Liberty Authentication Context Specification

Version 1.0

11 July 2002

Document Description: liberty-architecture-authentication-context-v1.0

Notice

Liberty Alliance Project
Licensing Administrator
c/o IEEE-ISTO
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA

Editor

Paul Madsen, Entrust, Inc.

Contributors

The following Liberty Alliance Project Sponsor companies contributed to the development of this specification:

ActivCard
American Express Travel Related Services
America Online, Inc.
Bank of America
Bell Canada
Catavault
Cingular Wireless
Cisco Systems, Inc.
Citigroup
Cyberun Corporation
Deloitte & Touche LLP
EarthLink, Inc.
Electronic Data Systems, Inc.
Entrust, Inc.
Ericsson
Fidelity Investments
France Telecom
Gemplus
General Motors
Hewlett-Packard Company
i2 Technologies, Inc.
Intuit Inc.
MasterCard International
Nextel Communications
Nippon Telegraph and Telephone Company
Nokia Corporation
Novell, Inc.
NTT DoCoMo, Inc.
OneName Corporation
Openwave Systems Inc.
PricewaterhouseCoopers LLP
Register.com
RSA Security Inc
Sabre Holdings Corporation
SAP AG
SchlumbergerSema
Sony Corporation
Sun Microsystems, Inc.
United Airlines
VeriSign, Inc.
Visa International
Vodafone Group Plc
Wave Systems

Table of Contents

1 Introduction

This specification defines a syntax for the definition of authentication context statements and an initial list of Liberty authentication context classes.

1.1 Notation

This specification uses schema documents conforming to W3C XML schema (see [Schema1]) and normative text to describe the syntax and semantics of XML-encoded SAML assertions and protocol messages. Note: Phrases and numbers in brackets [ ] refer to other documents; details of these references can be found in Section 5 (at the end of this document).

The key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this specification are to be interpreted as described in [RFC2119]: "they MUST only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)."

These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

Note: Non-normative notes and explanations appear like this.

```
Listings of XML schemas appear like this.
```

```
Example code listings appear like this.
```

Conventional XML namespace prefixes are used throughout the listings in this specification to stand for their respective namespaces as follows, regardless of whether a namespace declaration is present in the example:

- The prefix lib: stands for the Liberty namespace (`http://projectliberty.org`)
- The prefix saml: stands for the SAML assertion namespace (`http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-assertion-15.xsd`).
- The prefix samlp: stands for the SAML request-response protocol namespace (`http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-protocol-15.xsd`).
- The prefix ds: stands for the W3C XML signature namespace (`http://www.w3.org/2000/09/xmldsig#`).
- The prefix xsd: stands for the W3C XML schema namespace in example listings (`http://www.w3.org/2001/XMLSchema`). In schema listings, this namespace is the default, and no prefix is shown.

Definitions for Liberty-specific terms can be found in [LibertyGloss].

2 Overview

Liberty will not prescribe a single technology, protocol, or policy for the processes by which identity providers issue identities to Principals and by which those Principals subsequently authenticate themselves to the identity provider. Different identity providers will choose different technologies, follow different processes, and be bound by different legal obligations with respect to how they authenticate Principals. The choices that an identity provider makes here will be driven in large part by the requirements of the service providers with which the identity provider has affiliated into a circle of trust. These requirements themselves will be determined by the nature of the service (that is, the sensitivity of any information exchanged, the associated financial value, the service providers risk tolerance, etc.) that the service provider will be providing to the Principal. Consequently, for anything other than trivial services, if the service provider is to place sufficient confidence in the authentication assertions it receives from an identity provider, it will be necessary for the service provider to know which technologies, protocols, and processes were used or followed for the original authentication mechanism on which the authentication assertion is based. Armed with this information and trusting the origin of the actual assertion, the service provider will be better able to make an informed entitlements decision regarding what services the subject of the authentication assertion should be allowed to access.

Authentication context is defined as the information additional to the authentication assertion itself that the service provider may require before it makes an entitlements decision.

3 Authentication Context

If a service provider is to rely on the authentication of a Principal by an identity provider, the service provider may require information additional to the authentication itself to allow it to put the authentication in a trust context. This information could include

- Initial user identification mechanisms (for example, face-to-face, online, shared secret)
- Mechanisms for minimizing compromise of a Principal's credentials (for example, credential renewal frequency, client-side key generation)
- Mechanisms for storing and protecting credentials (for example, smartcard, password rules)
- Authentication mechanism (for example, password, certificate-based SSL)

The variations and permutations in the examples above guarantee that not all authentication assertions are the same; a particular authentication assertion will be characterized by the values for each of these variables. A somewhat helpful model is to think of an authentication assertion as defined by its coordinates in a multidimensional space. This model is demonstrated in Figure 1 (where only three axes are shown).

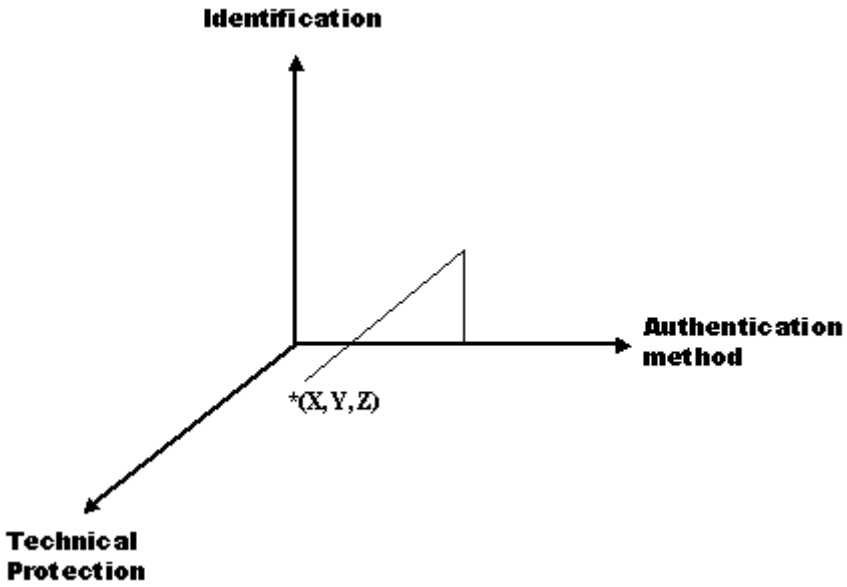

Figure 1: Authentication assertion as defined by its coordinates in multidimensional space

A particular authentication context statement will be characterized by its values along the different axes and consequently by its position in this space.

