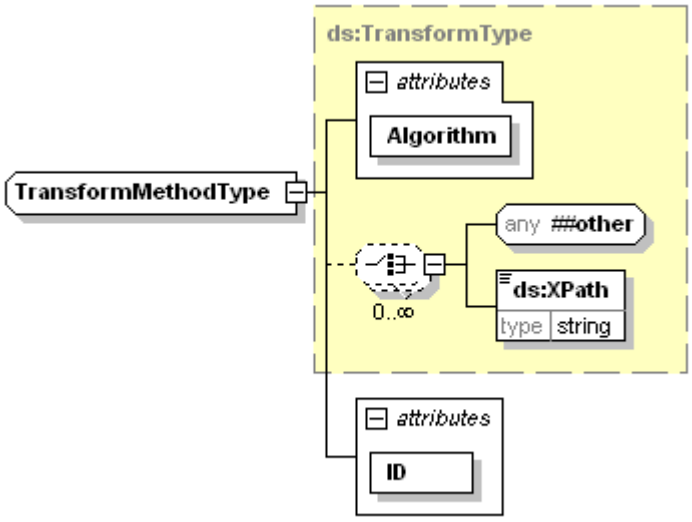
3.1.12.1 Description

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.

3.1.12.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/2_0/core_integrity#

type extension of **ds:TransformType**

properties base ds:TransformType

children **ds:XPath**

used by element **IntegrityManifestType/TransformMethod**

attributes

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

3.1.12.3 Attribute Detail

Component	Description
Algorithm	URI pointing to a transformation algorithm identifier
Id	An identifier unique to *this* document

3.1.12.4 XML

source

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

3.1.13 complexType ValueType

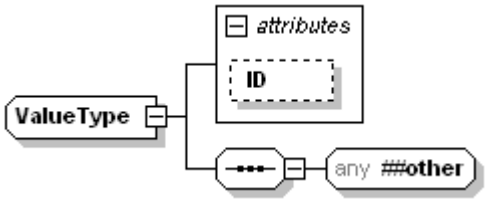
3.1.13.1 Description

ValueType consist of a record identifier and an element **any##other**. It is anticipated that another schema defines integrity measurements to be included in the parent element. The TCG Simple Object **Error! Reference source not found.**schema is an example of an XML document containing integrity values.

The Id attribute is used to uniquely identify instances of child elements included in *this* document.

3.1.13.2 Diagram

diagram



namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/2_0/core_integrity#

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

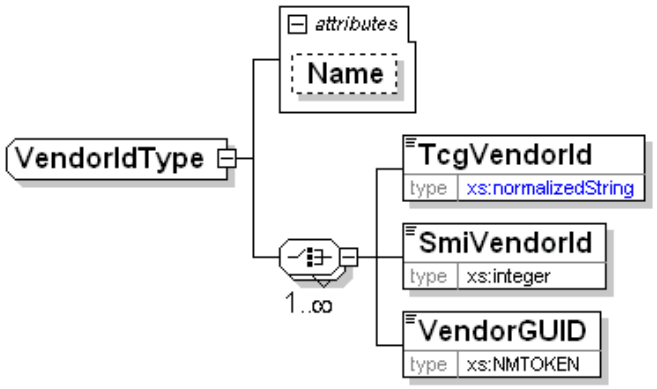
The VendorIdType is used to uniquely identify a vendor, manufacturer or other entity. There are two elements (SmiVendorId and TcgVendorId) that have managed number spaces ensuring uniqueness. VendorGUID uniqueness is derived algorithmically.

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/2_0/core_integrity#

children **TcgVendorId** **SmiVendorId** **VendorGUID**

used by element **ComponentIDType/VendorID**

attributes

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

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

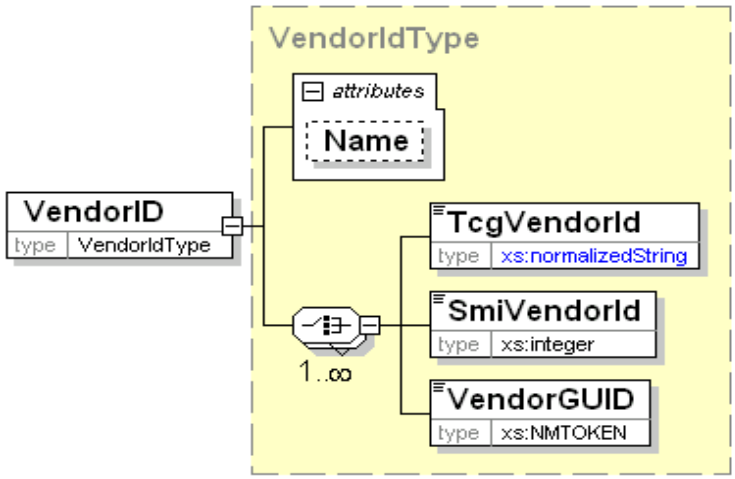
3.2.1 element ComponentIDType/VendorID

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/2_0/core_integrity#				
type	**VendorIdType**				
properties	isRef 0 content complex				
children	**TcgVendorId** **SmiVendorId** **VendorGUID**				
attributes	Name	Type	Use	Default	Fixed
	Name	**xs:string**			

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

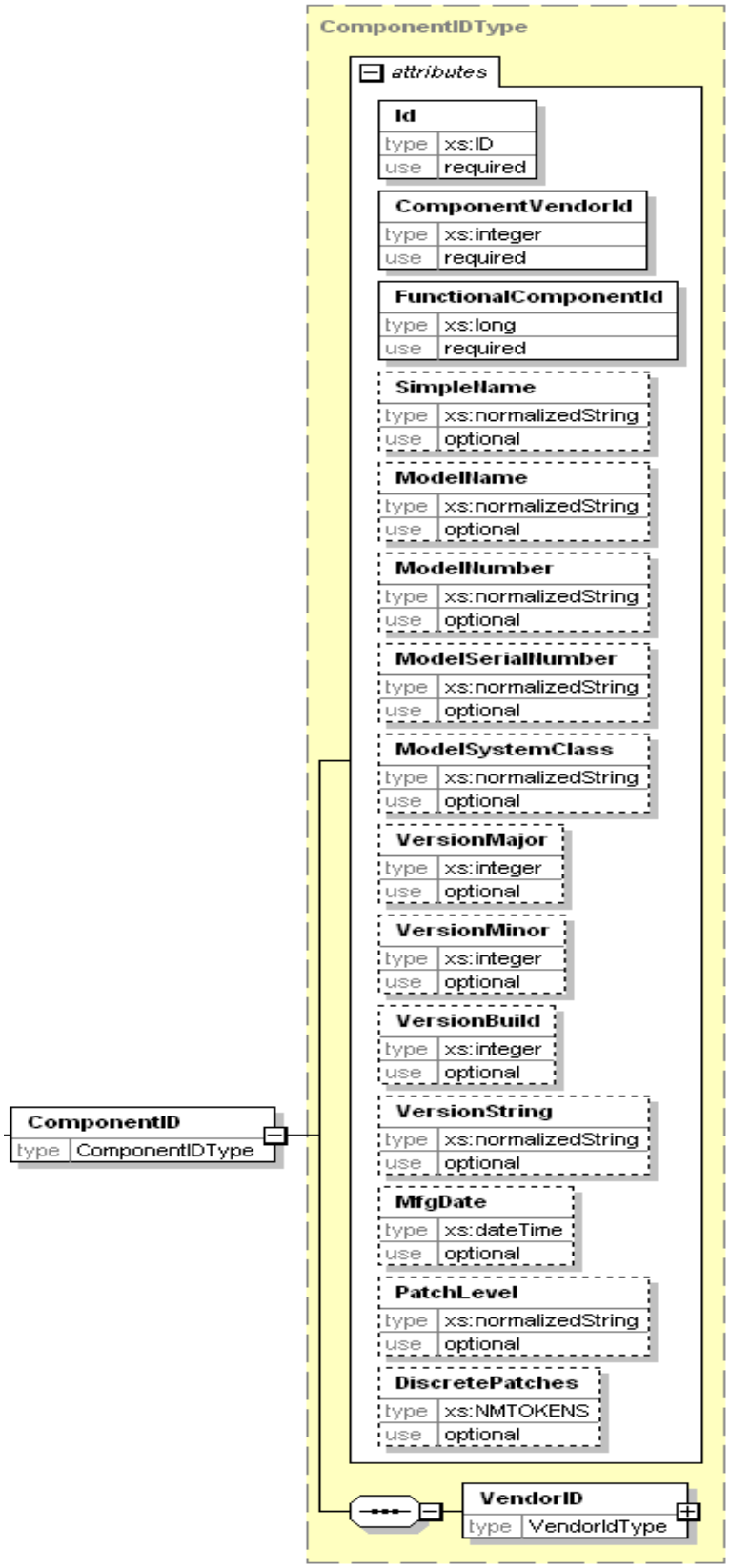
3.2.1.1 Description

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/2_0/core_integrity#

type **ComponentIDType**

properties
- isRef 0
- content complex

children **VendorID**

attributes

Name	Type	Use	Default	Fixed
Id	**xs:ID**	required		
ComponentVendorID	**xs:integer**	required		
FunctionalComponentID	**xs:long**	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

