Send comments to:
Phillip Hallam-Baker, Senior Author
401 Edgewater Place, Suite 280
Wakefield MA 01880
Tel 781 245 6996 x227
Email: pbaker@verisign.com

X-TAML: XML Trust Axiom Markup Language

Phillip Hallam-Baker *VeriSign*

Draft Version 0.12: October 17th 2001

Printed on Wednesday, October 17, 2001

X-TAML: XML Trust Axiom Markup Language

Version 0.12

Table Of Contents

Printed on Wednesday, October 17, 2001

Table of Figures

Executive Summary

The XML Trust Axiom Markup Language (XTAML) defines SAML Trust Assertions that support the management of trust axioms. A trust axiom is analogous to a root certificate in a certificate based PKI. An important application of trust axioms is managing the trust relationship between a client and a trust service.

XTAML is layered on the Security Assertion Markup Language (SAML). XTAML defines statement elements for specifying axiomatic and delegate keys and for asserting the validity status of another assertion. A new condition element is defined that makes the validity status of an assertion dependent on online verification. Two new advice elements are defined to allow an assertion to provide advice on the reissue of the assertion and for issue of related assertions.

1 Introduction

This document describes mechanisms that support management of long-term trust relationships between parties.

This specification is intended to complement other XML security standards and proposals, in particular XML Signature [XML-SIG], XML Encryption [XML-ENC], XML Key Management [XKMS] and Security Assertion Markup Language [SAML].

1.1 Introduction to this Document

X-TAML supports management of Trust Axioms. A Trust Axiom defines a trust relationship that is not dependent on the existence of any other trust relationship. A trust axiom typically defines a long-term trust relationship such as that embedded in client devices. An important application of trust axioms is their use to establish a trust relationship between a client and a Trust Service such as XKMS or SAML.

X-TAML provides a general mechanism for reporting assertion status. All assertions carry a unique identifier specified by means of a URI. Meta-Assertions may be issued that make specific claims about the status of a single assertion or a group of assertions.

1.2 Structure of this document

The remainder of this document describes the XTAML Specification.

Section 2: Architecture
The XTAML Architecture is described

Section 3: Message Set.
The syntax and semantics of the protocol messages is defined.

2 Architecture

XTAML is layered on SAML and defines additional trust assertion elements to be defined to support the following functions:

?? Declaration and delegation of trust axioms

?? Assertion Status Management

?? Online Verification Conditions

?? Assertion Issue and Reissue Advice

2.1 Trust Axiom Declaration and Delegation

A Trust Axiom defines a trust relationship that is not dependent on the existence of any other trust relationship. A trust axiom typically defines a long-term trust relationship and is equivalent to a ‘root of trust’ in a certificate based PKI.

Every system of trust must ultimately be built on assertions that are accepted axiomatically. In a certificate based PKI the trust axioms are one or more ‘roots of trust’ that are typically represented by a self signed digital certificate.

Compromise of a trust axiom is a serious matter with limited scope for mitigation. For this reason it is generally preferable that the private keys corresponding to widely distributed trust axioms be maintained in highly secure facilities that do not have any direct or indirect connection to any external network. Such an ‘offline key’ is typically used for the sole purpose of signing periodic assertions that delegate authority to an ‘online key’.

2.1.1 Trust Axiom Declaration

A Trust Axiom declaration defines a key that is to be trusted axiomatically. A trust axiom may constrain the use of delegation using the `MaximumChainLength` attribute which specifies the maximum length of a chain of delegation assertions that are to be considered trustworthy.

Example: The following assertion states that the RSA key with the specified modulus and exponent values is a trust axiom. The key may be used to sign two levels of key delegation assertions below the axiomatic key. The assertion also contains advice that indicates that a key delegation assertion shall be issued as described in section 2.4 below.

```
<?xml version="1.0" encoding="UTF-8"?>
<saml:Assertion
   xmlns:xtaml="http://www.xmltrustcenter.org/schema/xtaml-01.xsd"
   xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
   xmlns:saml="http://www.oasis-
open.org/committees/security/docs/draft-sstc-schema-assertion-18.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.xmltrustcenter.org/schema/xtaml-
01.xsd xtaml-01.xsd"
```

```
   AssertionID="http://www.xmltrustcenter.org/xtaml/2001-09-24/0112"
   IssueInstant="2001-09-24T12:01:00Z"
   Issuer="http://www.xmltrustcenter.org/"
   MajorVersion="1" MinorVersion="0">
   <saml:Advice>
      <xtaml:IssueAdvice
         Schema="http://www.xmltrustcenter.org/schema/xtaml-01.xsd"
         Statement="KeyDelegationStatement">
   <xtaml:Location>http://www.xmltrustcenter.org/xtaml/issue/MasterKey<
/xtaml:Location>
      </xtaml:IssueAdvice>
   </saml:Advice>
   <xtaml:AxiomaticKeyStatement>
      <xtaml:AxiomaticKey>
         <ds:KeyInfo>
            <ds:KeyName>MasterKey</ds:KeyName>
            <ds:KeyValue>
               <ds:RSAKeyValue>

   <ds:Modulus>998/T2PUN8HQlnhf9YIKdMHHGM7HkJwA56UD0a1oYq
7EfdxSXAidruAszNqBoOqfarJIsfcVKLob1hGnQ/l6xw==</ds:Modulus>
                  <ds:Exponent>AQAB</ds:Exponent>
               </ds:RSAKeyValue>
            </ds:KeyValue>
         </ds:KeyInfo>
      </xtaml:AxiomaticKey>
      <xtaml:Constraints MaximumChainLength="2"/>
   </xtaml:AxiomaticKeyStatement>
</saml:Assertion>
```