3.1 Authentication Context Classes

Liberty can simplify for service providers the task of assessing and comparing authentication assertions by defining particular authentication contexts that are representative of current technologies and practices among identity providers. For instance, a typical authentication context will be when a Principal uses a self-chosen password over a server-authenticated SSL session to authenticate to an identity provider. (This identity would have been issued when the Principal was originally identified after proving knowledge of some personal information, for example, a frequent flier account number.) Liberty should acknowledge the relevance of this authentication context, and remove from service providers the burden of parsing an XML document that captures this context, by identifying this authentication context as a Liberty *class* and by giving it a unique identifier so that service providers can recognize it and place an appropriate level of assurance on the associated authentication assertion.

A particular Liberty authentication context class will define a list of required characteristics of the processes, procedures, and mechanisms by which the identity provider verifies the Principal before issuing an identity, protects the secrets on which subsequent authentications are based, and the mechanisms used for this authentication. These characteristics can be categorized as

- **Identification** – Characteristics that describe the processes and mechanism the identity provider uses to initially create an association between a Principal and the identity (or name) by which the Principal will be known.
- **Physical Protection** – Characteristics that specify physical controls on the facility housing the identity provider's systems (for example, site location and construction, access controls).
- **Operational Protection –** Characteristics that describe procedural security controls employed by the identity provider (for example, security audits, records archival).

- **Technical Protection** – Characteristics that describe how the "secret" (the knowledge or possession of which allows the Principal to authenticate to the identity provider) is kept secure.
- **Authentication Method –** Characteristics that define the mechanisms by which the Principal authenticates to the identity provider (for example, a password versus a smartcard).

Rather than a class being a rigid collection of these characteristics, a class will define a set of conformant authentication context statements (for example, multiple and different authentication context statements will satisfy the requirements of a given class). The relationship between an authentication context class and particular authentication context statements is shown in Figure 2, where all the authentication context statements satisfy the requirements expressed by the class.

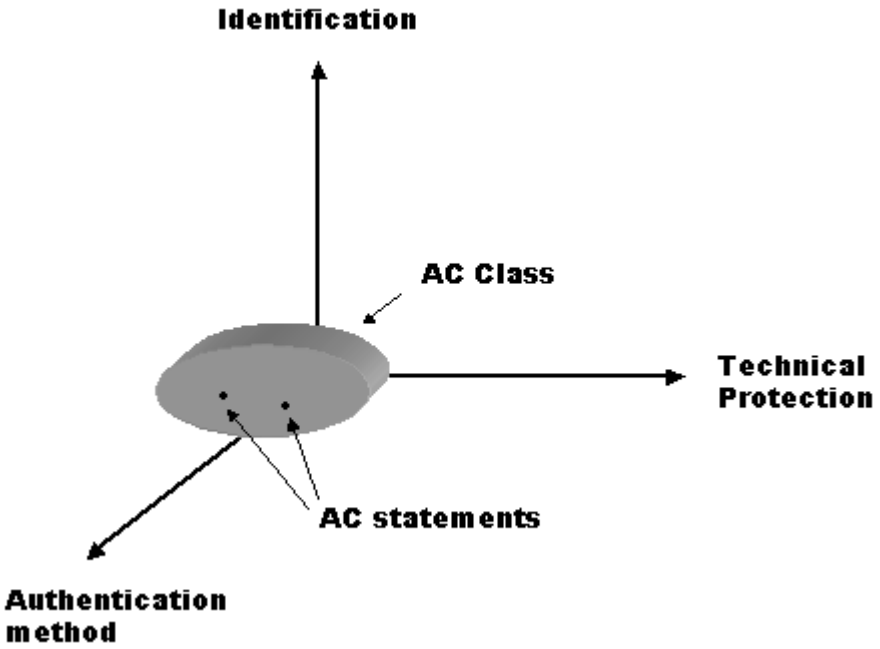

Figure 2: Relationship between authentication context class and statements

By introducing the additional layer of classes and by defining an initial list of representative and flexible classes, Liberty architecture

- Makes it easier for the identity provider and service provider to come to an agreement on what are acceptable authentication contexts by giving them a framework for discussion.
- Makes it easier for service providers to indicate their preferences when requesting a step-up authentication assertion from an identity provider.
- Simplifies for service providers the burden of processing authentication context statements by giving them the option of being satisfied by the associated class.
- Protects service providers from impact of new authentication technologies.
- Makes it easier for identity providers to publish their authentication capabilities, for example, through WSDL.

3.2 Authentication Quality

Authentication quality refers to the level of assurance that a service provider can place in an authentication assertion it receives from an identity provider. Authentication quality is motivated by two goals: An identity provider must be able to indicate to a service provider the level of confidence it has in an authentication assertion, and a service provider should be able to indicate its preferences for an authentication context without necessarily specifying the exact context characteristics. The fundamental concern with the concept of authentication quality is the difficulty for Liberty to make the necessary assessments of the classes to enable this flexibility.

3.2.1 Service Provider Request

To provide the desired flexibility without requiring Liberty to itself assess the quality of particular authentication classes, the service provider will be provided a flexible mechanism by which it can indicate its preferences for authentication context to the identity provider. The <lib:AuthnAndFedRequest> message will allow the service provider to request any of the following:

1. A match on a particular authentication context statement
2. A match within a specific authentication context class
3. A match or better on a particular authentication context class
4. A match within an ordered list (which is designated by the service provider) of authentication context classes

Option 1 will require that the identity provider and service provider have previously agreed on the details of a particular authentication context that either does not fall into one of the Liberty-defined authentication context classes or needs to be constrained more tightly.

