



Guide to Schemas for DHZMC-TDF

DHZMC-TDF Schema Guide

Version 2021-NOV

December 1, 2022

Distribution Notice:

This document has been approved for Public Release and is available for use without restriction.

Table of Contents

Chapter 1 - Introduction	1
1.1 - Purpose	1
Chapter 2 - Schema Files	2
2.1 - DHZMC-TDF.xsd	2
2.2 - DHZMC-TDF-guard.xsd	29

Chapter 1 - Introduction

1.1 - Purpose

This is an informative supplement for DHZMC-TDF. This guide is generated from the DHZMC-TDF Schemas and provides a consolidated reference for the schemas of this specification.

Chapter 2 - Schema Files

2.1 - DHZMC-TDF.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
            xmlns="urn:us:gov:ic:tdf"
            xmlns:enc="http://www.w3.org/2001/04/xmlenc#"
            xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
            xmlns:xhtml="http://www.w3.org/1999/xhtml-StopBrowserRendering"
            xmlns:tdfsigal="urn:us:gov:ic:cenum:tdf:signaturealgorithm"
            xmlns:tdfstate="urn:us:gov:ic:cenum:tdf:state"
            xmlns:tdfhashal="urn:us:gov:ic:cenum:tdf:hashalgorithm"
            xmlns:sfhashv="urn:us:gov:ic:sf:hashverification"
            xmlns:sf="urn:us:gov:ic:sf"
            xmlns:dhzm="urn:us:gov:ic:digitalhazmat"
            xmlns:anlys="urn:us:gov:ic:anlysassert"
            targetNamespace="urn:us:gov:ic:tdf"
            elementFormDefault="qualified"
            attributeFormDefault="qualified"
            ism:compliesWith="USGov USIC"
            ism:resourceElement="true"
            ism:createDate="2019-09-18"
            ism:DESVersion="202111"
            ism:ISMCA TCESVersion="202205"
            ism:classification="U"
            ism:ownerProducer="USA"
            version="202111">
  <xs:annotation>
    <xs:documentation>
      <xhtml:h1 ism:ownerProducer="USA" ism:classification="U">Intelligence Community
        Technical Specification XML Data Encoding Specification for DigitalHazMat Commercial TDF
        (DHZMC-TDF.XML)</xhtml:h1>
      </xs:documentation>
      <xs:documentation>
        <xhtml:h2 ism:ownerProducer="USA" ism:classification="U">Notices</xhtml:h2>
        <xhtml:p ism:ownerProducer="USA" ism:classification="U">Distribution Notice:
          This document has been approved for Public Release and is available for use without restriction.
        </xhtml:p>
      </xs:documentation>
      <xs:documentation>
        <xhtml:h2 ism:ownerProducer="USA" ism:classification="U">Description</xhtml:h2>
        <xhtml:p ism:ownerProducer="USA" ism:classification="U">W3C XML Schema for the Intelligence Community
          Technical Specification XML Data Encoding Specification for DigitalHazMat Commercial TDF
          (DHZMC-TDF.XML).</xhtml:p>
      </xs:documentation>
      <xs:documentation>
        <xhtml:h2 ism:ownerProducer="USA" ism:classification="U">Introduction</xhtml:h2>
        <xhtml:p ism:ownerProducer="USA" ism:classification="U">This XML Schema file is one
          component of the XML Data Encoding Specification (DES). Please see the document titled<xhtml:i>
            <xhtml:a href="../../Documents/DHZMC-TDF/DesDhzmctdfXml.pdf">XML Data Encoding Specification for DigitalHazMat Commercial TDF</xhtml:a>
          </xhtml:i>for a complete description of the encoding as well as list of all
          components.</xhtml:p>
        <xhtml:p ism:ownerProducer="USA" ism:classification="U">It is envisioned that this
```

```

schema or its components, as well as other parts of the DES may be overridden for
localized implementations. Therefore, permission to use, copy, modify and distribute
this XML Schema and the other parts of the DES for any purpose is hereby granted in
perpetuity.</xhtml:p>
  <xhtml:p ism:ownerProducer="USA" ism:classification="U">Please reference the preceding
two paragraphs in all copies or variations. The developers make no representation
about the suitability of the schema or DES for any purpose. It is provided "as is"
without expressed or implied warranty.</xhtml:p>
  <xhtml:p ism:ownerProducer="USA" ism:classification="U">If you modify this XML Schema in
any way label your schema as a variant of DHZMC-TDF.XML.</xhtml:p>
  <xhtml:p ism:ownerProducer="USA" ism:classification="U">Please direct all questions, bug
reports,or suggestions for changes to the points of contact identified in the
document referenced above.</xhtml:p>
</xs:documentation>
<xs:documentation>
  <xhtml:h2 ism:ownerProducer="USA" ism:classification="U">Implementation Notes</xhtml:h2>
  <xhtml:p ism:ownerProducer="USA" ism:classification="U">The root element for DHZMC-TDF is:
<xhtml:a href="DHZMC-TDF_xsd_Element_TrustedDataObject.html#TrustedDataObject">tdf:TrustedDataObject</xhtml:a>
  </xhtml:p>
</xs:documentation>
<xs:documentation>
  <xhtml:h2 ism:ownerProducer="USA" ism:classification="U">Creators</xhtml:h2>
  <xhtml:p ism:ownerProducer="USA" ism:classification="U">Office of the Director of
National Intelligence Intelligence Community Chief Information Officer</xhtml:p>
</xs:documentation>
</xs:annotation>
<!-- ***** -->
<!-- Import statements -->
<!-- ***** -->
<!--Replacing:
<xs:import namespace="urn:us:gov:ic:cvenum:tdf:signaturealgorithm" schemaLocation="./CVEGenerated/CVEnumTDFSignatureAlgorithm.xsd"/>
-->
<!-- DigitalHazMatCom: Update import path to BASE-TDF CVEGenerated for CVEnumTDFHashAlgorithm -->
<xs:import namespace="urn:us:gov:ic:cvenum:tdf:signaturealgorithm"
  schemaLocation=" ../BASE-TDF/CVEGenerated/CVEnumTDFSignatureAlgorithm.xsd"/>
  <!--Replacing:
<xs:import namespace="urn:us:gov:ic:cvenum:tdf:hashalgorithm" schemaLocation="./CVEGenerated/CVEnumTDFHashAlgorithm.xsd"/>
-->
<!-- DigitalHazMatCom: Update import path to BASE-TDF CVEGenerated for CVEnumTDFSignatureAlgorithm -->
<xs:import namespace="urn:us:gov:ic:cvenum:tdf:hashalgorithm"
  schemaLocation=" ../BASE-TDF/CVEGenerated/CVEnumTDFHashAlgorithm.xsd"/>
  <!--Replacing:
<xs:import namespace="urn:us:gov:ic:cvenum:tdf:state" schemaLocation="./CVEGenerated/CVEnumTDFAppliesToState.xsd"/>
-->
<!-- DigitalHazMatCom: Update import path to BASE-TDF CVEGenerated for CVEnumTDFAppliesToState -->
<xs:import namespace="urn:us:gov:ic:cvenum:tdf:state"
  schemaLocation=" ../BASE-TDF/CVEGenerated/CVEnumTDFAppliesToState.xsd"/>
  <xs:import namespace="urn:us:gov:ic:sf:hashverification"
  schemaLocation=" ../IC-SF/HashVerification.xsd"/>
  <xs:import namespace="urn:us:gov:ic:sf" schemaLocation=" ../IC-SF/IC-SF.xsd"/>
  <!-- ***** -->
<!-- Elements -->
<!-- ***** -->
<!-- TDC root element -->
```