Note that the Offline Root does not specify a validity interval constraint. Although there is a risk that the Offline Root might be compromised there is no means of mitigating the consequences of the compromise or otherwise recovering from them except by out of band means.

2.1.2 Key Delegation

X-TAML supports a limited delegation mechanism to support online/offline key management. Delegation is ‘all or nothing’, that is the key to which authority is delegated is directly equivalent to the signing key for the period of the assertion. Constrained delegation where the delegated key has signing authority within a limited name space, corresponding to mutual key recognition agreements between peers (i.e. cross certification) is NOT SUPPORTED.

A Delegation Assertion MUST be signed with a Digital Signature. The `KeyInfo` element contained within the `KeyAssertion` must specify the `KeyValue` of the key to which signing authority is delegated.

As with a trust axiom assertion, a key delegation assertion may constrain further delegation using the `MaximumChainLength` attribute with the length of the chain being measured from the Delegate key down. If multiple chain length constraints are specified all must be satisfied.

Example : The following example shows an assertion delegating signing authority from the offline key described in section 2.1.1 above to an online signing key. The delegation assertion does not permit any further delegation and is valid for one year. A new delegation may be obtained before expiry from the location specified as described in section 2.4 below.

```
<?xml version="1.0" encoding="UTF-8"?>
<saml:Assertion
   xmlns:xtaml="http://www.xmltrustcenter.org/schema/xtaml-01.xsd"
   xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
   xmlns:saml="http://www.oasis-
open.org/committees/security/docs/draft-sstc-schema-assertion-18.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.xmltrustcenter.org/schema/xtaml-
01.xsd xtaml-01.xsd"
   AssertionID="http://www.xmltrustcenter.org/xtaml/2001-09-24/0114"
   IssueInstant="2001-09-24T12:01:00Z"
   Issuer="http://www.xmltrustcenter.org/"
   MajorVersion="1" MinorVersion="0">
   <saml:Conditions
      NotBefore="2001-09-24T12:01:00Z"
      NotOnOrAfter="2002-09-24T00:00:00Z"/>
   <saml:Advice>
      <xtaml:ReissueAdvice
         Earliest="2002-08-24T00:00:00Z" Latest="2002-09-23T00:00:00Z">
   <xtaml:Location>http://www.xmltrustcenter.org/xtaml/issue/MasterKey<
/xtaml:Location>
      </xtaml:ReissueAdvice>
   </saml:Advice>
   <xtaml:KeyDelegationStatement>
      <xtaml:MasterKey>
         <ds:KeyInfo>
            <ds:KeyName>MasterKey</ds:KeyName>
            <ds:KeyValue>
               <ds:RSAKeyValue>

   <ds:Modulus>998/T2PUN8HQlnhf9YIKdMHHGM7HkJwA56UD0a1oYq
7EfdxSXAidruAszNqBoOqfarJIsfcVKLob1hGnQ/l6xw==</ds:Modulus>
                  <ds:Exponent>AQAB</ds:Exponent>
               </ds:RSAKeyValue>
            </ds:KeyValue>
         </ds:KeyInfo>
      </xtaml:MasterKey>
      <xtaml:DelegateKey>
         <ds:KeyInfo>
            <ds:KeyName>DelegateKey</ds:KeyName>
            <ds:KeyValue>
               <ds:RSAKeyValue>

   <ds:Modulus>998AW59twgnWAas35SszETWset973w3trkjsde9735
hkjaerf0wa5302qih3r0a8qw3j0953rsdjfaq309850u==</ds:Modulus>
                  <ds:Exponent>AQAB</ds:Exponent>
               </ds:RSAKeyValue>
            </ds:KeyValue>
         </ds:KeyInfo>
      </xtaml:DelegateKey>
      <xtaml:Constraints MaximumChainLength="0"/>
   </xtaml:KeyDelegationStatement>
</saml:Assertion>
```

2.2 Assertion Status Management

XTAML allows assertions to make assertions about assertions. Such Meta-Assertions provide equivalent functionality to the Certificate Revocation List and Online Certificate Status Protocols of PKIX.