Option 2 is expected to be the typical scenario.

For option 3, the decision as to what is better is left to the entity best qualified to make that determination, the identity provider. The service provider, trusting the identity provider's judgment, will accept the assertion it receives back because it will be confident the assertion meets (or exceeds) the provider's requirements.

Option 4 will give the service provider greater control over the authentication context classes to which the authentication assertions it receives conform. The identity provider is given no leeway in providing an authentication assertion conforming to a class not on the list.

If the service provider does not specify any of the above options in the <lib:AuthnAndFedRequest>, the identity provider will be free to provide an authentication context of its choosing.

3.2.2 Identity Provider Response

The authentication assertion that the identity provider returns to the service provider may indicate the authentication context class to which the authentication assertion conforms (if it does conform to any such authentication context class), which may or may not be the same as the class requested.

The returned authentication assertion will include a URI specifying the associated authentication context statement.

4 Previous work

The concept of authentication context has been addressed in other work.

4.1 PKI

An X.509 certificate is a signed assertion of identity just as a SAML authentication assertion is. Consequently it is not surprising that the issue of authentication context has been addressed within the PKI world. A number of different standards or proposals for capturing this sort of information have been written:

- **Certificate Practice Statement (CPS)** is a statement of the practices that a certification authority employs in issuing certificates. A certificate practice statement may take the form of a declaration by the certification authority of the details of its trustworthy systems and the practices it employs in support of its issuance of certificates.
- **Certificate Policy** is a named set of rules that indicates the applicability of a certificate to a particular community and/or class of application. For example, a certificate policy might indicate that a particular type of certificate is appropriate for the authentication of participants in a business-to-business transaction within a given price range. The fundamental difference between the certificate practice statement and the certificate policy is that the former is "owned" by the issuing certification authority and the latter by the entities who will use the issued certificates. Certificate users define certificate policies, and certification authorities (with different certificate practice statements) attest that a particular certificate is appropriate for that certificate policy. (See [RFC2527].)
- **PKI Disclosure Statement** is a supplementary instrument that discloses critical information about the policies and practices of a certificate authority or PKI. A PKI disclosure statement is a vehicle for disclosing and emphasizing information normally covered in detail by associated certificate policy and/or certification practice statement documents. Consequently, a PKI disclosure statement is not intended to replace a certificate policy or practice statement. (See [PDS].)
- **Key Usage**, as defined in X.509, defines the intended use for a key contained in a certificate. These uses (or *values*) are digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment, keyAgreement, keyCertSign, CRLSign, encipherOnly, and decipherOnly.
- **Extended Key Usage**, as the name indicates, extends the possible uses for a key beyond the original nine, each use identified by an object identifier. Extended key usage is primarily used by the relying party. As part of its validation algorithm, a relying party will check for these values to determine whether a given certificate is appropriate for the application.

4.2 SAML

SAML provides limited support for the concept of authentication context, it defines an AuthenticationMethod attribute on the <AuthenticationStatement> element and an unconstrained (schema model of ANY) <Advice> element. The following listing is an example (where the relevant elements and attributes are bolded):

```
<?xml version="1.0"?>
<saml:Assertion>
<saml:AuthenticationStatement AuthenticationMethod=" urn:ietf:rfc:2246">
    <saml:Subject>
            <saml:NameIdentifier
```

```
       Format="http://www.oasis-open.org/committees/security/docs/draft-
sstc-core-28#X509SubjectName">cn=Joe User,dc=projectliberty,dc=org
            </saml:NameIdentifier>
          </saml:Subject>
</saml:AuthenticationStatement>
<saml:Advice>
<!–additional elements in separate namespace →
</saml:Advice>
</saml:Assertion>
```

Note: SAML also defines a <Condition> element, the purpose of which is somewhat complementary to the <Advice> element (see [SAMLCore]).

- <Conditions> [Optional]. Conditions that MUST be taken into account in assessing the validity of the assertion.
- <Advice> [Optional]. Additional information related to the assertion that assists processing in certain situations, but MAY be ignored by applications that do not support its use.

The intent seems to be that the <Conditions> element protects the issuing party, and the <Advice> element protects the relying party.

SAML also defines the <SubjectConfirmation> element as "a URI that identifies a protocol to be used to authenticate the subject" where authenticate refers to how the bearer of a SAML assertion proves that it is authorized to hold the assertion as opposed to how it convinced the identity provider to issue the assertion. As such, <SubjectConfirmation> is distinct from authentication context.

SAML identified a list of common authentication protocols as possible values for both the AuthenticationMethod attribute and the <SubjectConfirmation> element, including SAML Artifact, Holder of Key, Sender Vouches, Password, Kerberos, and SSL/TLS.

5 Liberty Authentication Context Mechanisms

5.1 Authentication Context Classes

The initial Liberty authentication context classes are listed in 5.1.1 through 5.1.10.

The classes are listed in alphabetical order, no ranking is implied.

Classes are identified by URIs with the initial stem:

```
http://www.projectliberty.org/schemas/authctx/classes
```

5.1.1 MobileContract

The MobileContract class is identified when a mobile Principal has an identity for which the identity provider has vouched.