diagram

namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/2_0/core_integrity#

type **xs:IDREF**

properties
- isRef 0
- content simple

3.2.3.3 XML

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

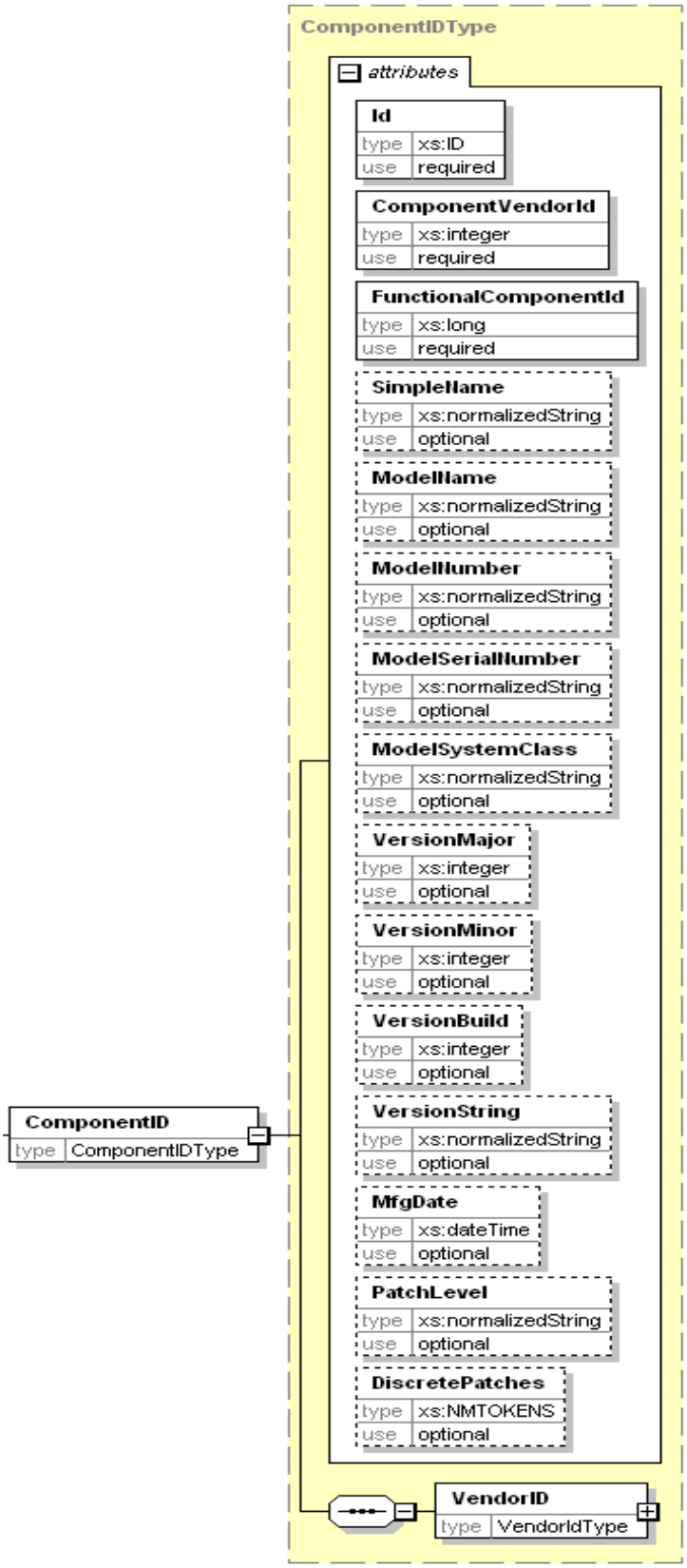
3.2.4 element IntegrityManifestType/ComponentID

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/2_0/core_integrity#

type **ComponentIDType**

properties
- isRef 0
- content complex

children **VendorID**

attributes

Name	Type	Use	Default	Fixed
Id	**xs:ID**	required		
ComponentVendorId	**xs:integer**	required		
FunctionalComponentId	**xs:long**	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

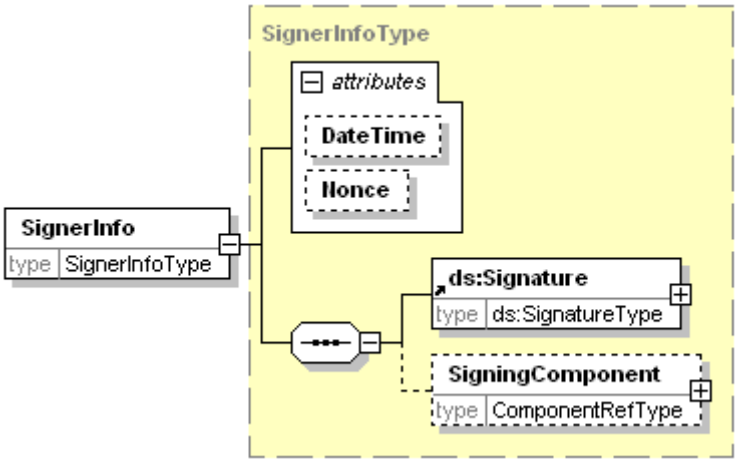
3.2.5.1 Description

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/2_0/core_integrity#

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.

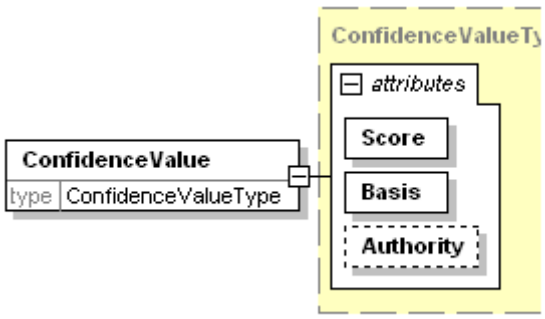
If the signer determines that confidence can be described in terms of levels and there are four possible levels then the first level could have a score of (1) with a basis of (4). Alternatively, a score of (25) would have a basis of (100).

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/2_0/core_integrity#

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"/>
```

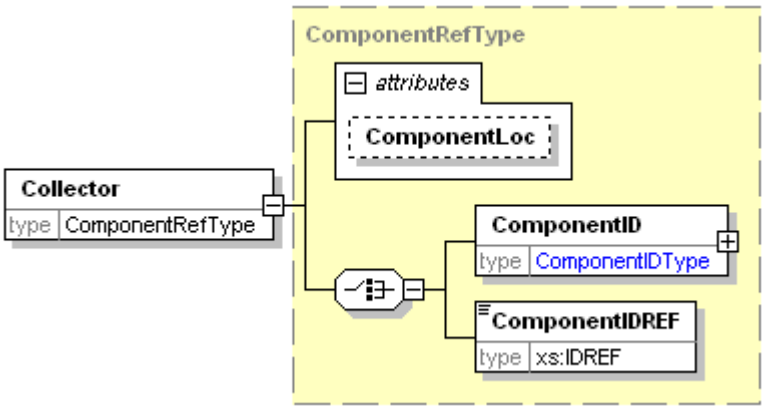
3.2.7 element IntegrityManifestType/Collector

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/2_0/core_integrity#

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" />
```

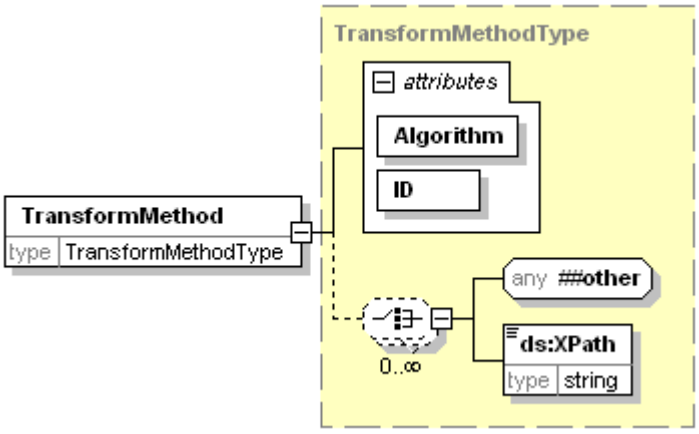
3.2.8 element IntegrityManifestType/TransformMethod