```
<!-- DigitalHazMatCom: Add DHZM and ANLYS schema imports -->
<xs:import namespace="urn:us:gov:ic:digitalhazmat"
            schemaLocation="../DHZM/DHZM.xsd"/>
<xs:import namespace="urn:us:gov:ic:anlysassert"
            schemaLocation="../ANLYS/ANLYS.xsd"/>
<xs:element name="TrustedDataCollection" type="TrustedDataCollectionType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:ownerProducer="USA" ism:classification="U"> The root element of a
Trusted Data Collection. </xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:element>
<!-- TDO root element -->
<xs:element name="TrustedDataObject" type="TrustedDataObjectType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:ownerProducer="USA" ism:classification="U"> The root element of a
Trusted Data Object. A Trusted Data Collection may contain many Trusted Data
Objects. </xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:element>
<!-- ***** -->
<!-- Attributes -->
<!-- ***** -->
<xs:attribute name="version">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:ownerProducer="USA" ism:classification="U"> The version number of the
DES. </xhtml:p>
    </xs:documentation>
  </xs:annotation>
  <xs:simpleType>
    <xs:restriction base="xs:string">

<!--Replacing:
<xs:pattern value="[0-9]{6}(\.[0-9]{6})?(\-{1,23})?" />
-->
<!-- DigitalHazMatCom: Update tdf:version regex to be for DHZMC-TDF customization -->

    <xs:pattern value="[0-9]{6}(\.[0-9]{6})?\-DHZMC\-TDF\.[0-9]{6}(\.[0-9]{6})?(\-{1,23})?" />
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="mediaType" type="MediaTypeType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA"> An attribute for expressing
the mediaType of an object as defined in <xhtml:a href="http://tools.ietf.org/html/rfc4288">RFC 4288</xhtml:a>. </xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
<xs:attribute name="id" type="xs:ID">
  <xs:annotation>
```

```

        <xs:documentation>
            <xhtml:p ism:ownerProducer="USA" ism:classification="U"> A unique local identifier
            used for binding and signing purposes. Not guaranteed to be unique across
            multiple TDC/TDOs but must be unique within a single instance of
            either.</xhtml:p>
        </xs:documentation>
    </xs:annotation>
</xs:attribute>
<xs:attribute name="idRef" type="xs:IDREF">
    <xs:annotation>
        <xs:documentation>
            <xhtml:p ism:ownerProducer="USA" ism:classification="U"> A reference to a
            unique local identifier.</xhtml:p>
        </xs:documentation>
    </xs:annotation>
</xs:attribute>
<xs:attribute name="filename" type="xs:string">
    <xs:annotation>
        <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA"> This is the filename of the
            payload. </xhtml:p>
        </xs:documentation>
    </xs:annotation>
</xs:attribute>
<xs:attribute name="scope">
    <xs:annotation>
        <xs:documentation>
            <xhtml:p ism:ownerProducer="USA" ism:classification="U"> The grouping of objects to
            which the assertion applies. Please see the "Assertion Scopes" section in the
            DES document for more information. </xhtml:p>
        </xs:documentation>
    </xs:annotation>
<xs:simpleType>
    <xs:restriction base="xs:string">
        <xs:enumeration ism:classification="U" ism:ownerProducer="USA" value="TDO"/>
        <xs:enumeration ism:classification="U" ism:ownerProducer="USA" value="TDC"/>
        <xs:enumeration ism:classification="U" ism:ownerProducer="USA" value="PAYL"/>
        <xs:enumeration ism:classification="U" ism:ownerProducer="USA" value="EXPLICIT"/>
        <xs:enumeration ism:classification="U" ism:ownerProducer="USA" value="DESC_TDO"/>
        <xs:enumeration ism:classification="U" ism:ownerProducer="USA" value="DESC_PAYL"/>
        <xs:enumeration ism:classification="U" ism:ownerProducer="USA" value="TDC_MEMBER"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="isEncrypted" type="xs:boolean">
    <xs:annotation>
        <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to denote if contents
            are encrypted. When this optional attribute is absent, it is assumed to be
            false.</xhtml:p>
        </xs:documentation>
    </xs:annotation>
</xs:attribute>
<xs:attribute name="includesStatementMetadata" type="xs:boolean">

```



```

    <xs:annotation>
      <xs:documentation>
        <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to indicate whether or
not to include element StatementMetadata when referencing an Assertion. In the
case of signatures and binding, this attribute indicates whether or not the
statement metadata is covered by the signature or binding. If not, it cannot be
cryptographically verified and should be considered informative only.
IncludesStatementMetadata should never be set on SignatureValue if there is a
boundValueList, because the BoundValue elements in the list each have their own
explicit includesStatementMetadata attribute.</xhtml:p>
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="normalizationMethod" type="xs:anyURI">
    <xs:annotation>
      <xs:documentation>
        <xhtml:p ism:classification="U" ism:ownerProducer="USA">A URI that provides guidance
on how to format the included values such as whitespace, attributes, and child
nodes in a universally consistent manner. The normalization method is essential
to prevent formatting such as whitespace and order from interfering with the
validation of the cryptographic integrity of data.</xhtml:p>
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="uri" type="xs:anyURI">
    <xs:annotation>
      <xs:documentation>
        <xhtml:p ism:classification="U" ism:ownerProducer="USA">A uri expressing the
location of the referenced material.</xhtml:p>
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <!-- ***** -->
<!--      Complex Types      -->
<!-- ***** -->
<xs:complexType name="TrustedDataCollectionType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">A list of Trusted Data
Collections.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:group ref="AssertionGroup" maxOccurs="1" minOccurs="1"/>
    <xs:choice maxOccurs="unbounded" minOccurs="1">
      <xs:element ref="TrustedDataCollection"/>
      <xs:element ref="TrustedDataObject" maxOccurs="1"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute ref="version" use="required"/>
  <xs:attribute ref="sf:DESVersion" use="optional"/>
</xs:complexType>
<!--Replacing:
<xs:complexType name="TrustedDataObjectType">
```

```
<xs:annotation>
  <xs:documentation>
    <xhtml:p ism:classification="U" ism:ownerProducer="USA">A list of Trusted Data
      Objects.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:sequence>
  <xs:group ref="AssertionGroup" maxOccurs="1" minOccurs="1"/>
  <xs:group ref="EncryptionInformationGroup"/>
  <xs:group ref="PayloadGroup"/>
</xs:sequence>
<xs:attribute ref="version"/>
<xs:attribute ref="sf:DESVersion" use="optional"/>
<xs:attribute ref="id" use="optional"/>
</xs:complexType>
-->
<!-- DigitalHazMatCom: Replace EncryptionInformationGroup with EncryptionInformationGroupRequired -->
<xs:complexType name="TrustedDataObjectType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">A list of Trusted Data
        Objects.</xhtml:p>
      </xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:group maxOccurs="1" minOccurs="1" ref="AssertionGroup"/>
      <xs:group maxOccurs="1"
        minOccurs="1"
        ref="EncryptionInformationGroupRequired"/>
      <xs:group ref="PayloadGroup"/>
    </xs:sequence>
    <xs:attribute ref="version"/>
    <xs:attribute ref="sf:DESVersion" use="optional"/>
    <xs:attribute ref="id" use="optional"/>
  </xs:complexType>
  <xs:complexType name="EncryptionMethodType">
    <xs:annotation>
      <xs:documentation>
        <xhtml:p ism:classification="U" ism:ownerProducer="USA">Describes the encryption
          method</xhtml:p>
        </xs:documentation>
      </xs:annotation>
      <xs:sequence>
        <xs:element name="KeySize" type="xs:integer" minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              <xhtml:p ism:classification="U" ism:ownerProducer="USA">The size of the key
                used for encryption expressed as an integer.</xhtml:p>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="KeyEncodingFormat" type="xs:string" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
```