XTAML supports two types of status query:

?? Querying the status of a statement made by an assertion

?? Querying the status of the asser tion itself

The distinction between the two types of query is not always material.

When liability insurance or other layered services are bound to an assertion there is however a considerable difference between repeating the question and asking if a previous answer is still valid.

A second application of meta-assertions is distribution of assertion status information between distribution points. Figure 1 shows an architecture in which Trust services acting as local trust distribution points communicate with a Registration Service via the Tier 4 protocol but support only tier 1 & 2 queries from end users.

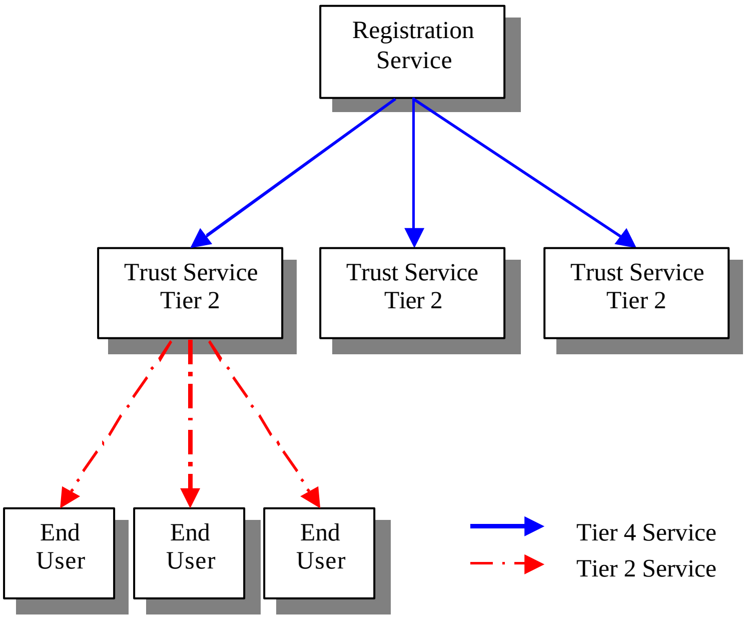

Figure 1: Distribution of Trust Data Between Distribution Points

A third application of meta-assertions is to alert systems that have previously obtained an assertion of a change in the status of an assertion they might otherwise continue to rely on.

The `Declare` Assertion is used to specify the status of a set of TASS assertions. The Declare assertion is designed to permit the status of individual assertions or collections of assertions to be specified.

The Declare assertion is designed to support both dynamic and static signing of responses. Dynamic responses are signed in response to specific requests. Static responses are signed in advance of any request and MAY anticipate multiple requests.

```
<?xml version="1.0" encoding="UTF-8"?>
<saml:Assertion
   xmlns:xtaml="http://www.xmltrustcenter.org/schema/xtaml-01.xsd"
   xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
   xmlns:saml="http://www.oasis-
open.org/committees/security/docs/draft-sstc-schema-assertion-18.xsd"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.xmltrustcenter.org/schema/xtaml-
01.xsd xtaml-01.xsd"
   AssertionID="http://www.xmltrustcenter.org/xtaml/2001-09-24/0124"
   IssueInstant="2001-09-24T12:01:00Z"
   Issuer="http://www.xmltrustcenter.org/"
   MajorVersion="1" MinorVersion="0">
   <saml:Conditions
      NotBefore="2001-09-24T12:01:00Z"
      NotOnOrAfter="2002-09-24T00:00:00Z"/>
   <saml:Advice>
      <xtaml:ReissueAdvice
         Earliest="2002-08-24T00:00:00Z" Latest="2002-09-23T00:00:00Z">
   <xtaml:Location>http://www.xmltrustcenter.org/xtaml/issue/MasterKey<
/xtaml:Location>
      </xtaml:ReissueAdvice>
   </saml:Advice>
   <xtaml:StatusStatement>
      <xtaml:DeclareStatusRange
         First="http://www.xmltrustcenter.org/xtaml/2001-09-24/0001"
         Last="http://www.xmltrustcenter.org/xtaml/2001-09-24/0123"
         Terminal="false" ValidityStatus="Valid"/>
      <xtaml:DeclareStatus
      AssertionID="http://www.xmltrustcenter.org/xtaml/2001-09-24/0021"
         Terminal="true" ValidityStatus="Invalid"/>
      <xtaml:DeclareStatus
      AssertionID="http://www.xmltrustcenter.org/xtaml/2001-09-24/0054"
         Terminal="true" ValidityStatus="Invalid"/>
   </xtaml:StatusStatement>
</saml:Assertion>
```

2.3 Online Verification Conditions

In certain circumstances it is desirable to make a claim that is conditional on obtaining online verification at the time of use.