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/2_0/core_integrity#

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"/>
```

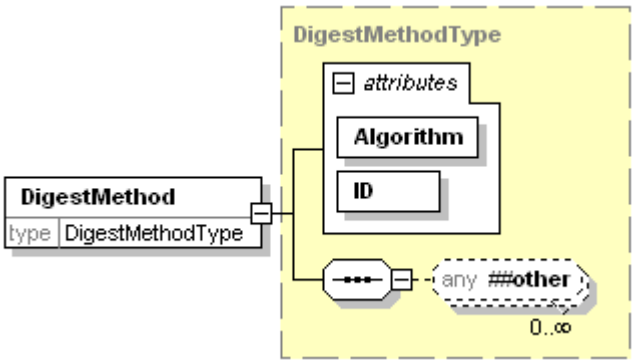
3.2.9 element IntegrityManifestType/DigestMethod

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/2_0/core_integrity#

type **DigestMethodType**

properties
isRef 0
content complex

attributes

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

3.2.9.3 XML

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

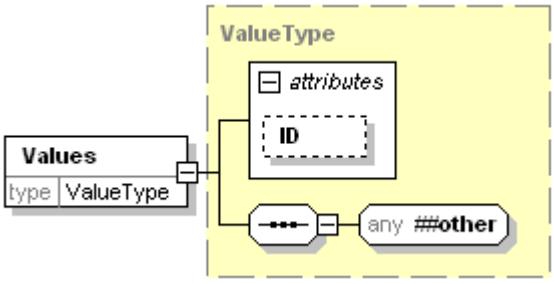
3.2.10 element IntegrityManifestType/Values

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/2_0/core_integrity#

type **ValueType**

properties
isRef 0
content complex

attributes

Name	Type	Use	Default	Fixed
Id	**xs:ID**			

3.2.10.3 XML

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

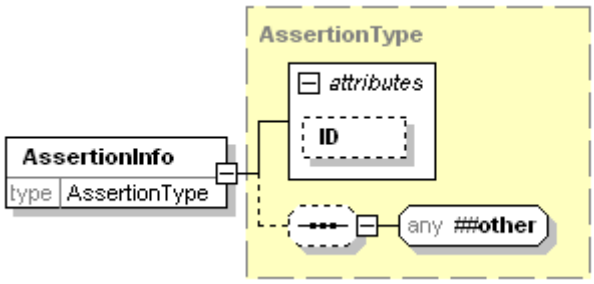
3.2.11 element IntegrityManifestType/AssertionInfo

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/2_0/core_integrity#

type **AssertionType**

properties
isRef 0
content complex

attributes

Name	Type	Use	Default	Fixed
Id	**xs:ID**			

3.2.11.3 XML

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

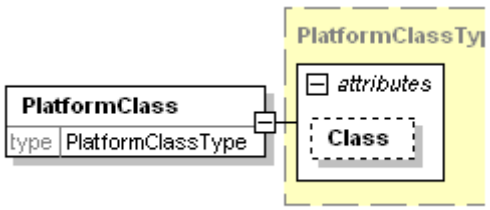
3.2.12 element IntegrityManifestType/PlatformClass

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/2_0/core_integrity#

type **PlatformClassType**

properties
isRef 0
content complex

attributes

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

3.2.12.3 XML

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

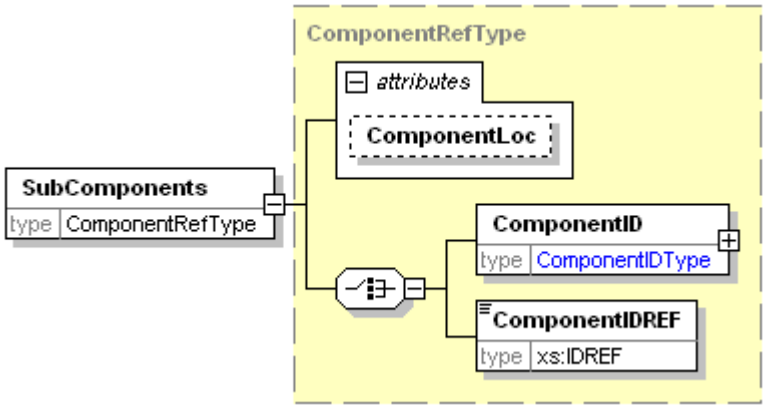
3.2.13 element IntegrityManifestType/SubComponents

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/2_0/core_integrity#

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

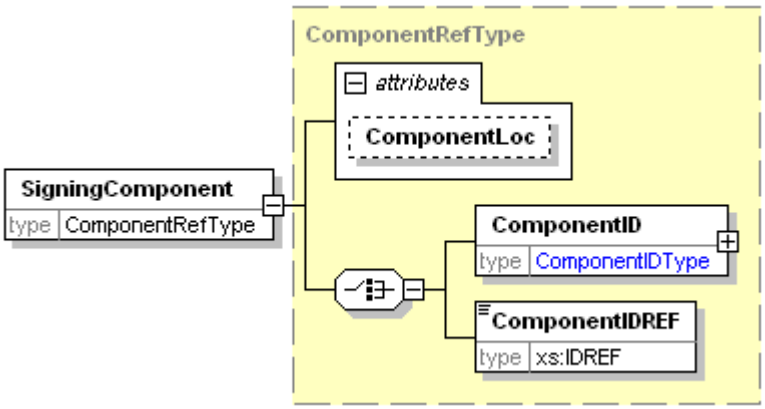
3.2.14.1 Description

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/2_0/core_integrity#

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/2_0/core_integrity#

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

diagram

namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/2_0/core_integrity#

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

diagram

namespace http://www.trustedcomputinggroup.org/XML/SCHEMA/2_0/core_integrity#

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] Trusted Computing Group, Integrity Management Architecture (Architecture Part 2), Specification Version 1.0, TCG Published, September 2006.

[2] Trusted Computing Group, Reference Manifest Schema, Specification Version 2.0, TCG Published, September, 2011.

[3] Trusted Computing Group, Integrity Report Schema, Specification Version 2.0, TCG Published, September, 2011

[4] Trusted Computing Group, Simple Object Schema, Specification Version 1.0, TCG Published, September, 2006.

[5] Trusted Computing Group, Security Qualities Schema, Specification Version 1.0, TCG Published, September, 2006.

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.

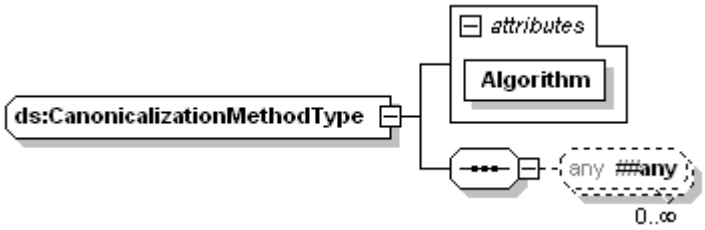
schema location: http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd
attribute form default:
element form default: **qualified**
targetNamespace: **http://www.w3.org/2000/09/xmldsig#**

Elements	Complex types	Simple types
ds:CanonicalizationMethod	ds:CanonicalizationMethodType	ds:CryptoBinary
ds:DigestMethod	ds:DigestMethodType	ds:DigestValueType
ds:DigestValue	ds:DSAKeyValueType	ds:HMACOutputLengthType
ds:DSAKeyValue	ds:KeyInfoType	
ds:KeyInfo	ds:KeyValueType	
ds:KeyName	ds:ManifestType	
ds:KeyValue	ds:ObjectType	
ds:Manifest	ds:PGPDataType	
ds:MgmtData	ds:ReferenceType	
ds:Object	ds:RetrievalMethodType	
ds:PGPData	ds:RSAKeyValueType	
ds:Reference	ds:SignatureMethodType	
ds:RetrievalMethod	ds:SignaturePropertiesType	
ds:RSAKeyValue	ds:SignaturePropertyType	
ds:Signature	ds:SignatureType	
ds:SignatureMethod	ds:SignatureValueType	
ds:SignatureProperties	ds:SignedInfoType	
ds:SignatureProperty	ds:SPKIDataType	
ds:SignatureValue	ds:TransformsType	
ds:SignedInfo	ds:TransformType	
ds:SPKIData	ds:X509DataType	
ds:Transform	ds:X509IssuerSerialType	
ds:Transforms		
ds:X509Data		

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		

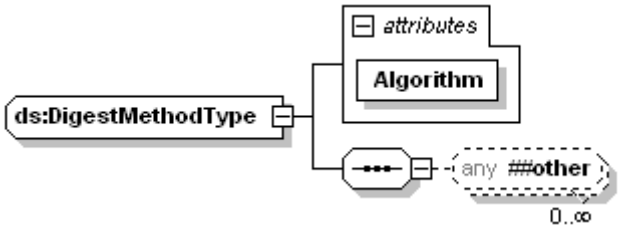
source

```
<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>
```

5.1.2 complexType ds:DigestMethodType

diagram



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

properties mixed true

used by element **ds:DigestMethod**
complexType **DigestMethodType**

attributes

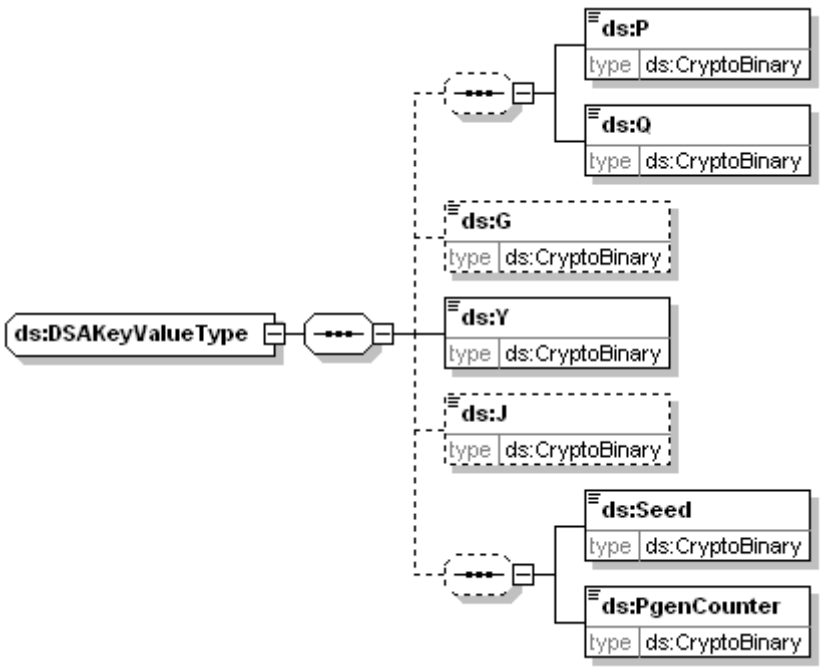
Name	Type	Use	Default	Fixed
Algorithm	**xs:anyURI**	required		

source