```

    <xhtml:p ism:classification="U" ism:ownerProducer="USA">The name of the
    primary encoding format of the key. The primary encoding format is named
    in terms of the appropriate ASN.1 data format, if an ASN.1 specification
    for the key exists. For example, the name of the ASN.1 data format for
    public keys is SubjectPublicKeyInfo, as defined by the X.509 standard;
    in this case, the returned format is "X.509". Similarly, the name of the
    ASN.1 data format for private keys is PrivateKeyInfo, as defined by the
    PKCS #8 standard; in this case, the returned format is "PKCS#8".
  </xhtml:p>
</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element name="IVParams" type="xs:base64Binary" minOccurs="0">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to express the
      Initialization Vector (IV) used by block cipher modes of operation.
    </xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="OaepParams" type="xs:base64Binary" minOccurs="0">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to express the
      Optimal Asymmetric Encryption Padding (OAEP) scheme</xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="HashAlgorithm" type="xs:anyURI" minOccurs="0">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to express the
      Hash function used with the Optimal Asymmetric Encryption Padding (OAEP)
      scheme.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="MGFAlgorithm" type="xs:anyURI" minOccurs="0">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to express the
      Mask Generation Function used with the Optimal Asymmetric Encryption
      Padding (OAEP) scheme.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="Tweak" type="xs:base64Binary" minOccurs="0">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to express the
      Tweak used by various Cipher Block Chaining (CBC) schemes.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:element>

```

```

    </xs:element>
    <xs:element name="Nonce" type="xs:base64Binary" minOccurs="0">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to express the
            Nonce used by various Offset Codebook (OCB) mode schemes.</xhtml:p>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="AdditionalAuthenticatedData"
        type="xs:base64Binary"
        minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to express the
              Additional Authentication Data (AAD) for Galois Counter Mode (GCM) of
              block cipher algorithms.</xhtml:p>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="AuthenticationTag" type="xs:base64Binary" minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              <xhtml:p ism:classification="U" ism:ownerProducer="USA">A cryptographic
                checksum on data that is designed to reveal both accidental errors and
                the intentional modification of the data in Galois Counter Mode (GCM) of
                block cipher algorithms.</xhtml:p>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="algorithm" type="xs:anyURI" use="required">
          <xs:annotation>
            <xs:documentation>
              <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to express the
                encryption algorithm utilized</xhtml:p>
              </xs:documentation>
            </xs:annotation>
          </xs:attribute>
        </xs:complexType>
        <xs:group name="AssertionGroup">
          <xs:annotation>
            <xs:documentation>
              <xhtml:p ism:ownerProducer="USA" ism:classification="U">The group of possible
                Assertion elements in a TDO or TDC.</xhtml:p>
              </xs:documentation>
            </xs:annotation>
          </xs:sequence>
        </xs:sequence>
      <!--Replacing:
      <xs:element name="HandlingAssertion" type="HandlingAssertionType" maxOccurs="unbounded" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">A specific type of
              assertion designed to be used for access, rights, and handling

```

```

        instructions. It is expected that handling instructions should never
        have metadata about themselves and they should never be encrypted.
        Therefore, unlike regular assertions, handling assertions do not support
        statement metadata or encryption.</xhtml:p>
    </xs:documentation>
</xs:annotation>
</xs:element>
-->
<!-- DigitalHazMatCom: Remove HandlingAssertion So we can't do any classified or CUI -->

    <xs:element name="Assertion"
                type="AssertionType"
                maxOccurs="unbounded"
                minOccurs="0">
        <xs:annotation>
            <xs:documentation>
                <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to express
                metadata about the objects expressed in the scope attribute of the
                assertion. An assertion also supports metadata about the assertion
                statement for the purposes of indicating any handling instructions
                pertinent to the statement itself. Also supports encrypted statements
                and binding the statement with objects in its scope.</xhtml:p>
            </xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:sequence>
</xs:group>
<xs:group name="BindingGroup">
    <xs:annotation>
        <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">The group of elements that
            contains information needed to express, understand, and/or cryptographically
            validate the binding of the objects that belong to the scope of the assertion.
        </xhtml:p>
    </xs:documentation>
</xs:annotation>
    <xs:choice>
        <xs:element name="Binding"
                    type="BindingType"
                    minOccurs="1"
                    maxOccurs="unbounded">
            <xs:annotation>
                <xs:documentation>
                    <xhtml:p ism:classification="U" ism:ownerProducer="USA">Contains information
                    needed to express, understand, and/or cryptographically validate the
                    binding of the objects that belong to the scope of the assertion.
                </xhtml:p>
            </xs:documentation>
        </xs:annotation>
    </xs:element>
        <xs:element name="ReferenceList" type="ReferenceListType">
            <xs:annotation>
                <xs:documentation>
                    <xhtml:p ism:classification="U" ism:ownerProducer="USA">Contains information
```

```

        needed to express, understand, and/or validate the informative
        (non-cryptographic) binding of the objects that belong to the scope of
        the assertion.</xhtml:p>
      </xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:choice>
</xs:group>
<!-- DigitalHazMatCom: Add EncryptionInformationGroupRequired group -->
<xs:group name="EncryptionInformationGroupRequired">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">The group of elements used
to express encryption information in an Assertion or a TDO.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="1" name="EncryptionInformation">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Top level element
for holding information related to the encryption of an assertion or
payload. Multiple child KeyAccess and/or EncryptionMethod elements
represent onion or layered encryption. In this case, the first child
represents the outermost layer of encryption.</xhtml:p>
        </xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:choice maxOccurs="1">
          <xs:sequence>
            <xs:element minOccurs="1" name="KeyAccess" type="KeyAccessType">
              <xs:annotation>
                <xs:documentation>
                  <xhtml:p ism:classification="U" ism:ownerProducer="USA">
Contains information pertaining to the key for which the
application value(s) was/were encrypted and/or that is
necessary for decryption.</xhtml:p>
                </xs:documentation>
              </xs:annotation>
            </xs:element>
            <xs:element maxOccurs="1"
              minOccurs="1"
              name="EncryptionMethod"
              type="EncryptionMethodType">
              <xs:annotation>
                <xs:documentation>
                  <xhtml:p ism:classification="U" ism:ownerProducer="USA">
Contains information pertaining to the methods for which
the applicable value(s) was/were encrypted. (i.e.
SHA256)</xhtml:p>
                </xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>

```

```
</xs:choice>
<xs:attribute name="sequenceNum" type="xs:integer" use="optional">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">The sequence number of the encryption info.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:group>
<xs:group name="EncryptionInformationGroup">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">The group of elements used
to express encryption information in an Assertion or a TDO.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element maxOccurs="unbounded" minOccurs="0" name="EncryptionInformation">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Top level element
for holding information related to the encryption of an assertion or
payload. Multiple child KeyAccess and/or EncryptionMethod elements
represent onion or layered encryption. In this case, the first child
represents the outermost layer of encryption.</xhtml:p>
        </xs:documentation>
      </xs:annotation>
    </xs:complexType>
    <xs:choice maxOccurs="1">
      <xs:sequence>
        <xs:element minOccurs="1" name="KeyAccess" type="KeyAccessType">
          <xs:annotation>
            <xs:documentation>
              <xhtml:p ism:classification="U" ism:ownerProducer="USA">
Contains information pertaining to the key for which the
application value(s) was/were encrypted and/or that is
necessary for decryption.</xhtml:p>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element maxOccurs="1"
minOccurs="1"
name="EncryptionMethod"
type="EncryptionMethodType">
          <xs:annotation>
            <xs:documentation>
              <xhtml:p ism:classification="U" ism:ownerProducer="USA">
Contains information pertaining to the methods for which
the applicable value(s) was/were encrypted. (i.e.
SHA256)</xhtml:p>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:choice>
  </xs:sequence>
</xs:group>
</xs:complexType>
</xs:element>
```



```

        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:choice>
  <xs:attribute name="sequenceNum" type="xs:integer" use="optional">
    <xs:annotation>
      <xs:documentation>
        <xhtml:p ism:classification="U" ism:ownerProducer="USA">The sequence number of the encryption info.</xhtml:p>
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:group>
<xs:group name="PayloadGroup">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">The group of elements intended
for textual payload content encoded as a string. Perhaps the contents of a text
file.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
  <xs:choice>
<!--Replacing:
<xs:element name="StringPayload" type="StringValueType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">Intended for textual
value content encoded as a string. Perhaps the contents of a text
file.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:element>
-->
<!-- DigitalHazMatCom: Remove references to StringPayload -->