5.1.1.1 Associated Liberty URI

http://www.projectliberty.org/schemas/authctx/classes/MobileContract

5.1.1.2 Class Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<annotation>
<documentation> http://www.projectliberty.org/schemas/authctx/classes/MobileContract
</documentation>
</annotation>
       <xs:element name="AuthenticationContextStatement">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1" ref="Identification"/>
                            <xs:element minOccurs="1" maxOccurs="1"
ref="TechnicalProtection"/>
                            <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticationMethod"/>
                            <xs:element minOccurs="1" maxOccurs="1"
ref="OperationalProtection"/>
                            <xs:element minOccurs="1" maxOccurs="1"
ref="GoverningAgreements"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="AuthenticationMethod">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1" ref="Authenticator"/>
                            <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticatorTransportProtocol"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>

       <xs:element name="Authenticator">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1"
ref="SharedSecretChallengeResponse"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>

       <xs:element name="AuthenticatorTransportProtocol">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1" ref="MobileNetwork"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>

       <xs:element name="DeactivationCallCenter">
              <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
       </xs:element>

       <xs:element name="GoverningAgreementRef">
              <xs:complexType>
                     <xs:attribute name="ref"
fixed="http://SomeMobileforum.org/namespaces/authcontext/classes/Mobile-Class2.pdf"/>
              </xs:complexType>
       </xs:element>
```

```
      <xs:element name="GoverningAgreements">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element minOccurs="1" maxOccurs="1"
ref="GoverningAgreementRef"/>
                         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="Identification">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element minOccurs="1" maxOccurs="1"
ref="PhysicalVerification"/>
                         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
                  <xs:attribute name="nym" type="xs:string" use="required"/>
            </xs:complexType>
      </xs:element>
      <xs:element name="MobileAuthCard">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="MobileDevice">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="MobileNetwork">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="OperationalProtection">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element minOccurs="1" maxOccurs="1" ref="SecurityAudit"/>
                        <xs:element minOccurs="1" maxOccurs="1"
ref="DeactivationCallCenter"/>
                         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>

      <xs:element name="PhysicalVerification">
            <xs:complexType>
                  <xs:attribute name="credentialLevel" type="xs:string" use="required"/>
            </xs:complexType>
      </xs:element>

      <xs:element name="SecurityAudit">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element minOccurs="1" maxOccurs="1" ref="SwitchAudit"/>
                         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="SharedKeyProtection">
            <xs:complexType>
                  <xs:choice>
                        <xs:element minOccurs="1" maxOccurs="1" ref="MobileAuthCard"/>
                        <xs:element minOccurs="1" maxOccurs="1" ref="MobileDevice"/>
                  </xs:choice>
            </xs:complexType>
      </xs:element>
      <xs:element name="SharedSecretChallengeResponse">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
```

```
      <xs:element name="SwitchAudit">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="TechnicalProtection">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element minOccurs="1" maxOccurs="1"
ref="SharedKeyProtection"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
</xs:schema>
```

5.1.2 MobileDigitalID

The MobileDigitalID class is identified by detailed and verified registration procedures, users' consent to sign and authorize transactions, and DigitalID-based authentication.

5.1.2.1 Associated Liberty URI

http://www.projectliberty.org/schemas/authctx/classes/MobileDigitalID

5.1.2.2 Class Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<annotation>
<documentation> http://www.projectliberty.org/schemas/authctx/classes/MobileDigitalID
</documentation>
</annotation>

      <xs:element name="AuthenticationContextStatement">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="Identification"/>
                        <xs:element ref="TechnicalProtection"/>
                        <xs:element ref="AuthenticationMethod"/>
                        <xs:element ref="OperationalProtection"/>
                        <xs:element ref="GoverningAgreements"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="AuthenticationMethod">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="Authenticator"/>
                        <xs:element ref="AuthenticatorTransportProtocol"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="Authenticator">
            <xs:complexType>
                  <xs:choice>
                        <xs:element ref="Dig-sig"/>
                        <xs:element ref="ZeroKnowledge"/>
                  </xs:choice>
            </xs:complexType>
      </xs:element>
```

```
      <xs:element name="AuthenticatorTransportProtocol">
            <xs:complexType>
                  <xs:choice>
                        <xs:element ref="MobileNetwork"/>
                        <xs:element ref="SSL"/>
                        <xs:element ref="WTLS"/>
                        <xs:element ref="IPSec"/>
                  </xs:choice>
            </xs:complexType>
      </xs:element>
      <xs:element name="DeactivationCallCenter">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="Dig-sig">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="GoverningAgreementRef">
            <xs:complexType>
                  <xs:attribute name="ref"
fixed="http://SomeMobileforum.org/namespaces/authcontext/classes/Mobile-Class3.pdf"/>
            </xs:complexType>
      </xs:element>
      <xs:element name="GoverningAgreements">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="GoverningAgreementRef"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="IPSec">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="Identification">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="PhysicalVerification"/>
                        <xs:element ref="WrittenConsent"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
                  <xs:attribute name="nym" type="xs:string" use="required"/>
            </xs:complexType>
      </xs:element>
      <xs:element name="KeyStorage">
            <xs:complexType>
                  <xs:attribute name="medium" type="xs:string" use="required"/>
            </xs:complexType>
      </xs:element>
      <xs:element name="MobileNetwork">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="OperationalProtection">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="SecurityAudit"/>
                        <xs:element ref="DeactivationCallCenter"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="PhysicalVerification">
            <xs:complexType>
                  <xs:attribute name="credentialLevel" type="xs:string" use="required"/>
            </xs:complexType>
```

```
      </xs:element>
      <xs:element name="PrivateKeyProtection">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="KeyStorage"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="SSL">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="SecurityAudit">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="SwitchAudit"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="SwitchAudit">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="TechnicalProtection">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="PrivateKeyProtection"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="WTLS">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="WrittenConsent">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="ZeroKnowledge">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
</xs:schema>
```

5.1.3 MobileUnregistered

The MobileUnregistered class is identified when the real identity of a mobile Principal has not been strongly verified.

