Tokens and Protocol for the Temporal Integrity Markup Language (TIML)

1 Introduction

This submission defines an XML schema for a timestamping protocol. The schema is based upon the RFC 3161 ASN.1 timestamping protocol, but uses the XML Signature standard for signature formatting.

2 Terminology

The key words *must*, *must not*, *required*, *shall*, *shall not*, *should*, *should not*, *recommended*, *may*, and *optional* in this document are to be interpreted as described in **[RFC2119]**.

3 TIML Protocol Elements

References to the ASN.1 protocol fields in this description refer to fields defined in **[RFC3161].**

3.1 Schema Header and Namespace Declarations

The following schema fragment defines the XML namespaces and other header information for the timestamping schema:

```
<?xml version="1.0"
      encoding="UTF-8" ?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        targetNamespace="http://www.entrust.com/schemas/timestamp-
protocol-20020207"
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
       xmlns:ts="http://www.entrust.com/schemas/timestamp-protocol-
20020207"
       xmlns:xs="http://www.w3.org/2001/XMLSchema"
        elementFormDefault="qualified"
        attributeFormDefault="unqualified"
       version="1.0"
       xml:lang="en">
  <import namespace="http://www.w3.org/2000/09/xmldsig#"
          schemaLocation="http://www.w3.org/TR/2001/PR-xmldsig-core-
20010820/xmldsig-core-schema.xsd" />
```

Note: The targetNamespace will have to be changed to one defined by OASIS, not one defined by Entrust.

3.2 Elements <StatusText> and <StatusInfo>

The <StatusText> element MAY be used to include reason text such as "Digest element is not correctly formatted".

The following schema fragment defines the <StatusText> element:

```
  <element name="StatusText" type="string"/>
```

The <StatusInfo> element specifies the status of the response to a timestamp request. It includes the following elements and attributes:

<StatusInfo> [Optional]

MAY be used to include reason text such as "Digest element is not correctly formatted".

status [Required]

Indicates the status of the response. The values correspond to the values in the ASN.1 protocol field PKIStatus, with the same semantics and restrictions.

failureInfo [Optional]

When the request was not fulfilled, the failureInfo attributee indicates the reason why the time-stamp request was rejected. The values correspond to the values in the ASN.1 protocol field PKIFailureInfo, with the same semantics and restrictions.

The following schema fragment defines the <StatusInfo> element:

```
  <element name="StatusInfo">
    <complexType>
      <sequence>
        <element ref="ts:StatusText" minOccurs="0" />
      </sequence>
      <attribute name="status" use="required">
        <simpleType>
          <restriction base="NMTOKEN">
            <enumeration value="granted" />
            <enumeration value="grantedWithModifications" />
            <enumeration value="rejection" />
            <enumeration value="waiting" />
            <enumeration value="revocationWarning" />
            <enumeration value="revocationNotification" />
          </restriction>
        </simpleType>
      </attribute>
      <attribute name="failureInfo" use="optional">
        <simpleType>
          <restriction base="NMTOKEN">
            <enumeration value="badAlgorithm" />
            <enumeration value="badRequest" />
            <enumeration value="badDataFormat" />
            <enumeration value="timeNotAvailable" />
            <enumeration value="unacceptedPolicy" />
            <enumeration value="unacceptedExtension" />
            <enumeration value="additionalInformationNotAvailable"
/>
            <enumeration value="systemFailure" />
          </restriction>
        </simpleType>
      </attribute>
    </complexType>
```

```
  </element>
```

3.3 Element <Policy>

The <Policy> element specifies the policy under which the timestamp was created. It contains the following attribute:

id [Required]

Identifies the policy, by means of a URI.

The following schema fragment defines the <Policy> element:

```
<element name="Policy">
    <complexType>
      <attribute name="id" type="anyURI" use="required" />
    </complexType>
  </element>
```

3.4 Element <Digest>

The <Digest> element contains the message digest of the document that is being timestamped. It contains the following elements:

<Transforms> [Optional]

Indicates the transforms (e.g., canonicalization algorithms) that must be applied to the source before being digested. This element is imported from **[XMLSig]**.

<DigestMethod> [Required]

Indicates the digest method used to create the message digest. This element is imported from **[XMLSig]**.