  <xs:element name="Base64BinaryPayload" type="Base64BinaryValueType">
    <xs:annotation>
      <xs:documentation>
        <xhtml:p ism:classification="U" ism:ownerProducer="USA">Intended for holding
base64binary values such as a file or other binary data.</xhtml:p>
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="ReferenceValuePayload" type="ReferenceValueType">
    <xs:annotation>
      <xs:documentation>
        <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to reference
payloads that are not embedded in the TDO but stored in a
remote/external location.</xhtml:p>
      </xs:documentation>
    </xs:annotation>
  </xs:element>

```



```

        <!--Replacing:
<xs:element name="StructuredPayload" type="StructuredValueType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">Intended for
        structured content encoded in the same data encoding of the
        encapsulating TDO (i.e. If the encoded format is XML this is intended
        for XML statements).</xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:element>
-->
<!-- DigitalHazMatCom: Remove references to StructuredPayload -->
</xs:choice>

</xs:group>
<xs:group name="StatementGroup">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">The group of elements
        intended for contextual statement content encoded as a string.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
  <xs:choice>
    <xs:element name="StringStatement" type="StringValueType">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Intended for textual
            statement content encoded as a string. Perhaps the contents of a text
            file.</xhtml:p>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    <xs:element name="Base64BinaryStatement" type="Base64BinaryValueType">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Intended for holding
            base64binary statement values such as a file or other binary encoded
            data.</xhtml:p>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    <xs:element name="ReferenceStatement" type="ReferenceValueType">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to reference
            statements that are not embedded in the TDO but stored in a
            remote/external location.</xhtml:p>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    <xs:element name="StructuredStatement" type="StructuredValueType">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Intended for

```

```

        structured content encoded in the same data encoding of the
        encapsulating Assertion (i.e. If the encoded format is XML this is
        intended for XML statements).</xhtml:p>
      </xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:choice>
</xs:group>
<xs:complexType name="AssertionType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">An attribute for expressing
the AssertionType of an object Used to express metadata about the objects
expressed in the scope attribute of the assertion.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="StatementMetadata"
      type="StatementMetadataType"
      minOccurs="0"
      maxOccurs="2">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Intended for access,
rights, handling or other metadata that applies to the assertion
statement. Use EDH security options whenever an assertion already has a
unique enterprise identifier or is intended for potential extraction and
should be able stand on it's own as a separate referenceable object. Use
arh security only when assertions are not intended to be extracted and
do not require enterprise identifiers.</xhtml:p>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:group ref="EncryptionInformationGroup"/>
    <xs:group ref="StatementGroup"/>
    <xs:group ref="BindingGroup" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="scope" use="required"/>
  <xs:attribute name="type" type="xs:string">
    <xs:annotation>
      <xs:documentation>
        <xhtml:p ism:ownerProducer="USA" ism:classification="U">The logical grouping to
which the assertion belongs. The Assertion type attribute is intended to
provide additional context, allowing various systems to pre-determine
relevance of assertions without parsing or reading all of the assertions.
Type might include categorizations such as discovery, mission, or task order
to allow various systems to determine which assertions are relevant for them
to parse.</xhtml:p>
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute ref="id" use="optional"/>
</xs:complexType>
<xs:complexType name="StatementMetadataType">

```

```

        <xs:annotation>
          <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to indicate if the
statement metadata applies to encrypted or unencrypted data.</xhtml:p>
          </xs:documentation>
        </xs:annotation>
      </xs:sequence>
      <xs:sequence>
        <xs:any namespace="##other" processContents="skip"/>
      </xs:sequence>
      <xs:attribute name="appliesToState"
        type="tdfstate:CVEEnumTDFAppliesToState"
        use="optional">
        <xs:annotation>
          <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to indicate if the
statement metadata applies to encrypted or unencrypted data. If a TDO
payload or assertion statement is encrypted, there are in fact two
potentially different markings needed for decision making, analysis and
querying, one describing the handling required for the encrypted blob, and
the other for the handling required for the unencrypted (and in effect
external) state. In cases where statements and/or payloads are encrypted,
allow handling assertions and statement metadata elements to indicate
whether their marks apply to the encrypted blob state vs. actual data by
using an attribute appliesToState. </xhtml:p>
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:complexType>
    <!--Replacing:
<xs:complexType name="HandlingAssertionType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">The group of elements intended
for access, rights, and/or handling instructions that apply to the scope of the
assertion. </xhtml:p>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="HandlingStatement" type="HandlingStatementType">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Intended for access,
rights, and/or handling instructions that apply to the scope of the
assertion.</xhtml:p>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:group ref="BindingGroup" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute ref="scope" use="required"/>
  <xs:attribute ref="id" use="optional"/>
  <xs:attribute name="appliesToState" type="tdfstate:CVEEnumTDFAppliesToState" use="optional">
    <xs:annotation>
      <xs:documentation>

```

```

        <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to indicate if the
        statement metadata applies to encrypted or unencrypted data. If a TDO
        payload or assertion statement is encrypted, there are in fact two
        potentially different markings needed for decision making, analysis and
        querying, one describing the handling required for the encrypted blob, and
        the other for the handling required for the unencrypted (and in effect
        external) state. In cases where statements and/or payloads are encrypted,
        allow handling assertions and statement metadata elements to indicate
        whether their marks apply to the encrypted blob state vs. actual data by
        using an attribute appliesToState</xhtml:p>
    </xs:documentation>
</xs:annotation>
</xs:attribute>
</xs:complexType>
-->
<!-- DigitalHazMatCom: Remove HandlingAssertionType so we can't do any classified or CUI -->
<!--Replacing:
<xs:complexType name="HandlingStatementType">
    <xs:annotation>
        <xs:documentation>
            <xhtml:p ism:ownerProducer="USA" ism:classification="U">A list of assertions
            designed to be used for access, rights, and handling instructions.</xhtml:p>
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:any namespace="##other" processContents="skip"/>
    </xs:sequence>
</xs:complexType>
-->
<!-- DigitalHazMatCom: Remove HandlingStatementType so we can't do any classified or CUI -->
<xs:complexType name="Base64BinaryValueType">
    <xs:annotation>
        <xs:documentation>
            <xhtml:p ism:ownerProducer="USA" ism:classification="U">A type for holding
            base64binary values.</xhtml:p>
        </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
<!--Replacing:
<xs:extension base="xs:base64Binary">
    <xs:attribute ref="mediaType" use="optional"/>
    <xs:attribute ref="filename" use="optional"/>
    <xs:attribute ref="isEncrypted" use="optional"/>
    <xs:attribute ref="id" use="optional"/>
</xs:extension>
-->
<!-- DigitalHazMatCom: Replace xs:base64Binary with Base64BinaryRestrictedType -->

    <xs:extension base="Base64BinaryRestrictedType">
        <xs:attribute ref="mediaType" use="optional"/>
        <xs:attribute ref="filename" use="optional"/>
        <xs:attribute ref="isEncrypted" use="optional"/>
        <xs:attribute ref="id" use="optional"/>
    </xs:extension>
```