```
<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>
```

5.1.3 complexType ds:DSAKeyValueType

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**

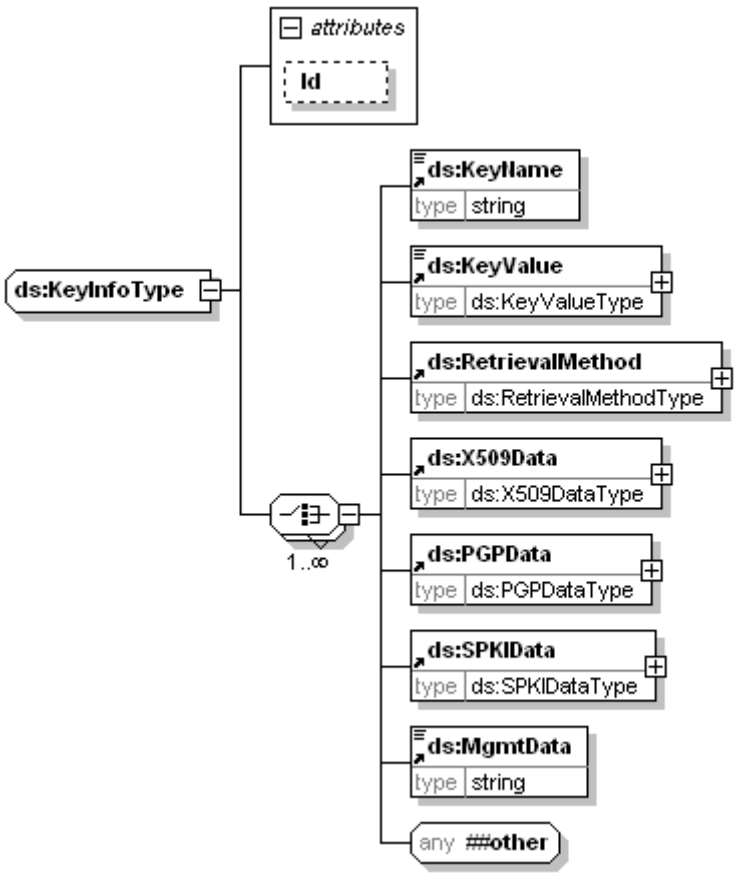
source

```
<xs:complexType name="DSAKeyValueType">
  <xs:sequence>
```

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

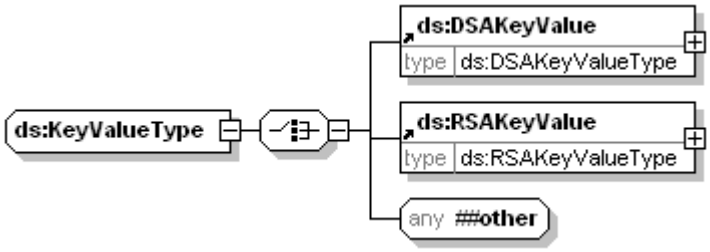
source

```
<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:KeyValueType

diagram



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

properties mixed true

children **ds:DSAKeyValue** **ds:RSAKeyValue**

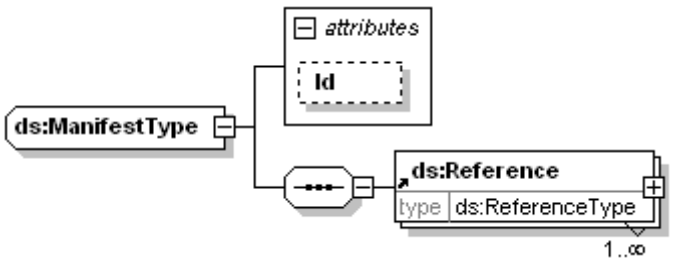
used by element **ds:KeyValue**

source

```
<xs:complexType name="KeyValueType" 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:ManifestType

diagram



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

children **ds:Reference**

used by element **ds:Manifest**

attributes

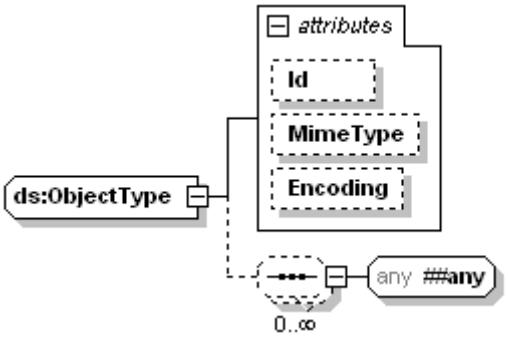
Name	Type	Use	Default	Fixed
Id	**xs:ID**	optional		

source

```
<xs:complexType name="ManifestType">
  <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/xmldsig#

properties mixed true

used by element **ds:Object**

attributes

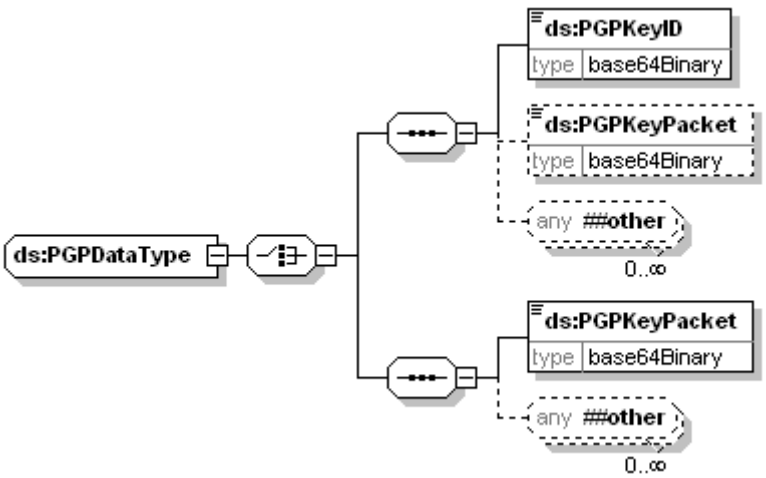
Name	Type	Use	Default	Fixed
Id	**xs:ID**	optional		
MimeType	**xs:string**	optional		
Encoding	**xs:anyURI**	optional		

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/xmldsig#

children **ds:PGPKeyID** **ds:PGPKeyPacket** **ds:PGPKeyPacket**

used by element **ds:PGPData**

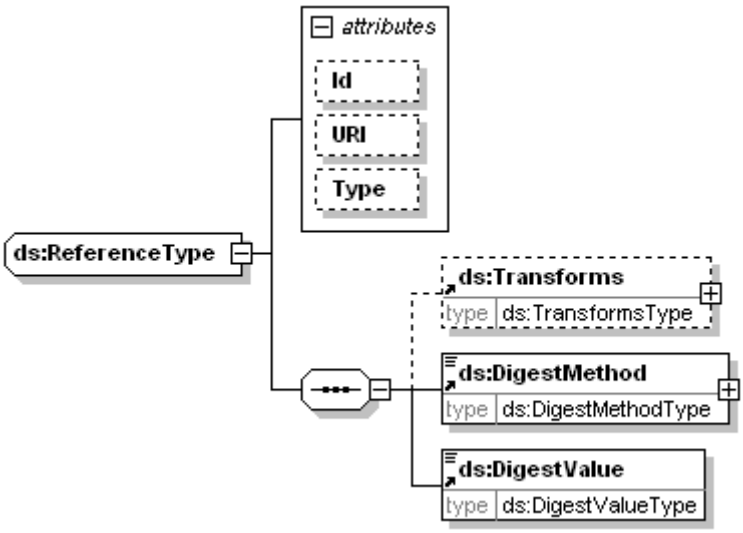
source

```
<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

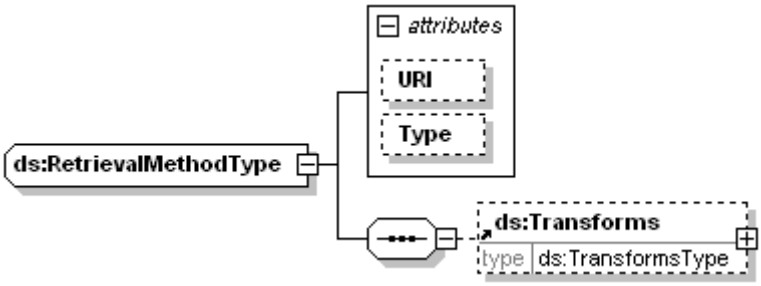
Name	Type	Use	Default	Fixed
Id	**xs:ID**	optional		
URI	**xs:anyURI**	optional		
Type	**xs:anyURI**	optional		

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

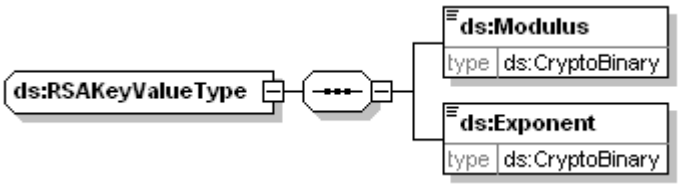
Name	Type	Use	Default	Fixed
URI	**xs:anyURI**			
Type	**xs:anyURI**	optional		

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**

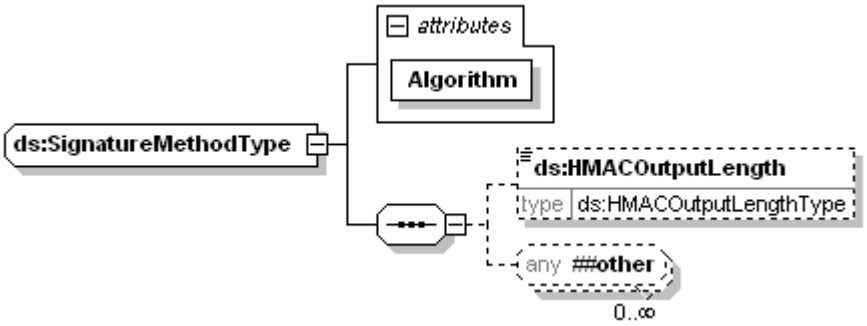
used by element **ds:RSAKeyValue**

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/xmldsig#

properties
mixed true

children **ds:HMACOutputLength**

used by
element **ds:SignatureMethod**

attributes

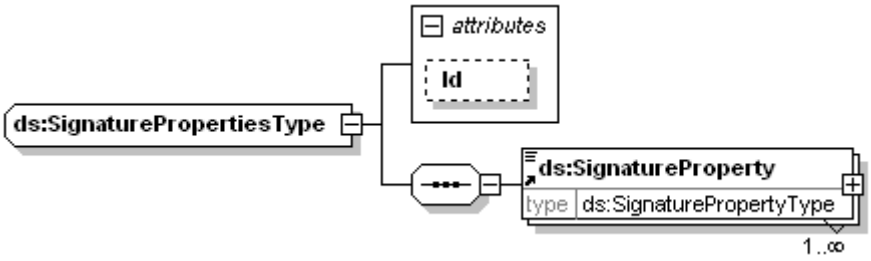
Name	Type	Use	Default	Fixed
Algorithm	**xs:anyURI**	required		

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/xmldsig#

children **ds:SignatureProperty**

used by
element **ds:SignatureProperties**

attributes

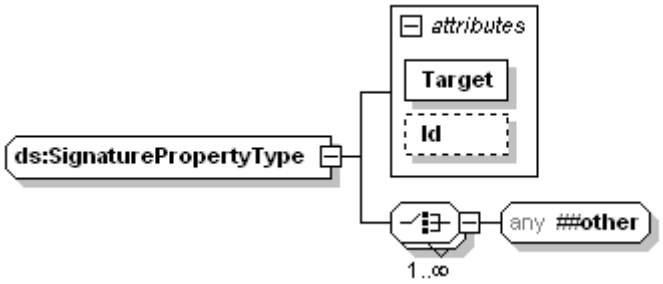
Name	Type	Use	Default	Fixed
Id	**xs:ID**	optional		

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

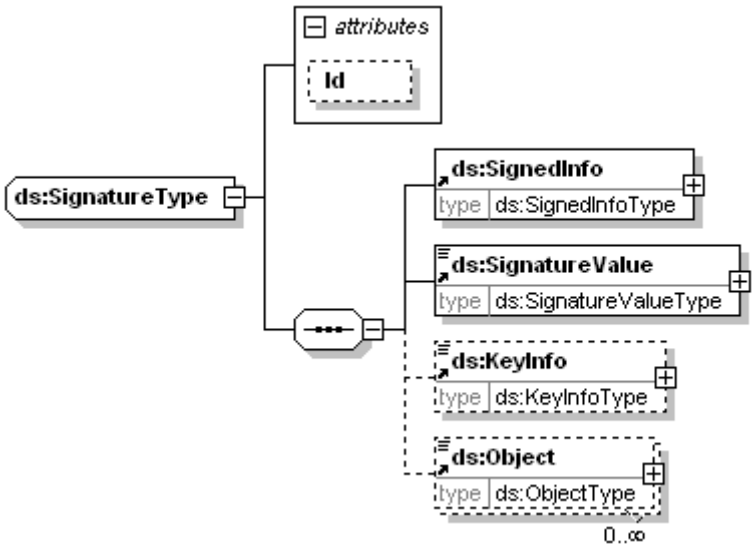
Name	Type	Use	Default	Fixed
Target	**xs:anyURI**	required		
Id	**xs:ID**	optional		

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	Type	Use	Default	Fixed
Id	**xs:ID**	optional		

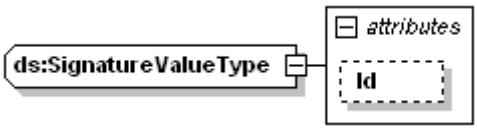
source