<DigestValue> [Required]

Contains the output of the digest method algorithm applied to the transformed document being timestamped. This element is imported from **[XMLSig]**.

The following schema fragment defines the <Digest> element:

```
  <element name="Digest">
    <complexType>
      <sequence>
        <element ref="ds:Transforms" minOccurs="0" />
        <element ref="ds:DigestMethod" />
        <element ref="ds:DigestValue" />
      </sequence>
    </complexType>
  </element>
```

3.5 Element <SerialNumber>

The <SerialNumber> element contains an integer serial number of the timestamp that is issued.

The following schema fragment defines the <SerialNumber> element.

```
<element name="SerialNumber" type="integer" />
```

3.6 Element <CreationTime>

The <CreationTime> element contains the time at which the timestamp token has been created by the TSA.

The following schema fragment defines the <CreationTime> element.

```
<element name="CreationTime" type="dateTime" />
```

3.7 Element <Accuracy>

The <Accuracy> element represents the time deviation around the time contained in <CreationTime>. By adding the <accuracy> value to the <CreationTime> value, an upper limit on the time at which the timestamp token has been created by the TSA can be obtained. In the same way, by subtracting the <accuracy> value from the <CreationTime> value, a lower limit on the time at which the timestamp token has been created by the TSA can be obtained. It contains the following attributes:

`seconds` [Optional]

A non-negative integer representing the accuracy in seconds.

`milliseconds` [Optional]

An integer in the range [1..999] representing the accuracy in milliseconds.

`microseconds` [Optional]

An integer in the range [1..999] representing the accuracy in microseconds.

If any of the attributes are not present a value of 0 is assumed.

The following schema fragment defines the <Accuracy> element:

```
  <element name="Accuracy">
    <complexType>
      <attribute name="seconds" type="nonNegativeInteger"
use="optional" />
      <attribute name="milliseconds" use="optional">
       <simpleType>
         <restriction base="positiveInteger">
            <maxInclusive value="999" />
         </restriction>
        </simpleType>
      </attribute>
      <attribute name="microseconds" use="optional">
       <simpleType>
         <restriction base="positiveInteger">
            <maxInclusive value="999" />
         </restriction>
        </simpleType>
      </attribute>
    </complexType>
  </element>
```

3.8 Element <Ordering>

The <Ordering> element represents whether or not the timestamps issued by this TSA can be strictly ordered based upon the value in the <CreationTime> element. It's semantics and restrictions are identical as the corresonding field in the ASN.1 protocol. The following schema fragment defines the <Ordering> element:

```
  <element name="Ordering" type="boolean" />
```

3.9 Element <Nonce>

The <Nonce> element contains a random value, used only once, to prevent replay attacks and to link requests and responses. The following schema fragment defines the <Nonce> element:

```
  <element name="Nonce" type="integer" />
```

3.10 Element <Extensions>

The <Extensions> element is a generic way to add additional information in the future. The following schema fragment defines the <Extensions> element:

```
  <element name="Extensions">
    <complexType>
      <sequence>
        <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" />
      </sequence>
    </complexType>
  </element>
```

3.11 Element <TimeStampRequest>

The <TimeStampRequest> element is used to request a timestamp on a particular document from a TSA. It contains the following elements:

<Policy> [Optional]

If present, it indicates the policy under which the timestamp token SHOULD be issued.

<Digest> [Required]

Contains the message digest of the document for which the timestamp is being requested.

<Nonce> [Optional]

If present, it contains a random value, used only once, to prevent replay attacks and to link requests and responses.

<Extensions> [Optional]

An extension point for future use.

Do we want to have a version number?

Do we want to have a certReq field as in RFC 3161 that requests TSA certification (or more generally other TSA authentication information) be sent back in the response?

The following schema fragment defines the <TimeStampRequest> element:

```
<element name="TimeStampRequest">
  <complexType>
    <sequence>
      <element ref="ts:Policy" minOccurs="0" />
      <element ref="ts:Digest" />
      <element ref="ts:Nonce" minOccurs="0" />
      <element ref="ts:Extensions" minOccurs="0" />
    </sequence>
  </complexType>
</element>
```

3.12 Element <TimeStampInfo>

The `<TimeStampInfo>` element contains the data that is included within a timestamp token (`<TimeStampResponse>`). It contains the following elements:

`<Policy>` [Required]

Contains the policy under which the token was issued. If the corresponding element appears in the `<TimeStampRequest>` then this element MUST be identical to at least one of those elements.