```

        </xs:simpleContent>
      </xs:complexType>
      <!-- DigitalHazMatCom: Add Base64BinaryRestrictedType type -->
<xs:simpleType name="Base64BinaryRestrictedType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">base64 binary with restricted length</xhtml:p>
    </xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:base64Binary">
    <xs:maxLength value="10000000"/>
  </xs:restriction>
</xs:simpleType>
<xs:complexType name="ReferenceValueType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA"> Incorporates a value by
reference to a URI where it can be found. </xhtml:p>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA"> To support division of a
value into smaller pieces for transport (AKA "chunking"), such as
across a CDS, the body of the element may contain a list of ReferenceValueBlock
elements. If so, each must have a URI to the block and an integer block number
indicating the order in which the blocks can be re-assembled into the original
payload. Block numbers must start at 1 and be sequential. When a list of
ReferenceValueBlocks is used, a TotalHash element must be present and must have
a totalBlocks attribute set to an integer indicating the number of such
elements. </xhtml:p>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">
        <xhtml:strong>Tailoring:</xhtml:strong> Not all systems will be willing or able
to support unbounded lists of blocks. When tailoring maxOccurs here to reflect
limitations imposed by a CDS or other implementation, that change should also be
reflected in the definition of a BlockedHashGroup.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence minOccurs="0" maxOccurs="1">
    <xs:sequence minOccurs="0" maxOccurs="1">
      <xs:element name="ReferenceValueBlock"
        type="ReferenceValueBlockType"
        minOccurs="2"
        maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation xml:lang="en">
            <xhtml:p ism:ownerProducer="USA" ism:classification="U">
              A smaller piece/block of the ReferenceValue that can be used to support transport
              (AKA "chunking") across a CDS.
            </xhtml:p>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:element ref="sfhashv:ContentEncodedHashVerification"
      minOccurs="0"
      maxOccurs="1"/>
    <xs:element ref="sfhashv:ContentDecodedHashVerification"

```

```

        minOccurs="0"
        maxOccurs="1"/>
</xs:sequence>
<xs:attribute ref="uri" use="required"/>
<xs:attribute ref="id" use="optional"/>
<xs:attribute ref="mediaType" use="optional"/>
<xs:attribute ref="isEncrypted" use="optional"/>
<xs:attribute ref="sfhashv:totalBlocks" use="optional"/>
</xs:complexType>
<xs:complexType name="ReferenceValueBlockType">
  <xs:annotation>
    <xs:documentation xml:lang="en">
      <xhtml:p ism:ownerProducer="USA" ism:classification="U">
        A smaller piece/block of the ReferenceValue that can be used to support transport
        (AKA "chunking") across a CDS.
      </xhtml:p>
    </xs:documentation>
  </xs:annotation>
  <xs:attribute ref="uri" use="required"/>
  <xs:attribute ref="sfhashv:block" use="required"/>
</xs:complexType>
<xs:complexType name="StringValueType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">Intended for textual content
      encoded as a string.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute ref="filename" use="optional"/>
      <xs:attribute ref="isEncrypted" use="optional"/>
      <xs:attribute ref="id" use="optional"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:complexType name="StructuredValueType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">Intended for structured
      content encoded in the same data encoding of the encapsulating TDO (i.e. If the
      encoded format is XML this is intended for XML values). For signable
      StructuredValueType elements, it can be safer to declare namespaces locally to
      the section being signed to reduce risk in moving sections between documents.
      Explicit namespace declarations should be used and c14n11 normalization should
      be preferred when signing since c14n11 normalization does not perform any
      namespace re-writing and as a result, signed assertions can not be copied
      between documents unless the namespaces used are identical, or the assertion
      locally overrides them. Older c14n 1.0 has two approaches to namespace
      re-writing, either of which could in some circumstances break signatures when
      copying signed assertions between documents. </xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:sequence>

```

```
<!--Replacing:
<xs:any namespace="##other" processContents="skip"/>
-->
<!-- DigitalHazMatCom: Replace StructuredValueType xs:any with xs:choice of dhzm:DigitalHazMatAssertion,
dhzm:ProvenanceAssertion or anlys:AnalysisAssertion -->

    <xs:choice>
        <xs:element ref="dhzm:DigitalHazMatAssertion"/>
        <xs:element ref="dhzm:ProvenanceAssertion"/>
        <xs:element ref="anlys:AnalysisAssertion"/>
    </xs:choice>
</xs:sequence>
<xs:attribute ref="id" use="optional"/>
<xs:attribute ref="filename" use="optional"/>
<xs:attribute ref="isEncrypted" use="optional"/>
</xs:complexType>
<xs:complexType name="BindingType">
    <xs:annotation>
        <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">A list of the objects
that belong to the scope of the assertion.</xhtml:p>
        </xs:documentation>
    </xs:annotation>
</xs:sequence>
<!-- This order is important because it allows for a single pass
verification of the actual SignatureValue using a streaming parser -->

<xs:choice>
    <xs:element name="Signer" maxOccurs="1" minOccurs="1">
        <xs:annotation>
            <xs:documentation>
                <xhtml:p ism:classification="U" ism:ownerProducer="USA">Information
pertaining to the person or entity that performed the
signing/binding and their credentials.</xhtml:p>
            </xs:documentation>
        </xs:annotation>
        <xs:complexType>
            <xs:attribute name="subject" type="xs:string">
                <xs:annotation>
                    <xs:documentation>
                        <xhtml:p ism:classification="U" ism:ownerProducer="USA">The
distinguished name of the person or entity who is doing the
signing. Refer to RFC 5280 for more information.</xhtml:p>
                    </xs:documentation>
                </xs:annotation>
            </xs:attribute>
            <xs:attribute name="issuer" type="xs:string">
                <xs:annotation>
                    <xs:documentation>
                        <xhtml:p ism:classification="U" ism:ownerProducer="USA">The
distinguished name of the authority that issued the
credentials to the subject. Refer to RFC 5280 for more
information.</xhtml:p>
                    </xs:documentation>
                </xs:annotation>
            </xs:attribute>
        </xs:complexType>
    </xs:element>
    <xs:element name="SignatureValue" type="xs:string">
        <xs:annotation>
            <xs:documentation>
                <xhtml:p ism:classification="U" ism:ownerProducer="USA">The
signature value for the binding.</xhtml:p>
            </xs:documentation>
        </xs:annotation>
    </xs:element>
</xs:choice>
```



```

        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="serial" type="xs:string">
        <xs:annotation>
          <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">The
              unique serial number of the credentials given to the subject
              by the issuer. Refer to RFC 5280 for more information.
            </xhtml:p>
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
</xs:choice>
<xs:element name="SignatureValue"
  type="SignatureValueType"
  minOccurs="1"
  maxOccurs="1">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">Stores the value of the
        Signature over the bound entities.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:element name="BoundValueList"
  type="BoundValueListType"
  minOccurs="0"
  maxOccurs="1">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">BoundValueList is a
        container of bound value references that point to the elements that are included
        in a cryptographic binding. The intent of the BoundValueList is to allow
        granular control over the scope of the binding signature. In the future, when
        BoundValueList is present, the SignatureValue will be calculated over the
        normalized value of the BoundValueList using the normalization method denoted in
        the Binding/SignatureValue/@normalizationMethod attribute.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
</xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="BoundValueListType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">BoundValueList is a
        container of bound value references that point to the elements that are included
        in a cryptographic binding. The intent of the BoundValueList is to allow
        granular control over the scope of the binding signature. In the future, when
        BoundValueList is present, the SignatureValue will be calculated over the
        normalized value of the BoundValueList using the normalization method denoted in
        the Binding/SignatureValue/@normalizationMethod attribute.</xhtml:p>
    </xs:documentation>
  </xs:annotation>

```