In many business applications where layered services (e.g. insurance) are tied to an assertion it is not the information itself that is acted on but the acceptance of responsibility. In other applications the status of an assertion might be exceptionally volatile requiring verification each time it is used. Use of a template assertion means that the statement is bound to a single assertion id rather than one for each query response.

For example the following assertion specifies that its status MUST be verified by reference to the specified service to be considered trustworthy:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.0 U beta 3.1 build Aug 27 2001
(http://www.xmlspy.com) by Phillip Hallam-Baker (Phillip Hallam-Baker)
-->
<saml:Assertion
xmlns:xtaml="http://www.xmltrustcenter.org/schema/xtaml-01.xsd"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:saml="http://www.oasis-open.org/committees/security/docs/draft-
sstc-schema-assertion-18.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.xmltrustcenter.org/schema/xtaml-01.xsd
xtaml-01.xsd" AssertionID="http://www.xmltrustcenter.org/xtaml/2001-09-
24/0116" IssueInstant="2001-09-24T12:01:00Z"
Issuer="http://www.xmltrustcenter.org/" MajorVersion="1"
MinorVersion="0">
   <saml:Conditions>
      <xtaml:VerificationCondition>

   <xtaml:Location>http://www.xmltrustcenter.org/xtaml/issue/MasterKey<
/xtaml:Location>
      </xtaml:VerificationCondition>
   </saml:Conditions>
   <xtaml:AxiomaticKeyStatement>
```

```
      <xtaml:AxiomaticKey>
         <ds:KeyInfo>
            <ds:KeyName>MasterKey</ds:KeyName>
            <ds:KeyValue>
               <ds:RSAKeyValue>

   <ds:Modulus>998/T2PUN8HQlnhf9YIKdMHHGM7HkJwA56UD0a1oYq7EfdxSXAidruAs
zNqBoOqfarJIsfcVKLob1hGnQ/l6xw==</ds:Modulus>
                  <ds:Exponent>AQAB</ds:Exponent>
               </ds:RSAKeyValue>
            </ds:KeyValue>
         </ds:KeyInfo>
      </xtaml:AxiomaticKey>
      <xtaml:Constraints MaximumChainLength="1"/>
   </xtaml:AxiomaticKeyStatement>
</saml:Assertion>
```

The actual status of the assertion is determined by means of the Tier4 status declaration service.

2.4 Assertion Issue and Reissue Advice

Assertions may specify a validity interval. Such an assertion is typically be reissued before it expires. XTAML provides a mechanism whereby clients are informed of the location(s) from which the reissued assertion SHOULD be available if it is reissued and the earliest and latest times at which the assertion SHOULD be obtained.

For example the key delegation assertion described in section 2.1 states that it will be reissued from a single locations on 8th August 2002 and will be available for a month:

```
<?xml version="1.0" encoding="UTF-8"?>
<saml:Assertion
   …>
   <saml:Conditions
      NotBefore="2001-09-24T12:01:00Z"
      NotOnOrAfter="2002-09-24T00:00:00Z"/>
   <saml:Advice>
      <xtaml:ReissueAdvice
         Earliest="2002-08-24T00:00:00Z" Latest="2002-09-23T00:00:00Z">
   <xtaml:Location>http://www.xmltrustcenter.org/xtaml/issue/MasterKey<
/xtaml:Location>
      </xtaml:ReissueAdvice>
   </saml:Advice>
   <xtaml:KeyDelegationStatement>
      …
   </xtaml:KeyDelegationStatement>
</saml:Assertion>
```

Clients SHOULD attempt to avoid overloading reissue servers that would occur if they scheduled the download of the reissued assertion for the earliest moment it is available. This MAY be achieved by scheduling the download at a randomly chosen instant between the earliest and latest time instant specified.

Assertions may also contain advice that describes the issue of a related assertion. For example the axiomatic key assertion described in section 2.1 states that a corresponding key delegation assertion is issued from a single location with a given schema and statement element:

```
<?xml version="1.0" encoding="UTF-8"?>
<saml:Assertion
   …>
   <saml:Advice>
      <xtaml:IssueAdvice
         Schema="http://www.xmltrustcenter.org/schema/xtaml-01.xsd"
         Statement="KeyDelegationStatement">
   <xtaml:Location>http://www.xmltrustcenter.org/xtaml/issue/MasterKey<
/xtaml:Location>
      </xtaml:IssueAdvice>
   </saml:Advice>
   <xtaml:AxiomaticKeyStatement>
   …
   </xtaml:AxiomaticKeyStatement>
</saml:Assertion>
```

3 Syntax

3.1 Namespace

In this document, certain namespace prefixes represent specific XML namespaces.

All SAML protocol elements are defined using XML schema[XML-Schema1][XML-Schema2]. For clarity unqualified elements in schema definitions are in the XML schema namespace:

```
   xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

References to XML Trust Axiom Markup Language schema defined herein use the prefix `xtaml:` and are in the namespace:

```
   xmlns:xtaml="http://www.xmltrustcenter.org/schema/xtaml-01.xsd"
```

References to Security Assertion Markup Language schema use the prefix `saml:` and are in the namespace:

```
   xmlns:saml="http://www.oasis-
   open.org/committees/security/docs/draft-sstc-schema-assertion-
   18.xsd"
```

The SAML and XTAML schema specifications use some elements already defined in the XML Signature namespace. The “XML Signature namespace” is represented by the prefix `ds` and is declared as:

```
   xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
```

The “XML Signature schema” is defined in [XML-SIG] and the `<ds:KeyInfo>` element (and all of its contents) are defined in [XML-SIG]§4.4.

The following schema specifies the schema namespaces:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by Phill
Hallam-Baker (VeriSign Inc.) -->
<schema
   targetNamespace="http://www.xmltrustcenter.org/schema/xtaml-01.xsd"
   xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
   xmlns:saml="http://www.oasis-
open.org/committees/security/docs/draft-sstc-schema-assertion-18.xsd"
   xmlns:xtaml="http://www.xmltrustcenter.org/schema/xtaml-01.xsd"
   xmlns="http://www.w3.org/2001/XMLSchema"
   elementFormDefault="unqualified">
   <import namespace="http://www.oasis-
open.org/committees/security/docs/draft-sstc-schema-assertion-18.xsd"
   schemaLocation="draft-sstc-schema-assertion-18.xsd"/>
   <import namespace="http://www.w3.org/2000/09/xmldsig#"
      schemaLocation="xmldsig-core-schema.xsd"/>
   <annotation>
      <documentation>xtaml-01.xsd</documentation>
   </annotation>
```

3.1.1 Basic Types

The type `ValidityStatusType` specifies the validity status of an assertion:

`Valid`
The assertion is valid.

`Invalid`
The assertion is invalid.

`Indeterminate`
The validity of the assertion cannot be determined.

The following schema defines the `<ValidityStatusType>` element:

```
   <!-- ValidityStatusType -->
   <simpleType name="ValidityStatusType">
      <restriction base="string">
         <enumeration value="Valid"/>
         <enumeration value="Invalid"/>
         <enumeration value="Indeterminate"/>
      </restriction>
   </simpleType>
```

3.2 Key Management

3.2.1 Element: `<AxiomaticKeyStatement>`

The `<KeyDelegationStatement>` element contains the following elements:

`<AxiomaticKey>` [Required]

The `<AxiomaticKey>` element specifies the key that is to provide an axiom of trust.

`<Constraints>` [Required]

The `<Constraints>` element specifies constraints on the use of the key as an axiom of trust.

The following schema defines the `<AxiomaticKeyStatement>` element:

```
<!-- AxiomaticKeyStatement -->
<element name="AxiomaticKeyStatement"
   type="xtaml:AxiomaticKeyStatementType"
   substitutionGroup="saml:Statement"/>
<complexType name="AxiomaticKeyStatementType">
   <complexContent>
      <extension base="saml:StatementAbstractType">
         <sequence>
            <element ref="xtaml:AxiomaticKey"/>
            <element ref="xtaml:Constraints"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
```

3.2.2 Element: `<KeyDelegationStatement>`

The `<KeyDelegationStatement>` element contains the following elements:

`<MasterKey>` [Required]

The `<MasterKey>` element specifies the trusted key from which trust is being delegated.

`<DelegateKey>` [Required]

The `<DelegateKey>` element specifies the key to which trust is being delegated.

`<Constraints>` [Required]

The `<Constraints>` element specifies constraints on the use of the key as an axiom of trust.

The following schema defines the <KeyDelegationStatement> element:

```
<!-- KeyDelegationStatement-->
<element name="KeyDelegationStatement"
   type="xtaml:KeyDelegationStatementType"
   substitutionGroup="saml:Statement"/>
<complexType name="KeyDelegationStatementType">
   <complexContent>
      <extension base="saml:StatementAbstractType">
         <sequence>
            <element ref="xtaml:MasterKey"/>
            <element ref="xtaml:DelegateKey"/>
            <element ref="xtaml:Constraints"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
```

3.2.3 Constraints

The <Constraints> element contains the following attribute:

MaximumChainLength [Optional]

The MaximumChainLength attribute specifies the maximum number of delegations that are permitted beneath the specified key. If the value of MaximumChainLength is negative or zero delegation is not permitted.

The following schema defines the <Constraints> element:

```
<!-- Constraints -->
<element name="Constraints" type="xtaml:ConstraintsType"/>
<complexType name="ConstraintsType">
   <attribute name="MaximumChainLength" type="integer"/>
</complexType>
```

3.2.4 Elements <AxiomaticKey>, <MasterKey> and <DelegateKey>

The <AxiomaticKey>, <MasterKey> and <DelegateKey> elements are of KeyBindingType and contain the following element:

<ds:KeyInfo> [Any Number]

The <ds:KeyInfo> element specifies the cryptographic key.

The following schema defines the `<AxiomaticKey>`, `<MasterKey>` and `<DelegateKey>` elements:

```
<!-- KeyBindingType -->
<element name="AxiomaticKey" type="xtaml:KeyBindingType"/>
<element name="MasterKey" type="xtaml:KeyBindingType"/>
<element name="DelegateKey" type="xtaml:KeyBindingType"/>
<complexType name="KeyBindingType">
   <sequence>
      <element ref="ds:KeyInfo"/>
   </sequence>
</complexType>
```

3.3 Validity Status Management

The `<StatusStatement>` element asserts the validity of an assertion identified by a URI as follows:

?? A `Declare` structure *matches* a URI if and only if the URI is greater or equal to the attribute First and less than or equal to the value Last.

?? A `Declare` structure is *terminal* if the value of the Terminal attribute is true.

?? The validity asserted by a `Declare` structure is `Valid` if the value of the `Valid` attribute is `True` and `Invalid` otherwise.

3.3.1 Element: `<StatusStatement>`

The `<StatusStatement>` element contains the following elements in any order:

`<DeclareStatus>` [Any Number]
The `<DeclareStatus>` element declares the validity status of a single assertion.

`<DeclareStatusRange>` [Any Number]
The `<DeclareStatusRange>` element declares the validity status of a range of assertions.

The following schema defines the `<StatusStatement>` element:

```
<!-- StatusStatement -->
<element name="StatusStatement"
   type="xtaml:StatusStatementType"
   substitutionGroup="saml:Statement"/>
<complexType name="StatusStatementType">
   <complexContent>
      <extension base="saml:StatementAbstractType">
         <choice minOccurs="0" maxOccurs="unbounded">
            <element ref="xtaml:DeclareStatus"/>
            <element ref="xtaml:DeclareStatusRange"/>
         </choice>
      </extension>
   </complexContent>
</complexType>
```

3.3.2 Element: <DeclareStatus>

The <DeclareStatusRange> element contains the following attributes:

AssertionID [Required]

The declaration matches an assertion if and only if its assertion identifier is equal to the value of the AssertionID attribute.

ValidityStatus [Required]

If the declaration matches the assertion, the provisional status value is replaced with the value of the ValidityStatus attribute.

Terminal [Required]

The Terminal attribute specifies the processing behavior when a match is found. If the value of terminal is true processing of the declaration list terminates if the declaration matches, otherwise processing continues.

The following schema defines the <DeclareStatus> element:

```
   <!-- DeclareStatus-->
   <element name="DeclareStatus" type="xtaml:DeclareStatusType"/>
   <complexType name="DeclareStatusType">
      <attribute name="ValidityStatus"
type="xtaml:ValidityStatusType"/>
      <attribute name="AssertionID" type="anyURI"/>
      <attribute name="Terminal" type="boolean"/>
   </complexType>
```

3.3.3 Element <DeclareStatusRange>

The <DeclareStatusRange> element contains the following attributes:

First [Optional] and Last [Optional]

A declaration matches an assertion if and only if its assertion identifier is greater than or equal to the value of the First attribute and less than or equal to the value of the Last attribute.

ValidityStatus [Required]

If the declaration matches the assertion, the provisional status value is replaced with the value of the ValidityStatus attribute.

Terminal [Required]

The Terminal attribute specifies the processing behavior when a match is found. If the value of terminal is true processing of the declaration list terminates if the declaration matches, otherwise processing continues.

The following schema defines the <DeclareStatusRange> element:

```
   <!-- DeclareStatusRange-->
   <element name="DeclareStatusRange"
type="xtaml:DeclareStatusRangeType"/>
   <complexType name="DeclareStatusRangeType">
      <attribute name="ValidityStatus"
type="xtaml:ValidityStatusType"/>
      <attribute name="First" type="anyURI"/>
      <attribute name="Last" type="anyURI"/>
      <attribute name="Terminal" type="boolean"/>
   </complexType>
```

3.4 Online Verification Condition

3.4.1 Element: <VerificationCondition>

The <VerificationCondition> element contains the following element:

<Location> [Any Number]

The <Location> element specifies the location from which the assertion will be issued.

The following schema defines the <VerificationCondition> element:

```
   <!-- Verification Condition -->
   <element name="VerificationCondition"
         type="xtaml:VerificationConditionType"
         substitutionGroup="saml:Condition"/>
   <complexType name="VerificationConditionType">
      <complexContent>
            <sequence>
               <element ref="xtaml:Location"
                  minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
      </complexContent>
   </complexType>
```

3.5 Reissue & Issue Advice

3.5.1 Element: <ReissueAdvice>

The <ReissueAdvice> element contains the following elements and attributes:

Earliest [Required]

The Earliest attribute specifies the earliest time instant at which a client should attempt to obtain the assertion.

Latest [Required]

The Latest attribute specifies the latest time instant at which a client should attempt to obtain the assertion.

`<Location>` [Any Number]

The `<Location>` element specifies the location from which the assertion will be issued.

The following schema defines the `<ReissueAdvice>` element:

```
   <!-- Reissue Advice -->
   <element name="ReissueAdvice" type="xtaml:ReissueAdviceType"
      substitutionGroup="saml:AdviceElement"/>
   <complexType name="ReissueAdviceType">
      <complexContent>
         <extension base="saml:AdviceAbstractType">
            <sequence>
               <element ref="xtaml:Location"
                  minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
            <attribute name="Earliest" type="dateTime"/>
            <attribute name="Latest" type="dateTime"/>
         </extension>
      </complexContent>
   </complexType>
   <element name="Location" type="anyURI"/>
```

3.5.2 Element <IssueAdvice>

The `<IssueAdvice>` element contains all the elements and attributes of a `<ReissueAdvice>` element and adds the following attributes:

`Schema` [Optional]

The `Schema` attribute specifies the SAML extension schema of the issued assertion. If the schema attribute is omitted the default is the SAML schema.

`Statement` [Required]

The `Statement` attribute specifies the statement element of the issued assertion.

The following schema defines the `<IssueAdvice>` element:

```
   <!-- Issue Advice -->
   <element name="IssueAdvice" type="xtaml:IssueAdviceType"
substitutionGroup="saml:AdviceElement"/>
   <complexType name="IssueAdviceType">
      <complexContent>
         <extension base="xtaml:ReissueAdviceType">
            <attribute name="Schema" type="anyURI"/>
            <attribute name="Statement" type="string"/>
         </extension>
      </complexContent>
   </complexType>
</schema>
```

4 Acknowlegements

The author would like to acknowledge the following people who provided assistance in this work: Warwick Ford, Thane Plambeck (VeriSign), Barbara Fox, Blair Dillaway, Brian LaMacchia (Microsoft), Jeremy Epstein, Joe Lapp (webMethods), David Solo (CitiGroup), Mack Hicks (Bank of America), Stephen Farrell (Baltimore Technologies), Prateek Mishra (Netegrity).

Appendix A Collected Syntax

The XTAML schema elements are collected below. In the case of a discrepancy the collected schema is authoritative.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
   targetNamespace="http://www.xmltrustcenter.org/schema/xtaml-01.xsd"
   xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
   xmlns:saml="http://www.oasis-
open.org/committees/security/docs/draft-sstc-schema-assertion-18.xsd"
   xmlns:xtaml="http://www.xmltrustcenter.org/schema/xtaml-01.xsd"
   xmlns="http://www.w3.org/2001/XMLSchema"
   elementFormDefault="unqualified">
   <import namespace="http://www.oasis-
open.org/committees/security/docs/draft-sstc-schema-assertion-18.xsd"
   schemaLocation="draft-sstc-schema-assertion-18.xsd"/>
   <import namespace="http://www.w3.org/2000/09/xmldsig#"
      schemaLocation="xmldsig-core-schema.xsd"/>
   <annotation>
      <documentation>xtaml-01.xsd</documentation>
   </annotation>
   <!-- ValidityStatusType -->
   <simpleType name="ValidityStatusType">
      <restriction base="string">
         <enumeration value="Valid"/>
         <enumeration value="Invalid"/>
         <enumeration value="Indeterminate"/>
      </restriction>
   </simpleType>
   <!-- AxiomaticKeyStatement -->
   <element name="AxiomaticKeyStatement"
      type="xtaml:AxiomaticKeyStatementType"
      substitutionGroup="saml:Statement"/>
   <complexType name="AxiomaticKeyStatementType">
      <complexContent>
         <extension base="saml:StatementAbstractType">
            <sequence>
               <element ref="xtaml:AxiomaticKey"/>
               <element ref="xtaml:Constraints"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- KeyDelegationStatement-->
   <element name="KeyDelegationStatement"
      type="xtaml:KeyDelegationStatementType"
      substitutionGroup="saml:Statement"/>
   <complexType name="KeyDelegationStatementType">
      <complexContent>
         <extension base="saml:StatementAbstractType">
            <sequence>
               <element ref="xtaml:MasterKey"/>
               <element ref="xtaml:DelegateKey"/>
               <element ref="xtaml:Constraints"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- Constraints -->
   <element name="Constraints" type="xtaml:ConstraintsType"/>
   <complexType name="ConstraintsType">
      <attribute name="MaximumChainLength" type="integer"/>
   </complexType>
   <!-- KeyBindingType -->
   <element name="AxiomaticKey" type="xtaml:KeyBindingType"/>
   <element name="MasterKey" type="xtaml:KeyBindingType"/>
   <element name="DelegateKey" type="xtaml:KeyBindingType"/>
   <complexType name="KeyBindingType">
      <sequence>
         <element ref="ds:KeyInfo"/>
      </sequence>
   </complexType>
   <!-- StatusStatement -->
   <element name="StatusStatement" type="xtaml:StatusStatementType"
      substitutionGroup="saml:Statement"/>
   <complexType name="StatusStatementType">
      <complexContent>
         <extension base="saml:StatementAbstractType">
            <sequence>
               <element ref="xtaml:Declare"
                  minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- DeclareStatus-->
   <element name="DeclareStatus" type="xtaml:DeclareStatusType"/>
   <complexType name="DeclareStatusType">
      <attribute name="ValidityStatus"
type="xtaml:ValidityStatusType"/>
      <attribute name="AssertionID" type="anyURI"/>
      <attribute name="Terminal" type="boolean"/>
   </complexType>
   <!-- DeclareStatusRange-->
   <element name="DeclareStatusRange"
      type="xtaml:DeclareStatusRangeType"/>
   <complexType name="DeclareStatusRangeType">
      <attribute name="ValidityStatus"
type="xtaml:ValidityStatusType"/>
      <attribute name="First" type="anyURI"/>
      <attribute name="Last" type="anyURI"/>
      <attribute name="Terminal" type="boolean"/>
   </complexType>
   <!-- Verification Condition -->
   <element name="VerificationCondition"
      type="xtaml:VerificationConditionType"
      substitutionGroup="saml:Condition"/>
   <complexType name="VerificationConditionType">
      <complexContent>
         <extension base="saml:StatementAbstractType">
            <sequence>
```

```
               <element ref="xtaml:Location"
                  minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- Reissue Advice -->
   <element name="ReissueAdvice" type="xtaml:ReissueAdviceType"
      substitutionGroup="saml:AdviceElement"/>
   <complexType name="ReissueAdviceType">
      <complexContent>
         <extension base="saml:AdviceAbstractType">
            <sequence>
               <element ref="xtaml:Location"
                  minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
            <attribute name="Earliest" type="dateTime"/>
            <attribute name="Latest" type="dateTime"/>
         </extension>
      </complexContent>
   </complexType>
   <element name="Location" type="anyURI"/>
</schema>
```