```
<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>
```

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

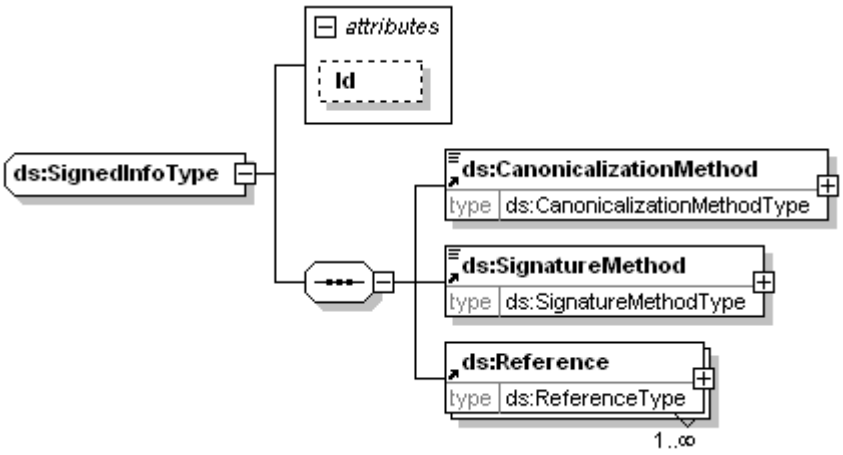
Name	Type	Use	Default	Fixed
Id	**xs:ID**	optional		

source

```
<xs:complexType name="SignatureValueType">
 <xs:simpleContent>
  <xs:extension base="base64Binary">
   <xs:attribute name="Id" type="ID" use="optional"/>
  </xs:extension>
 </xs:simpleContent>
</xs:complexType>
```

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		

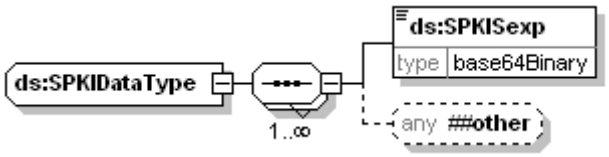
source

```
<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>
```

5.1.18 complexType ds:SPKIDataType

diagram



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

children **ds:SPKISexp**

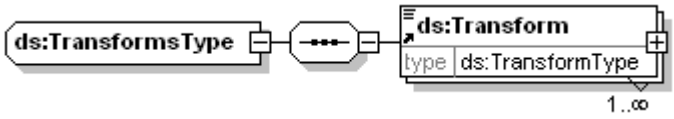
used by element **ds:SPKIData**

source

```
<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/xmldsig#

children **ds:Transform**

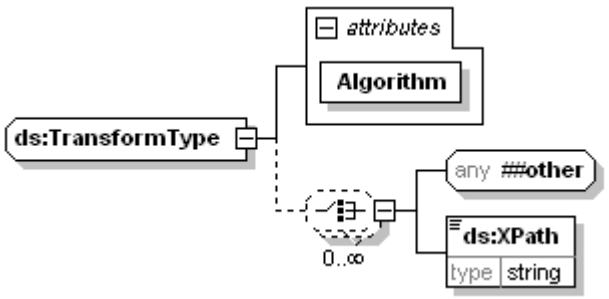
used by element **ds:Transforms**

source

```
<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

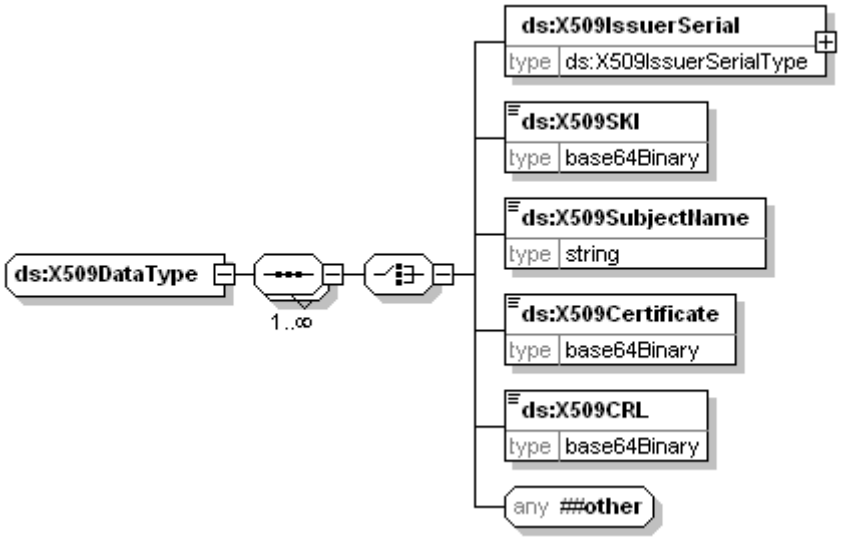
Name	Type	Use	Default	Fixed
Algorithm	**xs:anyURI**	required		

source

```
<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**

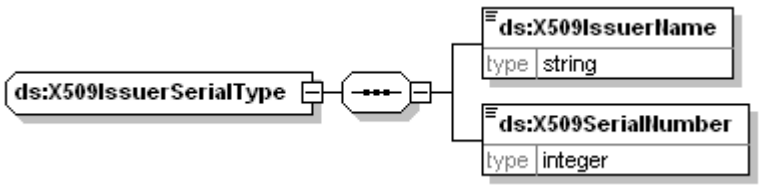
used by element **ds:X509Data**

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:RSAKeyValueType/Exponent** **ds:DSAKeyValueType/G** **ds:DSAKeyValueType/J** **ds:RSAKeyValueType/Modulus** **ds:DSAKeyValueType/P** **ds:DSAKeyValueType/PgenCounter** **ds:DSAKeyValueType/Q** **ds:DSAKeyValueType/Seed** **ds:DSAKeyValueType/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/xmldsig#

type **xs:integer**

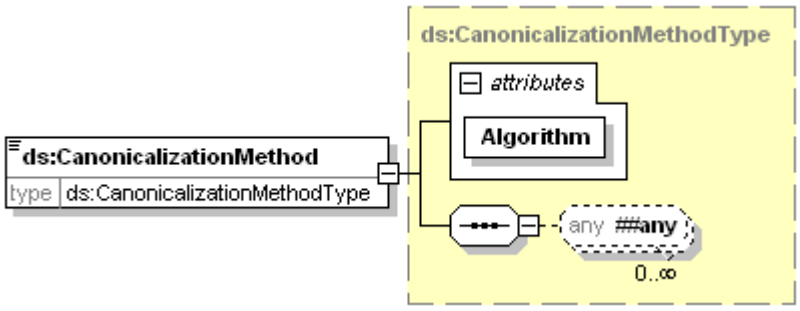
used by element **ds:SignatureMethodType/HMACOutputLength**