```

    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="BoundValue"
      type="BoundValueType"
      minOccurs="1"
      maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">A bound value is a reference
            that points to an element that is included in a cryptographic binding. A bound
            value is only meaningful in the context of a BoundValueList.</xhtml:p>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="BoundValueType">
    <xs:annotation>
      <xs:documentation>
        <xhtml:p ism:classification="U" ism:ownerProducer="USA">A bound value is a reference
          that points to an element that is included in a cryptographic binding. A bound
          value is only meaningful in the context of a BoundValueList.</xhtml:p>
        </xs:documentation>
      </xs:annotation>
    <xs:simpleContent>
      <xs:extension base="xs:base64Binary">
        <xs:attribute ref="idRef" use="required"/>
        <xs:attribute name="hashAlgorithm"
          type="tdfhashal:CVEEnumTDFHashAlgorithm"
          use="required">
          <xs:annotation>
            <xs:documentation>
              <xhtml:p ism:classification="U" ism:ownerProducer="USA">The hash algorithm used in the cryptographic binding.
            </xhtml:p>
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
        <xs:attribute ref="normalizationMethod" use="required"/>
        <xs:attribute ref="includesStatementMetadata" use="optional"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="SignatureValueType">
    <xs:annotation>
      <xs:documentation>
        <xhtml:p ism:classification="U" ism:ownerProducer="USA">Stores the value of the
          Signature over the bound entities.</xhtml:p>
        </xs:documentation>
      </xs:annotation>
    <xs:simpleContent>
      <xs:extension base="xs:base64Binary">
        <xs:attribute name="signatureAlgorithm"
          type="tdfsigal:CVEEnumTDFSignatureAlgorithm"

```

```

        use="required">
<xs:annotation>
  <xs:documentation>
    <xhtml:p ism:classification="U" ism:ownerProducer="USA">The algorithm or
    pattern used by the signature. The permissible values are defined in
    the Controlled Value Enumeration: CVEnumTDFSsignatureAlgorithm.xml
  </xhtml:p>
  </xs:documentation>
</xs:annotation>
</xs:attribute>
<xs:attribute ref="normalizationMethod" use="required"/>
<xs:attribute ref="includesStatementMetadata" use="optional"/>
</xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="KeyAccessType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">Stores the value of the
      key stored in remote locations.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
  <xs:choice maxOccurs="unbounded">
    <xs:element name="RemoteStoredKey"
      type="RemoteStoredKeyType"
      minOccurs="1"
      maxOccurs="1">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Stores retrieval
          information for keys stored in remote locations.</xhtml:p>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="WrappedKey"
      type="WrappedKeyType"
      minOccurs="1"
      maxOccurs="1">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Contains the key
          necessary for decryption in an encrypted state with information
          pertaining to the method in which the key was encrypted.</xhtml:p>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="PasswordKey"
      type="PasswordKeyType"
      minOccurs="1"
      maxOccurs="1">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Used to indicated
          that the key is based on a password.</xhtml:p>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:choice>
</xs:complexType>

```

```

        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="PreSharedKey"
      type="PreSharedKeyType"
      minOccurs="1"
      maxOccurs="1">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Stores the alias
            that references a key that has been previously shared.</xhtml:p>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="AttachedKey"
        type="AttachedKeyType"
        minOccurs="1"
        maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">Contains the key
              necessary for decryption.</xhtml:p>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element maxOccurs="1"
          minOccurs="1"
          name="WrappedPDPKey"
          type="WrappedPDPKeyType">
          <xs:annotation>
            <xs:documentation>
              <xhtml:p ism:classification="U" ism:ownerProducer="USA">Contains the key
                necessary for decryption in an encrypted state with information
                pertaining to the method in which the key was encrypted.</xhtml:p>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    <xs:complexType name="ReferenceListType">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">A list of reference
            Assertions.</xhtml:p>
          </xs:documentation>
        </xs:annotation>
      <xs:sequence>
        <xs:element name="Reference"
          type="ReferenceType"
          minOccurs="1"
          maxOccurs="unbounded">
          <xs:annotation>
            <xs:documentation>Used when referencing an Assertion.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:sequence>
</xs:element>
</xs:complexType>
</xs:schema>
```

```

        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ReferenceType">
    <xs:annotation>
        <xs:documentation>Used when referencing an Assertion.</xs:documentation>
    </xs:annotation>
    <xs:attribute ref="idRef" use="required"/>
    <xs:attribute ref="includesStatementMetadata" use="optional"/>
</xs:complexType>
<!-- Simple Types -->
<xs:simpleType name="MediaTypeType">
    <xs:annotation>
        <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">An attribute for expressing the
mediaType of an object as defined in RFC 4288.</xhtml:p>
        </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
        <xs:annotation>
            <xs:documentation>A restriction on string for the format of mediaType (i.e.
audio/GSM) as defined in <xhtml:a href="http://tools.ietf.org/html/rfc4288">RFC
4288</xhtml:a>. </xs:documentation>
        </xs:annotation>
        <xs:pattern value="[a-zA-Z]*/[a-zA-Z+-.]*"/>
    </xs:restriction>
</xs:simpleType>
<!-- - - - - - Key Access Type Definitions - - - - - -->
<xs:complexType name="AttachedKeyType">
    <xs:annotation>
        <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">Stores the value of the
key necessary for decryption.</xhtml:p>
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element name="KeyValue"
            type="xs:base64Binary"
            minOccurs="1"
            maxOccurs="1">
            <xs:annotation>
                <xs:documentation>The value of the key.</xs:documentation>
            </xs:annotation>
        </xs:element>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PreSharedKeyType">
    <xs:annotation>
        <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">A list of aliases
for referencing a key that has been previously shared.</xhtml:p>
        </xs:documentation>
    </xs:annotation>
    <xs:attribute name="alias" type="xs:string" use="required">
```

```

        <xs:annotation>
          <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">An alias for the shared key.</xhtml:p>
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="store" type="xs:anyURI" use="optional">
        <xs:annotation>
          <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">The store for the shared key.</xhtml:p>
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:complexType>
    <xs:complexType name="RemoteStoredKeyType">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">A list for retrieval
information for keys stored in remote locations.</xhtml:p>
        </xs:documentation>
      </xs:annotation>
      <xs:attribute name="protocol" type="xs:string" use="required">
        <xs:annotation>
          <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">The protocol related to retrieving remote keys.</xhtml:p>
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute ref="uri" use="required"/>
    </xs:complexType>
    <xs:complexType name="PasswordKeyType">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Stores the value of
the key is based on a password.</xhtml:p>
        </xs:documentation>
      </xs:annotation>
      <xs:attribute name="algorithm" type="xs:string" use="required">
        <xs:annotation>
          <xs:documentation>
            <xhtml:p ism:classification="U" ism:ownerProducer="USA">The algorithm related to the password key.</xhtml:p>
          </xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:complexType>
    <xs:complexType name="WrappedKeyType">
      <xs:annotation>
        <xs:documentation>
          <xhtml:p ism:classification="U" ism:ownerProducer="USA">Stores the value of
the key necessary for decryption in an encrypted state with information
pertaining to the method in which the key was encrypted.</xhtml:p>
        </xs:documentation>
      </xs:annotation>
    </xs:sequence>

```

```

    <xs:element name="KeyValue"
      type="xs:base64Binary"
      minOccurs="1"
      maxOccurs="1">
      <xs:annotation>
        <xs:documentation>The value of the decryption key.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:group ref="EncryptionInformationGroup" maxOccurs="1" minOccurs="1"/>
  </xs:sequence>
  <xs:attribute name="keyIdentifier" type="xs:string" use="optional">
    <xs:annotation>
      <xs:documentation>
        <xhtml:p ism:classification="U" ism:ownerProducer="USA">A key identifier.</xhtml:p>
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
<xs:complexType name="WrappedPDPKeyType">
  <xs:annotation>
    <xs:documentation>
      <xhtml:p ism:classification="U" ism:ownerProducer="USA">Stores the value of
the key necessary for decryption in an encrypted state with information
pertaining to the method in which the key was encrypted.</xhtml:p>
    </xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="EncryptedPolicyObject"
      type="xs:base64Binary"
      minOccurs="1"
      maxOccurs="1">
      <xs:annotation>
        <xs:documentation>The encrypted policy object.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:group ref="EncryptionInformationGroup" maxOccurs="1" minOccurs="1"/>
  </xs:sequence>
  <xs:attribute name="keyIdentifier" type="xs:string" use="optional">
    <xs:annotation>
      <xs:documentation>
        <xhtml:p ism:classification="U" ism:ownerProducer="USA">A key identifier.</xhtml:p>
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
<xs:annotation>
  <xs:documentation>
    <xhtml:h2 ism:ownerProducer="USA" ism:classification="U">Formal Change List</xhtml:h2>
    <xhtml:table ism:ownerProducer="USA" ism:classification="U" id="ChangeHistory">
      <xhtml:caption>Change History</xhtml:caption>
      <xhtml:thead>
        <xhtml:tr>
          <xhtml:th>Version</xhtml:th>
          <xhtml:th>Date</xhtml:th>
        </xhtml:tr>
      </xhtml:thead>
    </xhtml:table>
  </xs:documentation>
</xs:annotation>
```