5.1.3.1 Associated Liberty URI

http://www.projectliberty.org/schemas/authctx/classes/MobileUnregistered

5.1.3.2 Class Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<annotation>
```

```
<documentation> http://www.projectliberty.org/schemas/authctx/classes/MobileUnregistered
</documentation>
</annotation>

      <xs:element name="AuthenticationContextStatement">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="TechnicalProtection"/>
                        <xs:element ref="AuthenticationMethod"/>
                        <xs:element ref="OperationalProtection"/>
                        <xs:element ref="GoverningAgreements"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="AuthenticationMethod">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="Authenticator"/>
                        <xs:element ref="AuthenticatorTransportProtocol"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="Authenticator">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="SharedSecretChallengeResponse"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="AuthenticatorTransportProtocol">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="MobileNetwork"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="DeactivationCallCenter">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="GoverningAgreementRef">
            <xs:complexType>
                  <xs:attribute name="ref"
fixed="http://SomeMobileforum.org/namespaces/authcontext/classes/Mobile-Class1.pdf"/>
            </xs:complexType>
      </xs:element>
      <xs:element name="GoverningAgreements">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="GoverningAgreementRef"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="MobileAuthCard">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="MobileDevice">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="MobileNetwork">
```

```
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="OperationalProtection">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="SecurityAudit"/>
                        <xs:element ref="DeactivationCallCenter"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="SecurityAudit">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="SwitchAudit"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="SharedKeyProtection">
            <xs:complexType>
                  <xs:choice>
                        <xs:element ref="MobileAuthCard"/>
                        <xs:element ref="MobileDevice"/>
                  </xs:choice>
            </xs:complexType>
      </xs:element>
      <xs:element name="SharedSecretChallengeResponse">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="SwitchAudit">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="TechnicalProtection">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="SharedKeyProtection"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
</xs:schema>
```

5.1.4 Password

The Password class is identified when a Principal authenticates to an identity provider through the presentation of a password over an unprotected HTTP session.

5.1.4.1 Associated Liberty URI

http://www.projectliberty.org/schemas/authctx/classes/Password

5.1.4.2 Class Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<annotation>
<documentation> http://www.projectliberty.org/schemas/authctx/classes/Password
</documentation>
```

```
</annotation>

      <xs:element name="AuthenticationContextStatement">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="AuthenticationMethod"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="AuthenticationMethod">
            <xs:complexType>
                  <xs:all>
                        <xs:element ref="PrincipalAuthenticationMechanism"/>
                        <xs:element ref="AuthenticatorTransportProtocol"/>
                  </xs:all>
            </xs:complexType>
      </xs:element>
      <xs:element name="AuthenticatorTransportProtocol">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="HTTP"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="HTTP">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="Length">
            <xs:complexType>
                  <xs:attribute name="min" fixed="3"/>
            </xs:complexType>
      </xs:element>
      <xs:element name="Password">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="Length"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="PrincipalAuthenticationMechanism">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="Password"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
</xs:schema>
```

5.1.5 Password- ProtectedTransport

The Password-ProtectedTransport class is identified when a Principal authenticates to an identity provider through the presentation of a password over an SSL-protected session.

5.1.5.1 Associated Liberty URI

http://www.projectliberty.org/schemas/authctx/classes/Password-ProtectedTransport

5.1.5.2 Class Schema

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<annotation>
<documentation> http://www.projectliberty.org/schemas/authctx/classes/Password-
ProtectedTransport </documentation>
</annotation>

      <xs:element name="AuthenticationContextStatement">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="AuthenticationMethod"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="AuthenticationMethod">
            <xs:complexType>
                  <xs:all>
                        <xs:element ref="PrincipalAuthenticationMechanism"/>
                        <xs:element ref="AuthenticatorTransportProtocol"/>
                  </xs:all>
            </xs:complexType>
      </xs:element>
      <xs:element name="AuthenticatorTransportProtocol">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="SSL"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="Length">
            <xs:complexType>
                  <xs:attribute name="min" fixed="3"/>
            </xs:complexType>
      </xs:element>
      <xs:element name="Password">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="Length"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="PrincipalAuthenticationMechanism">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element ref="Password"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="SSL">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
</xs:schema>
```

5.1.6 Previous-Session

The Previous-Session class is identified when a Principal had authenticated to an identity provider at some point in the past using any authentication context supported by that identity provider. Consequently, a subsequent authentication event that the identity provider will assert to the service provider may be significantly separated in time from the Principal's current resource access request.

The context for the previously authenticated session is explicitly not included in this context class because the user has not authenticated during this session, and so the mechanism that the user employed to authenticate in a previous session should not be used as part of a decision on whether to *now* allow access to a resource.

5.1.6.1 Associated Liberty URI

http://www.projectliberty.org/schemas/authctx/classes/Previous-Session

5.1.6.2 Class Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<annotation>
<documentation> http://www.projectliberty.org/schemas/authctx/classes/Previous-Session
</documentation>
</annotation>

       <xs:element name="AuthenticationContextStatement">
              <xs:complexType>
                     <xs:sequence>
                           <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticationMethod"/>
                            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>

       <xs:element name="AuthenticationMethod">
              <xs:complexType>
                     <xs:sequence>
                           <xs:element ref="Authenticator" minOccurs="0" maxOccurs="1"/>
                             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>

       <xs:element name="Authenticator">
              <xs:complexType>
                     <xs:sequence>
                           <xs:element minOccurs="1" maxOccurs="1" ref="PreviousSession"/>
                             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>

       <xs:element name="PreviousSession">
              <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
       </xs:element>

</xs:schema>
```

5.1.7 Smartcard

The Smartcard class is identified when a Principal authenticates to an identity provider using a smartcard.