Appendix B References

[RFC-2396] T. Berners-Lee, R. Fielding and L. Masinter. *Uniform Resource Identifiers (URI): Generic Syntax* RFC 2396, August 1998, Internet Engineering Taskforce. http://www.rfc-editor.org/rfc/rfc2396.txt.

[SAML] *OASIS Security Assertion Markup Language* http://www.oasis-open.org/committees/security/

[SOAP] D. Box, D Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Frystyk Nielsen, S Thatte, D. Winer. *Simple Object Access Protocol (SOAP) 1.1*, W3C Note 08 May 2000, http://www.w3.org/TR/SOAP

[WDSL] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, *Web services Description Language (WSDL) 1.0* September 25, 2000, http://msdn.microsoft.com/xml/general/wsdl.asp

[XKMS] Warwick Ford, Phillip Hallam-Baker, Barbara Fox, Blair Dillaway, Brian LaMacchia, Jeremy Epstein, Joe Lapp, *XML Key Management Specification (XKMS)*, W3C Note 30 March 2001 http://www.w3.org/TR/xkms/

[XML-Schema1] H. S. Thompson, D. Beech, M. Maloney, N. Mendelsohn. *XML Schema Part 1: Structures*, W3C Working Draft 22 September 2000, http://www.w3.org/TR/xmlschema-1/

[XML-Schema2] P. V. Biron, A. Malhotra, *XML Schema Part 2: Datatypes*; W3C Working Draft 22 September 2000, http://www.w3.org/TR/xmlschema-2/

[XML-SIG] D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer , B. Fox , E. Simon. *XML-Signature Syntax and Processing*, World Wide Web Consortium. http://www.w3.org/TR/xmldsig-core/

Appendix C Legal Notices

Copyright

Intellectual Property Statement

Neither the authors of this document, nor their companies take any position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither do they represent that they have made any effort to identify any such rights.

Disclaimer

Printed on Wednesday, October 17, 2001

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