```

        <xhtml:th>By</xhtml:th>
        <xhtml:th>Description</xhtml:th>
      </xhtml:tr>
    </xhtml:thead>
    <xhtml:tbody>
      <xhtml:tr>
        <xhtml:td>2021-JAN</xhtml:td>
        <xhtml:td>2020-11-17</xhtml:td>
        <xhtml:td>ODNI/OCIO/ICEA</xhtml:td>
        <xhtml:td>
          <xhtml:ul>
            <xhtml:li ism:ownerProducer="USA" ism:classification="U">
              Reference the change history in the DES.</xhtml:li>
            </xhtml:li>
          </xhtml:ul>
        </xhtml:td>
      </xhtml:tr>
    </xhtml:tbody>
  </xhtml:table>
</xs:documentation>
</xs:annotation>
</xs:schema>
```

2.2 - DHZMC-TDF-guard.xsd

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="urn:us:gov:ic:tdf"
    xmlns:dhzm="urn:us:gov:ic:digitalhazmat"
    xmlns:sfhashv="urn:us:gov:ic:sf:hashverification"
    xmlns:sf="urn:us:gov:ic:sf"
    xmlns:anlys="urn:us:gov:ic:anlysassert"
    xmlns:tdfsigal="urn:us:gov:ic:cenum:tdf:signaturealgorithm"
    xmlns:tdfstate="urn:us:gov:ic:cenum:tdf:state"
    xmlns:tdfhashal="urn:us:gov:ic:cenum:tdf:hashalgorithm"
    targetNamespace="urn:us:gov:ic:tdf"
    elementFormDefault="qualified"
    attributeFormDefault="qualified"
    version="202111">
  <xs:import namespace="urn:us:gov:ic:cenum:tdf:signaturealgorithm"
    schemaLocation="../../BASE-TDF/CVEGenerated/CVEnumTDFSignatureAlgorithm.xsd"/>
  <xs:import namespace="urn:us:gov:ic:cenum:tdf:hashalgorithm"
    schemaLocation="../../BASE-TDF/CVEGenerated/CVEnumTDFHashAlgorithm.xsd"/>
  <xs:import namespace="urn:us:gov:ic:cenum:tdf:state"
    schemaLocation="../../BASE-TDF/CVEGenerated/CVEnumTDFAppliesToState.xsd"/>
  <xs:import namespace="urn:us:gov:ic:sf:hashverification"
    schemaLocation="../../IC-SF/HashVerification.xsd"/>
  <xs:import namespace="urn:us:gov:ic:sf" schemaLocation="../../IC-SF/IC-SF.xsd"/>
  <xs:import namespace="urn:us:gov:ic:digitalhazmat"
    schemaLocation="../../DHZM/DHZM.xsd"/>
  <xs:import namespace="urn:us:gov:ic:anlysassert"
    schemaLocation="../../ANLYS/ANLYS.xsd"/>
  <xs:element name="TrustedDataCollection" type="TrustedDataCollectionType"/>
  <xs:element name="TrustedDataObject" type="TrustedDataObjectType"/>
  <xs:attribute name="version">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:pattern value="[0-9]{6}(\.[0-9]{6})?\-DHZMC\-TDF\[0-9]{6}(\.[0-9]{6})?(\-\.{1,23})?" />
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="mediaType" type="MediaTypeType"/>
  <xs:attribute name="id" type="xs:ID"/>
  <xs:attribute name="idRef" type="xs:IDREF"/>
  <xs:attribute name="filename" type="xs:string"/>
  <xs:attribute name="scope">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="TDO"/>
        <xs:enumeration value="TDC"/>
        <xs:enumeration value="PAYL"/>
        <xs:enumeration value="EXPLICIT"/>
        <xs:enumeration value="DESC_TDO"/>
        <xs:enumeration value="DESC_PAYL"/>
        <xs:enumeration value="TDC_MEMBER"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
```



```

<xs:attribute name="isEncrypted" type="xs:boolean"/>
<xs:attribute name="includesStatementMetadata" type="xs:boolean"/>
<xs:attribute name="normalizationMethod" type="xs:anyURI"/>
<xs:attribute name="uri" type="xs:anyURI"/>
<xs:complexType name="TrustedDataCollectionType">
  <xs:sequence>
    <xs:group ref="AssertionGroup"/>
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="TrustedDataCollection"/>
      <xs:element ref="TrustedDataObject"/>
    </xs:choice>
  </xs:sequence>
  <xs:attribute ref="version" use="required"/>
  <xs:attribute ref="sf:DESVersion" use="optional"/>
</xs:complexType>
<xs:complexType name="TrustedDataObjectType">
  <xs:sequence>
    <xs:group ref="AssertionGroup"/>
    <xs:group ref="EncryptionInformationGroupRequired"/>
    <xs:group ref="PayloadGroup"/>
  </xs:sequence>
  <xs:attribute ref="version"/>
  <xs:attribute ref="sf:DESVersion" use="optional"/>
  <xs:attribute ref="id" use="optional"/>
</xs:complexType>
<xs:complexType name="EncryptionMethodType">
  <xs:sequence>
    <xs:element name="KeySize" type="xs:integer" minOccurs="0"/>
    <xs:element name="KeyEncodingFormat" type="xs:string" minOccurs="0"/>
    <xs:element name="IVParams" type="xs:base64Binary" minOccurs="0"/>
    <xs:element name="OaepParams" type="xs:base64Binary" minOccurs="0"/>
    <xs:element name="HashAlgorithm" type="xs:anyURI" minOccurs="0"/>
    <xs:element name="MGFAlgorithm" type="xs:anyURI" minOccurs="0"/>
    <xs:element name="Tweak" type="xs:base64Binary" minOccurs="0"/>
    <xs:element name="Nonce" type="xs:base64Binary" minOccurs="0"/>
    <xs:element name="AdditionalAuthenticatedData"
      type="xs:base64Binary"
      minOccurs="0"/>
    <xs:element name="AuthenticationTag" type="xs:base64Binary" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="algorithm" type="xs:anyURI" use="required"/>
</xs:complexType>
<xs:group name="AssertionGroup">
  <xs:sequence>
    <xs:element
      name="Assertion"
      type="AssertionType"
      maxOccurs="unbounded"
      minOccurs="0"/>
  </xs:sequence>
</xs:group>
<xs:group name="BindingGroup">
  <xs:choice>
    <xs:element name="Binding" type="BindingType" maxOccurs="unbounded"/>
    <xs:element name="ReferenceList" type="ReferenceListType"/>
  </xs:choice>
</xs:group>