5.1.7.1 Associated Liberty URI

http://www.projectliberty.org/schemas/authctx/classes/Smartcard

5.1.7.2 Class Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<annotation>
<documentation> http://www.projectliberty.org/schemas/authctx/classes/Smartcard
</documentation>
</annotation>

       <xs:element name="AuthenticationContextStatement">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticationMethod"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="AuthenticationMethod">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1"
ref="PrincipalAuthenticationMechanism"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="PrincipalAuthenticationMechanism">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1" ref="Smartcard"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="Smartcard">
              <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
       </xs:element>
</xs:schema>
```

5.1.8 Smartcard-PKI

The Smartcard-PKI class is identified when a Principal authenticates to an identity provider through a two-factor authentication mechanism using a smartcard with enclosed private key and a PIN.

5.1.8.1 Associated Liberty URI

http://www.projectliberty.org/schemas/authctx/classes/Smartcard-PKI

5.1.8.2 Class Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<annotation>
```

```
<documentation> http://www.projectliberty.org/schemas/authctx/classes/Smartcard-
PKI</documentation>
</annotation>

       <xs:element name="AuthenticationContextStatement">
              <xs:complexType>
                     <xs:sequence>
                           <xs:element minOccurs="1" maxOccurs="1"
ref="TechnicalProtection"/>
                           <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticationMethod"/>
                             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="AuthenticationMethod">
              <xs:complexType>
                     <xs:sequence>
                           <xs:element minOccurs="1" maxOccurs="1"
ref="PrincipalAuthenticationMechanism"/>
                           <xs:element minOccurs="1" maxOccurs="1" ref="Authenticator"/>
                           <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticatorTransportProtocol"/>
                             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="Authenticator">
              <xs:complexType>
                     <xs:sequence>
                           <xs:element minOccurs="1" maxOccurs="1" ref="Dig-sig"/>
                             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="AuthenticatorTransportProtocol">
              <xs:complexType>
                     <xs:sequence>
                           <xs:element minOccurs="1" maxOccurs="1" ref="SSL"/>
                             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="Dig-sig">
              <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
       </xs:element>
       <xs:element name="KeyActivation">
              <xs:complexType>
                     <xs:sequence>
                           <xs:element minOccurs="1" maxOccurs="1" ref="Password"/>
                             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="Length">
              <xs:complexType>
                     <xs:attribute name="min" type="xs:byte" use="required"/>
              </xs:complexType>
       </xs:element>
       <xs:element name="Password">
              <xs:complexType>
                     <xs:sequence>
                           <xs:element minOccurs="1" maxOccurs="1" ref="Length"/>
                             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
```

```
      <xs:element name="PrincipalAuthenticationMechanism">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element minOccurs="1" maxOccurs="1" ref="Smartcard"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="PrivateKeyProtection">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element minOccurs="1" maxOccurs="1" ref="KeyActivation"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
      <xs:element name="SSL">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="Smartcard">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
      </xs:element>
      <xs:element name="TechnicalProtection">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element minOccurs="1" maxOccurs="1"
ref="PrivateKeyProtection"/>
                          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
      </xs:element>
</xs:schema>
```

5.1.9 Software-PKI

Th Software-PKI class is identified when a Principal uses an X.509 certificate stored in software to authenticate to the identity provider over an SSL protected session.

5.1.9.1 Associated Liberty URI

http://www.projectliberty.org/schemas/authctx/classes/Software-PKI

5.1.9.2 Class Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<annotation>
<documentation> http://www.projectliberty.org/schemas/authctx/classes/Software-PKI
</documentation>
</annotation>

      <xs:element name="AuthenticationContextStatement">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticationMethod"/>
```

```
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="AuthenticationMethod">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1"
ref="PrincipalAuthenticationMechanism"/>
                            <xs:element minOccurs="1" maxOccurs="1" ref="Authenticator"/>
                            <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticatorTransportProtocol"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="Authenticator">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1" ref="Dig-sig"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="AuthenticatorTransportProtocol">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1" ref="SSL"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="Dig-sig">
              <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
       </xs:element>
       <xs:element name="Password">
              <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
       </xs:element>
       <xs:element name="PrincipalAuthenticationMechanism">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1" ref="Password"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="SSL">
              <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
       </xs:element>
</xs:schema>
```

5.1.10 Time-Sync-Token

The Time-Sync-Token class is identified when a Principal authenticates through a time synchronization token.

5.1.10.1 Associated Liberty URI

http://www.projectliberty.org/schemas/authctx/classes/Time-Sync-Token

5.1.10.2 Class Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

<annotation>
<documentation> http://www.projectliberty.org/schemas/authctx/classes/Time-Sync-Token
</documentation>
</annotation>

       <xs:element name="AuthenticationContextStatement">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticationMethod"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="AuthenticationMethod">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1"
ref="PrincipalAuthenticationMechanism"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="Generation">
              <xs:complexType>
                     <xs:attribute name="mechanism" fixed="principalchosen" />
              </xs:complexType>
       </xs:element>

       <xs:element name="PrincipalAuthenticationMechanism">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1" ref="Token"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
       <xs:element name="TimeSyncToken">
              <xs:complexType>
                     <xs:attribute name="deviceType" fixed="hardware" />
                     <xs:attribute name="seedLength" fixed="64" />
                     <xs:attribute name="deviceInHand"  fixed="true" />
              </xs:complexType>
       </xs:element>
       <xs:element name="Token">
              <xs:complexType>
                     <xs:sequence>
                            <xs:element minOccurs="1" maxOccurs="1" ref="TimeSyncToken"/>
                              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
              </xs:complexType>
       </xs:element>
</xs:schema>
```

5.2 Authentication Context Schema

The relationship between authentication context statements, authentication context classes, and the authentication context XML schema is shown in Figure 3.