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/xmldsig#

type **ds:CanonicalizationMethodType**

properties
content complex
mixed true

used by
complexType **ds:SignedInfoType**

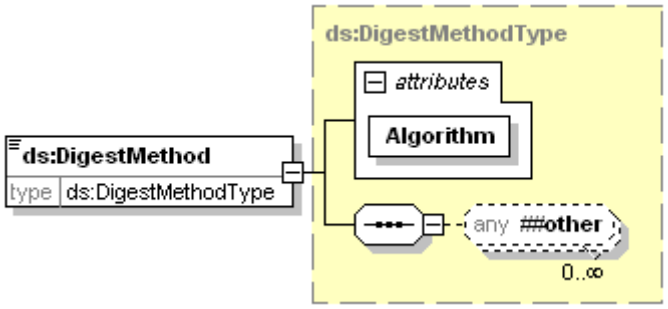
attributes

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/xmldsig#

type **ds:DigestMethodType**

properties
content complex
mixed true

used by
complexType **ds:ReferenceType**

attributes

Name	Type	Use	Default	Fixed
Algorithm	**xs:anyURI**	required		

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

5.3.3 element ds:DigestValue

diagram

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

type **ds:DigestValueType**

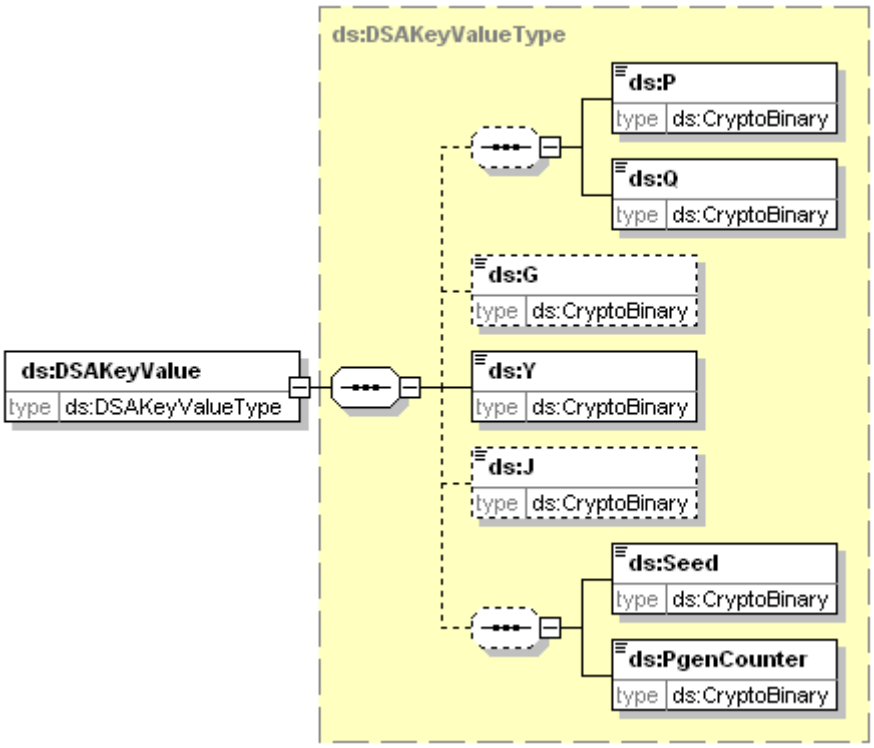
properties content simple

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:DSAKeyValueType**

properties content complex

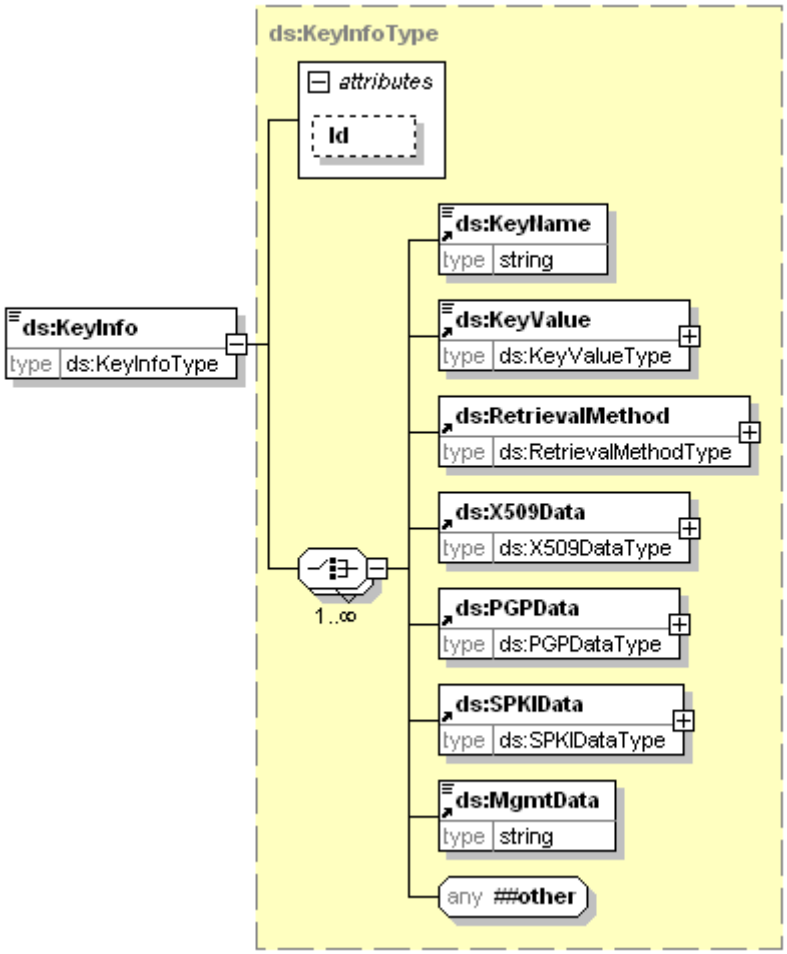
children **ds:P** **ds:Q** **ds:G** **ds:Y** **ds:J** **ds:Seed** **ds:PgenCounter**

used by complexType **ds:KeyValueType**

source <xs:element name="DSAKeyValue" type="ds:DSAKeyValueType"/>

5.3.5 element ds:KeyInfo

diagram



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

type **ds:KeyInfoType**

properties
- content complex
- mixed true

children **ds:KeyName** **ds:KeyValue** **ds:RetrievalMethod** **ds:X509Data** **ds:PGPData** **ds:SPKIData** **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**

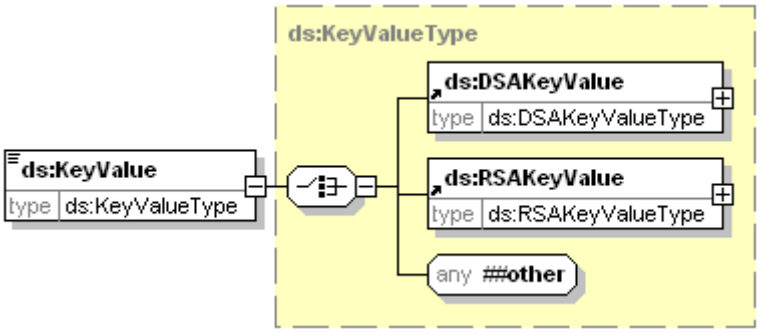
properties
- content simple

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:KeyValueType**

properties
content complex
mixed true

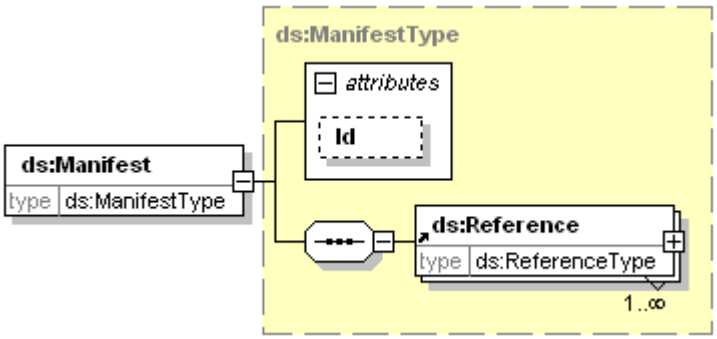
children **ds:DSAKeyValue** **ds:RSAKeyValue**