```

```

    </xs:choice>
  </xs:group>
  <xs:group name="EncryptionInformationGroupRequired">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="EncryptionInformation">
        <xs:complexType>
          <xs:choice>
            <xs:sequence>
              <xs:element name="KeyAccess" type="KeyAccessType"/>
              <xs:element name="EncryptionMethod" type="EncryptionMethodType"/>
            </xs:sequence>
          </xs:choice>
          <xs:attribute name="sequenceNum" type="xs:integer" use="optional"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:group>
  <xs:group name="EncryptionInformationGroup">
    <xs:sequence>
      <xs:element maxOccurs="unbounded" minOccurs="0" name="EncryptionInformation">
        <xs:complexType>
          <xs:choice>
            <xs:sequence>
              <xs:element name="KeyAccess" type="KeyAccessType"/>
              <xs:element name="EncryptionMethod" type="EncryptionMethodType"/>
            </xs:sequence>
          </xs:choice>
          <xs:attribute name="sequenceNum" type="xs:integer" use="optional"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:group>
  <xs:group name="PayloadGroup">
    <xs:choice>
      <xs:element name="Base64BinaryPayload" type="Base64BinaryValueType"/>
      <xs:element name="ReferenceValuePayload" type="ReferenceValueType"/>
    </xs:choice>
  </xs:group>
  <xs:group name="StatementGroup">
    <xs:choice>
      <xs:element name="StringStatement" type="StringValueType"/>
      <xs:element name="Base64BinaryStatement" type="Base64BinaryValueType"/>
      <xs:element name="ReferenceStatement" type="ReferenceValueType"/>
      <xs:element name="StructuredStatement" type="StructuredValueType"/>
    </xs:choice>
  </xs:group>
  <xs:complexType name="AssertionType">
    <xs:sequence>
      <xs:element name="StatementMetadata"
        type="StatementMetadataType"
        minOccurs="0"
        maxOccurs="2"/>
      <xs:group ref="EncryptionInformationGroup"/>
      <xs:group ref="StatementGroup"/>
    </xs:sequence>
  </xs:complexType>

```

```

        <xs:group ref="BindingGroup" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute ref="scope" use="required"/>
    <xs:attribute name="type" type="xs:string"/>
    <xs:attribute ref="id" use="optional"/>
</xs:complexType>
<xs:complexType name="StatementMetadataType">
    <xs:sequence>
        <xs:any namespace="##other" processContents="skip"/>
    </xs:sequence>
    <xs:attribute name="appliesToState"
        type="tdfstate:CVEEnumTDFAppliesToState"
        use="optional"/>
</xs:complexType>
<xs:complexType name="Base64BinaryValueType">
    <xs:simpleContent>
        <xs:extension base="Base64BinaryRestrictedType">
            <xs:attribute ref="mediaType" use="optional"/>
            <xs:attribute ref="filename" use="optional"/>
            <xs:attribute ref="isEncrypted" use="optional"/>
            <xs:attribute ref="id" use="optional"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:simpleType name="Base64BinaryRestrictedType">
    <xs:restriction base="xs:base64Binary">
        <xs:maxLength value="10000000"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="ReferenceValueType">
    <xs:sequence minOccurs="0">
        <xs:sequence minOccurs="0">
            <xs:element name="ReferenceValueBlock"
                type="ReferenceValueBlockType"
                minOccurs="2"
                maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:element ref="sfhashv:ContentEncodedHashVerification" minOccurs="0"/>
        <xs:element ref="sfhashv:ContentDecodedHashVerification" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute ref="uri" use="required"/>
    <xs:attribute ref="id" use="optional"/>
    <xs:attribute ref="mediaType" use="optional"/>
    <xs:attribute ref="isEncrypted" use="optional"/>
    <xs:attribute ref="sfhashv:totalBlocks" use="optional"/>
</xs:complexType>
<xs:complexType name="ReferenceValueBlockType">
    <xs:attribute ref="uri" use="required"/>
    <xs:attribute ref="sfhashv:block" use="required"/>
</xs:complexType>
<xs:complexType name="StringValueType">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute ref="filename" use="optional"/>
        </xs:extension>
    </xs:simpleContent>

```

```

        <xs:attribute ref="isEncrypted" use="optional"/>
        <xs:attribute ref="id" use="optional"/>
    </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="StructuredValueType">
    <xs:sequence>
        <xs:choice>
            <xs:element ref="dhzm:DigitalHazMatAssertion"/>
            <xs:element ref="dhzm:ProvenanceAssertion"/>
            <xs:element ref="anlys:AnalysisAssertion"/>
        </xs:choice>
    </xs:sequence>
    <xs:attribute ref="id" use="optional"/>
    <xs:attribute ref="filename" use="optional"/>
    <xs:attribute ref="isEncrypted" use="optional"/>
</xs:complexType>
<xs:complexType name="BindingType">
    <xs:sequence>
        <xs:choice>
            <xs:element name="Signer">
                <xs:complexType>
                    <xs:attribute name="subject" type="xs:string"/>
                    <xs:attribute name="issuer" type="xs:string"/>
                    <xs:attribute name="serial" type="xs:string"/>
                </xs:complexType>
            </xs:element>
        </xs:choice>
        <xs:element name="SignatureValue" type="SignatureValueType"/>
        <xs:element name="BoundValueList" type="BoundValueListType" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="BoundValueListType">
    <xs:sequence>
        <xs:element name="BoundValue" type="BoundValueType" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="BoundValueType">
    <xs:simpleContent>
        <xs:extension base="xs:base64Binary">
            <xs:attribute ref="idRef" use="required"/>
            <xs:attribute name="hashAlgorithm"
                type="tdfhashal:CVEEnumTDFHashAlgorithm"
                use="required"/>
            <xs:attribute ref="normalizationMethod" use="required"/>
            <xs:attribute ref="includesStatementMetadata" use="optional"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
<xs:complexType name="SignatureValueType">
    <xs:simpleContent>
        <xs:extension base="xs:base64Binary">
            <xs:attribute name="signatureAlgorithm"
                type="tdfsigal:CVEEnumTDFSignatureAlgorithm"

```

```

        use="required"/>
        <xs:attribute ref="normalizationMethod" use="required"/>
        <xs:attribute ref="includesStatementMetadata" use="optional"/>
    </xs:extension>
</xs:simpleContent>
</xs:complexType>
<xs:complexType name="KeyAccessType">
    <xs:choice maxOccurs="unbounded">
        <xs:element name="RemoteStoredKey" type="RemoteStoredKeyType"/>
        <xs:element name="WrappedKey" type="WrappedKeyType"/>
        <xs:element name="PasswordKey" type="PasswordKeyType"/>
        <xs:element name="PreSharedKey" type="PreSharedKeyType"/>
        <xs:element name="AttachedKey" type="AttachedKeyType"/>
        <xs:element name="WrappedPDPKey" type="WrappedPDPKeyType"/>
    </xs:choice>
</xs:complexType>
<xs:complexType name="ReferenceListType">
    <xs:sequence>
        <xs:element name="Reference" type="ReferenceType" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ReferenceType">
    <xs:attribute ref="idRef" use="required"/>
    <xs:attribute ref="includesStatementMetadata" use="optional"/>
</xs:complexType>
<xs:simpleType name="MediaTypeType">
    <xs:restriction base="xs:string">
        <xs:pattern value="[a-zA-Z]*/[a-zA-Z+-.]*"/>
    </xs:restriction>
</xs:simpleType>
<xs:complexType name="AttachedKeyType">
    <xs:sequence>
        <xs:element name="KeyValue" type="xs:base64Binary"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="PreSharedKeyType">
    <xs:attribute name="alias" type="xs:string" use="required"/>
    <xs:attribute name="store" type="xs:anyURI" use="optional"/>
</xs:complexType>
<xs:complexType name="RemoteStoredKeyType">
    <xs:attribute name="protocol" type="xs:string" use="required"/>
    <xs:attribute ref="uri" use="required"/>
</xs:complexType>
<xs:complexType name="PasswordKeyType">
    <xs:attribute name="algorithm" type="xs:string" use="required"/>
</xs:complexType>
<xs:complexType name="WrappedKeyType">
    <xs:sequence>
        <xs:element name="KeyValue" type="xs:base64Binary"/>
        <xs:group ref="EncryptionInformationGroup"/>
    </xs:sequence>
    <xs:attribute name="keyIdentifier" type="xs:string" use="optional"/>
</xs:complexType>
<xs:complexType name="WrappedPDPKeyType">

```

```

    <xs:sequence>
      <xs:element name="EncryptedPolicyObject" type="xs:base64Binary"/>
      <xs:group ref="EncryptionInformationGroup"/>
    </xs:sequence>
    <xs:attribute name="keyIdentifier" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:schema>
```