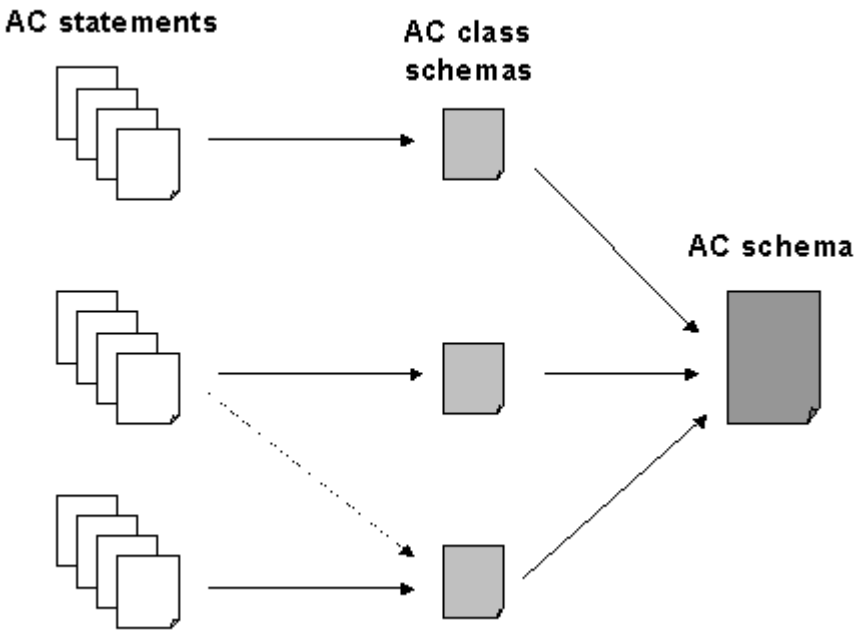

Figure 3: Relationship between authentication context statements, classes, and XML schema

Authentication context statements may conform to authentication context classes, which are themselves logical subsets of the authentication context XML schema.

5.2.1 XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>

<schema targetNamespace="http://www.projectliberty.org/schemas/authctx/2002/05"
xmlns:xsd="http://www.w3.org/2001/XMLSchema "
xmlns:AC="http://www.projectliberty.org/schemas/authctx/2002/05"
xmlns="http://www.w3.org/2001/XMLSchema" version="1.0">

<annotation>
<documentation> http://www.projectliberty.org/schemas/authctx/2002/05/
 </documentation>
</annotation>

      <element name="AuthenticationContextStatement">
            <annotation>
                  <documentation>A claim made by an identity provider with respect to
the authentication context associated with an authentication assertion. </documentation>
            </annotation>
            <complexType>
                  <sequence>
                        <element ref="AC:Identification" minOccurs="0"/>
```

```
                        <element ref="AC:TechnicalProtection" minOccurs="0"/>
                        <element ref="AC:OperationalProtection" minOccurs="0"/>
                        <element ref="AC:AuthenticationMethod" minOccurs="0"/>
                        <element ref="AC:GoverningAgreements" minOccurs="0"/>
                        <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" />
                  </sequence>
                  <attribute name="ID" type="ID"/>
            </complexType>
      </element>

      <element name="Identification">
            <annotation>
                  <documentation>Refers to those characteristics that describe the
processes and mechanisms the
                  identity provider uses to initially create an association between a
Principal and the identity
                  (or name) by which the Principal will be known</documentation>
            </annotation>
            <complexType>
                  <sequence>
                        <element ref="AC:PhysicalVerification" minOccurs="0"/>
                        <element ref="AC:WrittenConsent" minOccurs="0"/>
                        <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" />
                  </sequence>
                  <attribute name="nym">
                        <annotation>
                              <documentation>This attribute indicates whether or not
the Identification mechanisms allow the
                        actions of the Principal to be linked to an actual end
user.</documentation>
                        </annotation>
                        <simpleType>
                              <restriction base="NMTOKEN">
                                    <enumeration value="anonymity"/>
                                    <enumeration value="verinymity"/>
                                    <enumeration value="pseudonymity"/>
                              </restriction>
                        </simpleType>
                  </attribute>
            </complexType>
      </element>
      <element name="PhysicalVerification">
            <annotation>
                  <documentation>This element indicates that identification has been
performed in a physical
                  face-to-face meeting with the principal and not in an online manner.
</documentation>
            </annotation>
            <complexType>
                  <attribute name="credentialLevel">
                        <simpleType>
                              <restriction base="NMTOKEN">
                                    <enumeration value="primary"/>
                                    <enumeration value="secondary"/>
                              </restriction>
                        </simpleType>
                  </attribute>
            </complexType>
      </element>
      <element name="WrittenConsent">
            <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
      </element>
      <element name="TechnicalProtection">
            <annotation>
```

```
                        <documentation>Refers to those characterstics that describe how the
'secret' (the knowledge or possession of which allows the Principal to authenticate to the
identity provider) is kept secure</documentation>
                </annotation>
                <complexType>
                        <sequence>
                                <element ref="AC:PrivateKeyProtection" minOccurs="0"/>
                                <element ref="AC:SharedKeyProtection" minOccurs="0"/>
                                <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" />
                        </sequence>
                </complexType>
        </element>
        <element name="SharedKeyProtection">
                <annotation>
                        <documentation>This element indicates the types and strengths of
facilities
                        of a UA used to protect a shared secret key from unauthorized access
and/or use.</documentation>
                </annotation>
                <complexType>
                        <choice minOccurs="0">
                                <element ref="AC:MobileDevice"/>
                                <element ref="AC:MobileAuthCard"/>
                        </choice>
                </complexType>
        </element>
        <element name="MobileDevice">
                <annotation>
                        <documentation>This element idicates that the shared secret key is
securely maintained in a mobile device
                (as opposed to being stored in a mobile authentication
card).</documentation>
                </annotation>
                <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
        </element>
        <element name="MobileAuthCard">
                <annotation>
                        <documentation>This element indicates that the shared secret key is
securely maintained in a mobile authentication card (e.g., a SIM card).</documentation>
                </annotation>
                <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
        </element>
        <element name="PrivateKeyProtection">
                <annotation>
                        <documentation>This element indicates the types and strengths of
facilities
                        of a UA used to protect a private key from unauthorized access and/or
use.</documentation>
                </annotation>
                <complexType>
                        <sequence>
                                <element ref="AC:KeyActivation" minOccurs="0"/>
                                <element ref="AC:KeyStorage" minOccurs="0"/>
                                <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" />
                        </sequence>
                </complexType>
        </element>
        <element name="KeyActivation">
                <annotation>
                        <documentation>The actions that must be performed before the private
key can be used. </documentation>
                </annotation>
                <complexType>
                        <choice>
```

```
                        <element ref="AC:Password"/>
                  </choice>
            </complexType>
      </element>
      <element name="KeyStorage">
            <annotation>
                  <documentation>In which medium is the private key stored.

                  memory - the private key is stored in memory.

                  smartcard - the private key is stored in a smartcard and may not be
read from that smartcard unless authorized.

                  token - the private key is stored in a hardware token and may not be
read from that token unless authorized.