used by complexType **ds:KeyInfoType**

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

5.3.8 element ds:Manifest

diagram



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

type **ds:ManifestType**

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**

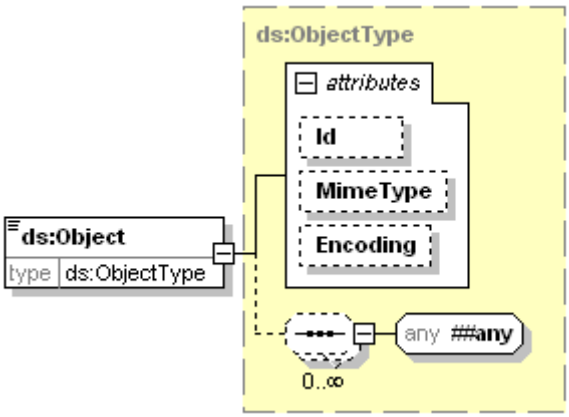
properties
content simple

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**

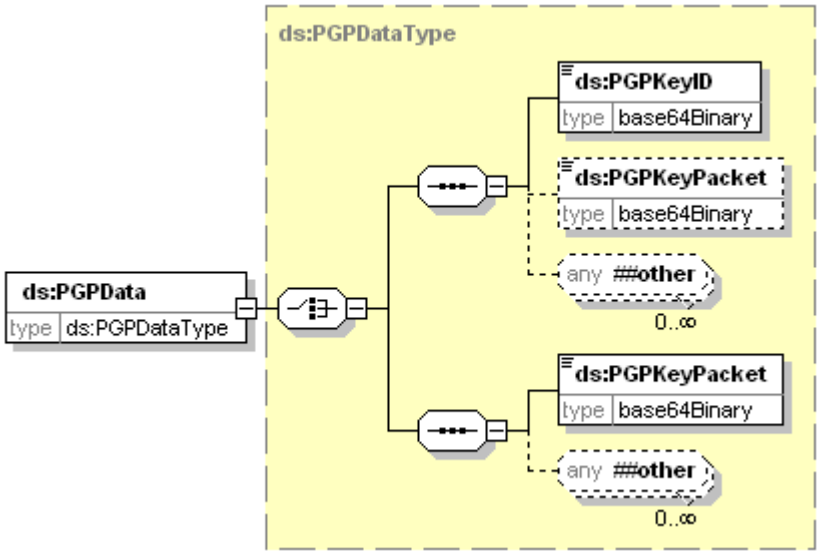
attributes

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

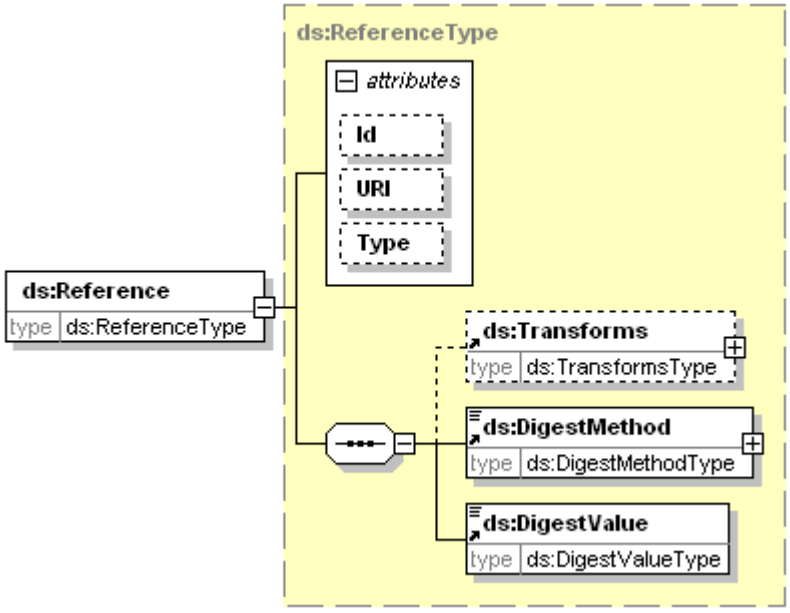
children **ds:PGPKeyID** **ds:PGPKeyPacket** **ds:PGPKeyPacket**