`<Digest>` [Required]

Contains the message digest of the document being timestamped. This element MUST be identical to the corresponding element in the `<TimeStampRequest>`.

`<SerialNumber>` [Optional]

Contains a serial number for the timestamp token. If present, it MUST be unique for each token issued by a TSA.

`<CreationTime>` [Required]

The time at which the token was issued.

`<Accuracy>` [Optional]

The accuracy of the timestamp.

`<Ordering>` [Optional]

Indicates whether or not timestamps issued by this TSA can be strictly ordered according to the value in the `<CreationTime>` element.

`<Nonce>` [Optional]

MUST be present if and only if the corresponding element is present in the `<TimeStampRequest>` and MUST contain an identical value.

`<Extensions>` [Optional]

An extension point for future use.

Do we want to include a TSA field to identify the TSA. It doesn't seem necessary, but RFC 3161 had one?

The following schema fragment defines the <TimeStampInfo> element:

```
<element name="TimeStampInfo">
  <complexType>
    <sequence>
      <element ref="ts:Policy" />
      <element ref="ts:Digest" />
       <element ref="ts:SerialNumber" minOccurs="0" />
      <element ref="ts:CreationTime" />
      <element ref="ts:Accuracy" minOccurs="0" />
      <element ref="ts:Ordering" minOccurs="0" />
      <element ref="ts:Nonce" minOccurs="0" />
      <element ref="ts:Extensions" minOccurs="0" />
    </sequence>
  </complexType>
</element>
```

3.13 Element <TimeStampResponse>

The `<TimeStampResponse>` element is the response to a `<TimeStampRequest>`. If the request is successful, then it contains the actual timestamp token produced by the TSA. It contains the following elements:

`<StatusInfo>` [Required]

Contains the status of the response.

`<Signature>` [Optional]

Contains the enveloping signature of the TSA over the `<TimeStampInfo>` element. Thus, `<TimeStampInfo>` is inside of this element. When the status attribute of the `<StatusInfo>` element contains the value granted or grantedWithModifications, this element MUST be present. When the status attribute of the `<StatusInfo>` element contains any other value, this element MUST NOT be present. This element is imported from **[XMLSig]**.

The following schema fragment defines the `<TimeStampResponse>` element:

```
<element name="TimeStampResponse">
  <complexType>
    <sequence>
      <element ref="ts:StatusInfo" />
     <!-- enveloping signature with TimeStampInfo inside -->
     <element ref="ds:Signature" minOccurs="0" />
      <element ref="ts:Extensions" minOccurs="0" />
    </sequence>
  </complexType>
</element>
```

4 TIML and XML Signature Syntax and Processing

These are requirements on the relying party.

4.1 Token Validation

We will need to define the process whereby a Time Stamp Token (<TimeStampResponse>) is validated by a relying party.

4.2 Signature Validation

We will also need to define the process of verifying an XML Signature when there is a Time Stamp Token associated with it.

5 Time Stamp Authority Requirements

These are requirements on the Time Stamp Authority.

6 Methods of Identifying the TSA

We will need to think about how to identify the TSA. If X.509 certificates are being used, then the id-kp-timeStamping Extended Key Usage extension from **[RFC3161]** should be used. We will also need to think about how to identify a TSA when SAML is used.

7 References

7.1 Normative

[RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, http://www.ietf.org/rfc/rfc2119.txt, IETF RFC 2119, March 1997.

[RFC3161] C. Adams, P. Cain, D. Pinkas, R. Zuccherato, *Internet X.509 Public Key Infrastructure Time Stamp Protocols*, http://www.ietf.org/rfc/rfc3161.txt, RFC 3161, August 2001.

[XMLSig] D. Eastlake et al., XML-Signature Syntax and Processing, http://www.w3.org/TR/xmldsig-core/, World Wide Web Consortium.