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**

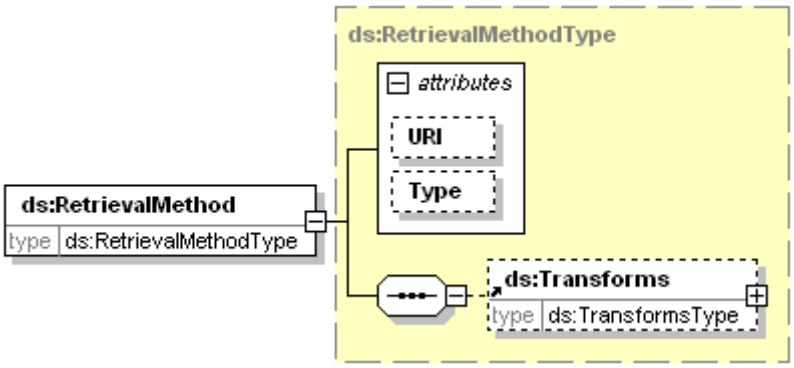
attributes

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**

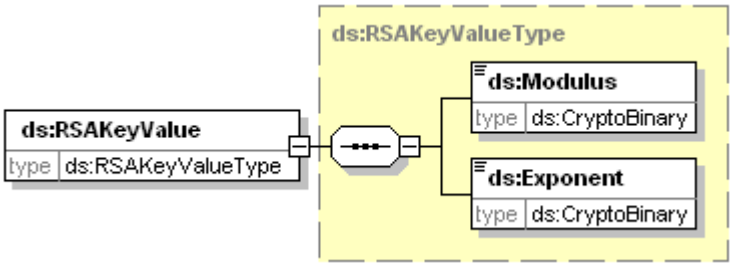
attributes

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:RSAKeyValueType**

properties content complex

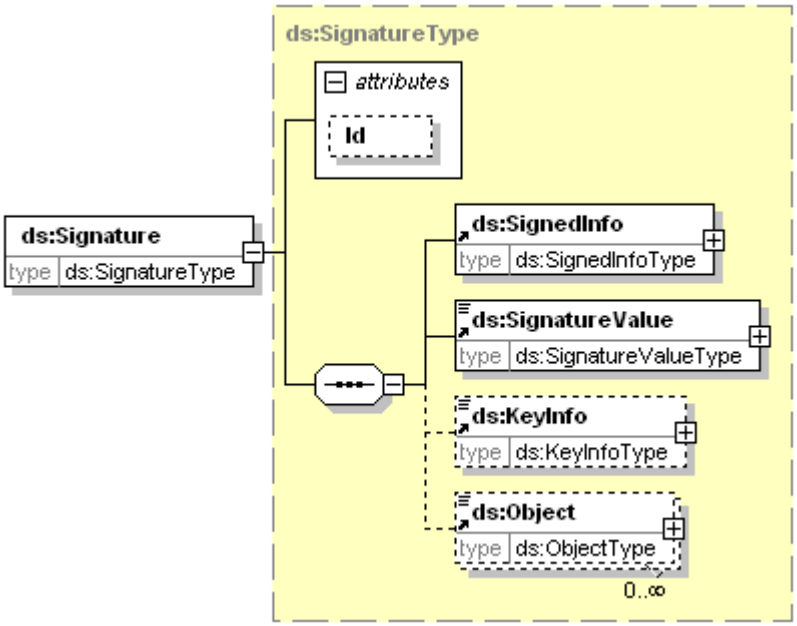
children **ds:Modulus** **ds:Exponent**

used by complexType **ds:KeyValueType**

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

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**

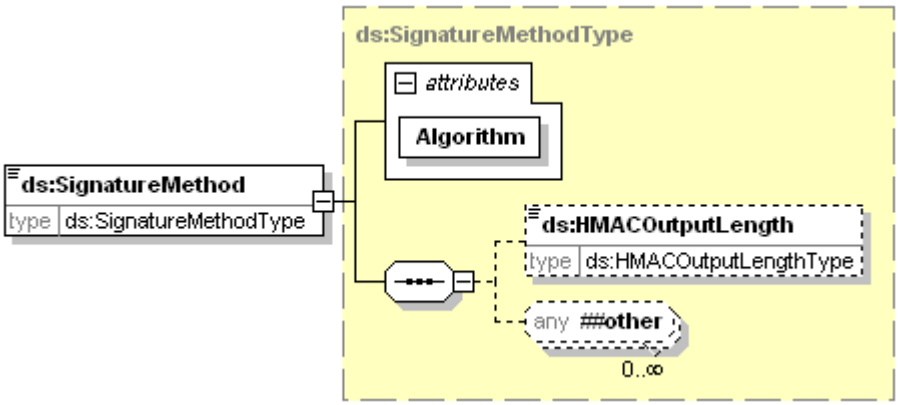
attributes

Name	Type	Use	Default	Fixed
Id	**xs:ID**	optional		

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**

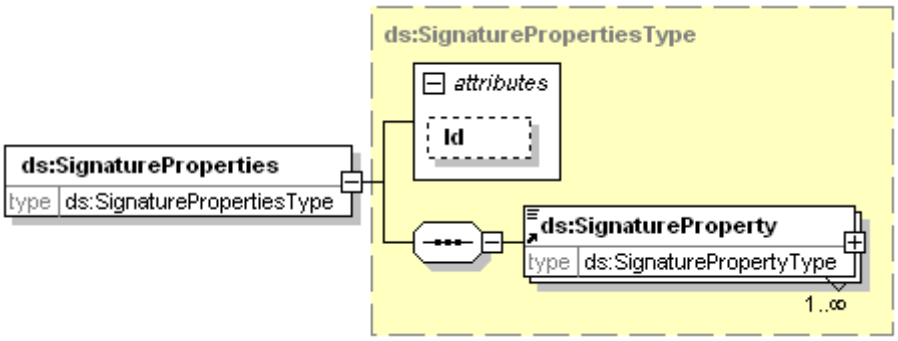
attributes

Name	Type	Use	Default	Fixed
Algorithm	**xs:anyURI**	required		

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

5.3.17 element ds:SignatureProperties

diagram



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

type **ds:SignaturePropertiesType**

properties
content complex

children **ds:SignatureProperty**

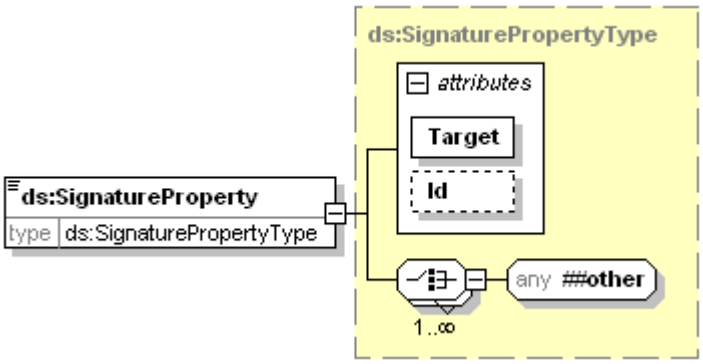
attributes

Name	Type	Use	Default	Fixed
Id	**xs:ID**	optional		

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

5.3.18 element ds:SignatureProperty

diagram



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

type **ds:SignaturePropertyType**

properties
content complex
mixed true

used by complexType **ds:SignaturePropertiesType**

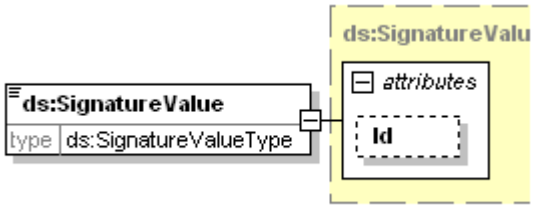
attributes

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

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

5.3.19 element ds:SignatureValue

diagram



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

type **ds:SignatureValueType**

properties content complex

used by complexType **ds:SignatureType**

attributes

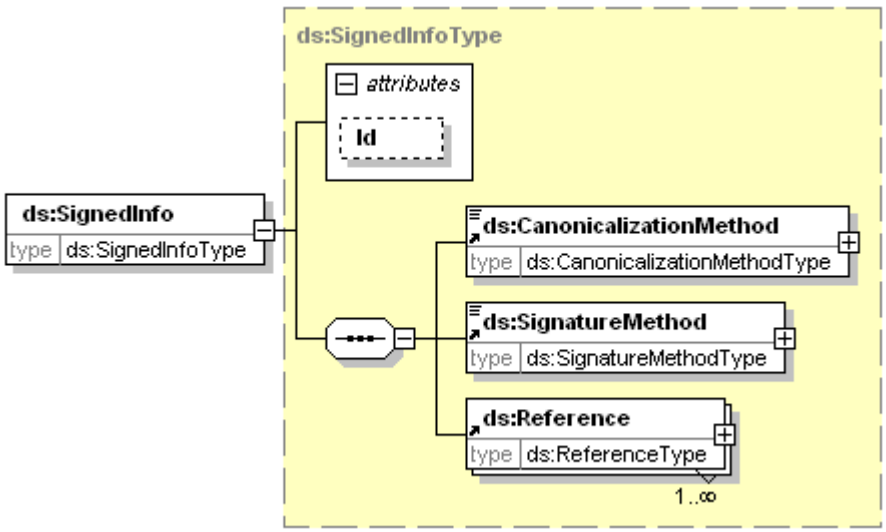
Name	Type	Use	Default	Fixed
Id	**xs:ID**	optional		

source

```
<xs:element name="SignatureValue" type="ds:SignatureValueType"/>
```

5.3.20 element ds:SignedInfo

diagram



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

type **ds:SignedInfoType**

properties content complex

children **ds:CanonicalizationMethod** **ds:SignatureMethod** **ds:Reference**

used by complexType **ds:SignatureType**

attributes

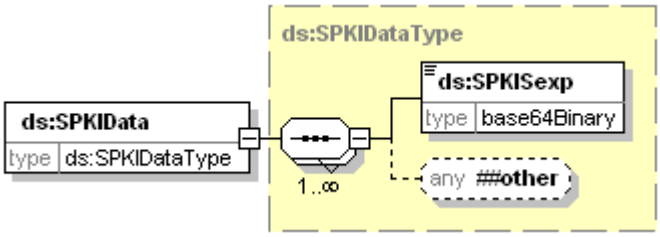
Name	Type	Use	Default	Fixed
Id	**xs:ID**	optional		

source

```
<xs:element name="SignedInfo" type="ds:SignedInfoType"/>
```

5.3.21 element ds:SPKIData

diagram



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

type **ds:SPKIDataType**

properties
content complex

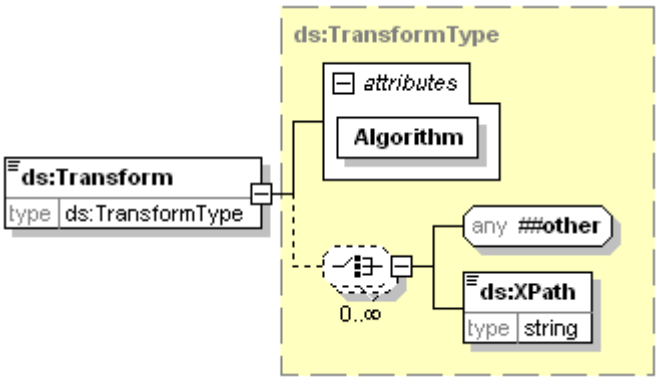
children **ds:SPKISexp**

used by
complexType **ds:KeyInfoType**

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

5.3.22 element ds:Transform

diagram



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

type **ds:TransformType**

properties
content complex
mixed true

children **ds:XPath**

used by
complexType **ds:TransformsType**

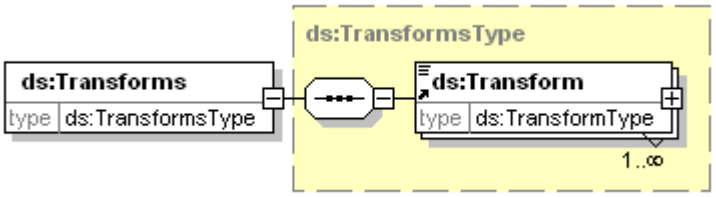
attributes

Name	Type	Use	Default	Fixed
Algorithm	**xs:anyURI**	required		

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

5.3.23 element ds:Transforms

diagram



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

type **ds:TransformsType**

properties content complex

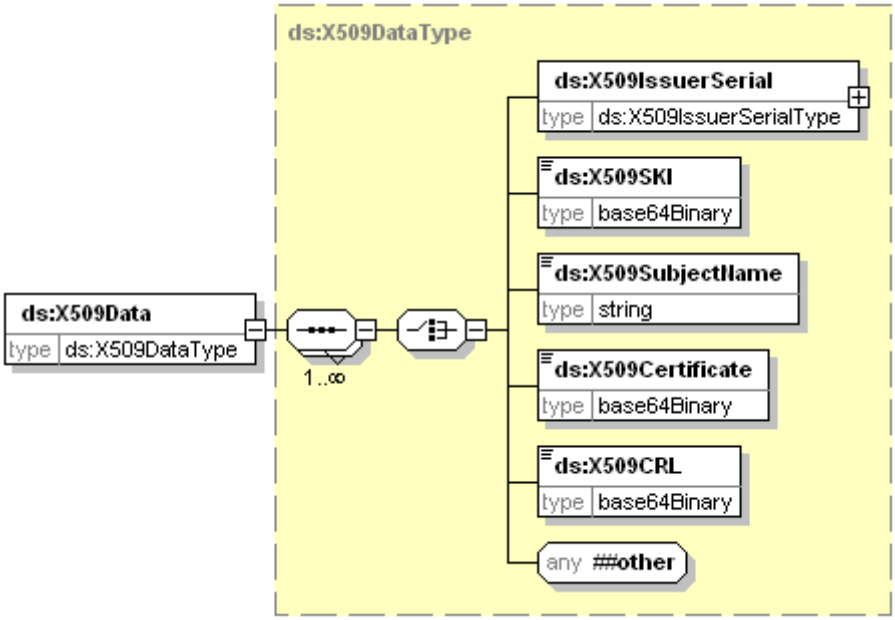
children **ds:Transform**

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:DSAKeyValueType/P

diagram

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:DSAKeyValueType/Q

diagram

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:DSAKeyValueType/G

diagram

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:DSAKeyValueType/Y

diagram

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:DSAKeyValueType/J

diagram

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:DSAKeyValueType/Seed

diagram

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:DSAKeyValueType/PgenCounter

diagram

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

diagram

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

diagram

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

diagram

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:RSAKeyValueType/Modulus

diagram

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:RSAKeyValueType/Exponent

diagram

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

diagram

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

diagram

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

diagram

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

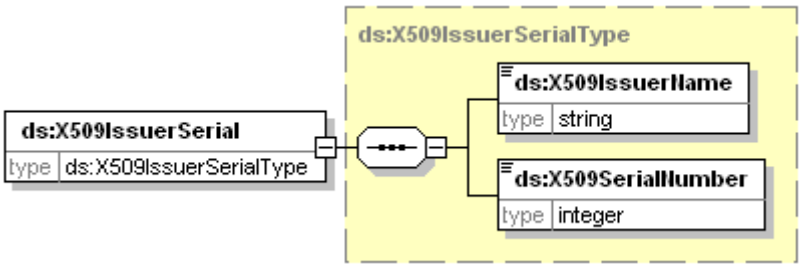
type **xs:string**

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#

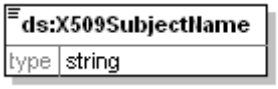
type **xs:base64Binary**

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

diagram

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

diagram

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

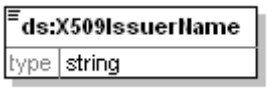
type **xs:base64Binary**

properties isRef 0
content simple

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

5.3.45 element ds:X509IssuerSerialType/X509IssuerName

diagram



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

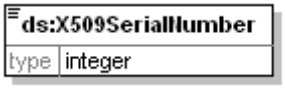
type **xs:string**

properties
isRef 0
content simple

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

5.3.46 element ds:X509IssuerSerialType/X509SerialNumber

diagram



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

type **xs:integer**

properties
isRef 0
content simple

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