                  MobileAuthCard - the private key is stored in a mobile authentication
card (e.g., SIM card) and may not be read from that token unless authorized.
                  </documentation>
            </annotation>
            <complexType>
                  <attribute name="medium" use="required">
                        <simpleType>
                              <restriction base="NMTOKEN">
                                    <enumeration value="memory"/>
                                    <enumeration value="smartcard"/>
                                    <enumeration value="token"/>
                                    <enumeration value="MobileAuthCard"/>
                              </restriction>
                        </simpleType>
                  </attribute>
            </complexType>
      </element>
      <element name="Password">
            <annotation>
                  <documentation>This element indicates that a password (or PIN or
passphrase) has been used to authenticate the Principal or
                  to gain access to some resource (for example, to gain access to the
private key).</documentation>
            </annotation>
            <complexType>
                  <sequence>
                        <element ref="AC:Length" minOccurs="0"/>
                        <element ref="AC:Generation" minOccurs="0"/>
                        <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" />
                  </sequence>
            </complexType>
      </element>
      <element name="Token">
            <annotation>
                  <documentation>This element indicates that a hardware or software
token is
                  used as a method of identifying the Principal.</documentation>
            </annotation>
            <complexType>
                        <sequence>
                              <element ref="AC:TimeSyncToken"/>
                              <any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" />
                        </sequence>
            </complexType>
      </element>
      <element name="TimeSyncToken">
            <annotation>
                  <documentation>This element indicates that a time synchronization
                  token is used to identify the Principal.
```

```
                    hardware - the time synchonization token has been implemented in
hardware.

                    software - the time synchronization token has been implemented in
software.

                    SeedLength - the length, in bits, of the random seed used in the time
synchronization token.
                    </documentation>
             </annotation>
             <complexType>
                    <attribute name="DeviceType" use="required">
                           <simpleType>
                                  <restriction base="NMTOKEN">
                                         <enumeration value="hardware"/>
                                         <enumeration value="software"/>
                                  </restriction>
                           </simpleType>
                    </attribute>
                    <attribute name="SeedLength" type="integer" use="required"/>
                    <attribute name="DeviceInHand" use="required">
                           <simpleType>
                                  <restriction base="NMTOKEN">
                                         <enumeration value="true"/>
                                         <enumeration value="false"/>
                                  </restriction>
                           </simpleType>
                    </attribute>
             </complexType>
      </element>
      <element name="Smartcard">
             <annotation>
                    <documentation>This element indicates that a smartcard is used to
identity the Principal.</documentation>
             </annotation>
             <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
      </element>
      <element name="Length">
             <annotation>
                    <documentation>This element indicates the minimum and/or maximum ASCII
                    length of the password which is enforced (by the UA or the IdP).  In
other words,
                    this is the minimum and/or maximum number of ASCII characters required
to represent a valid password.

                    min - the minimum number of ASCII characters required in a valid
password, as enforced by the UA or the IdP.

                    max - the maximum number of ASCII characters required in a valid
password, as enforced by the UA or the IdP.
                    </documentation>
             </annotation>
             <complexType>
                    <attribute name="min" type="integer" use="required"/>
                    <attribute name="max" type="integer" use="optional"/>
             </complexType>
      </element>
      <element name="Generation">
             <annotation>
                    <documentation>Indicates whether the password was chosen by the
Principal or auto-supplied by the identity provider.

                    principalchosen - the Principal is allowed to choose the value of the
password.  This is true even if the initial password is chosen at
                    random by the UA or the IdP and the Principal is then free to change
the password.
```

```
                    automatic - the password is chosen by the UA or the IdP to be
cryptographically strong in some sense, or to satisfy certain
                    password rules, and that the Principal is not free to change it or to
choose a new password.

                    </documentation>
             </annotation>
             <complexType>
                    <attribute name="mechanism" use="required">
                           <simpleType>
                                  <restriction base="NMTOKEN">
                                         <enumeration value="principalchosen"/>
                                         <enumeration value="automatic"/>
                                  </restriction>
                           </simpleType>
                    </attribute>
             </complexType>
      </element>
      <element name="AuthenticationMethod">
             <annotation>
                    <documentation>Refers to those characteristics that define the
mechanisms by which the Principal authenticates to the identity provider.</documentation>
             </annotation>
             <complexType>
                    <sequence>
                           <element ref="AC:PrincipalAuthenticationMechanism"/>
                           <element ref="AC:Authenticator" minOccurs="0"/>
                           <element ref="AC:AuthenticatorTransportProtocol"/>
                           <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" />
                    </sequence>
             </complexType>
      </element>
      <element name="PrincipalAuthenticationMechanism">
             <annotation>
                    <documentation>The method that a Principal employs to perform
authentication to local system components.</documentation>
             </annotation>
             <complexType>
                    <choice minOccurs="0" maxOccurs="unbounded">
                           <element ref="AC:Password"/>
                           <element ref="AC:Token"/>
                           <element ref="AC:Smartcard"/>
                    </choice>
             </complexType>
      </element>
      <element name="Authenticator">
             <annotation>
                    <documentation>The method applied to validate a principal's
authentication across a network </documentation>
             </annotation>
             <complexType>
                    <choice minOccurs="0" maxOccurs="unbounded">
                           <element ref="AC:PreviousSession"/>
                           <element ref="AC:Dig-sig"/>
                           <element ref="AC:ZeroKnowledge"/>
                           <element ref="AC:SharedSecretChallengeResponse"/>
                    </choice>
             </complexType>
      </element>
      <element name="PreviousSession">
             <annotation>
                    <documentation>Indicates that the Principal has been strongly
authenticated in a previous session during which
                    the IdP has set a cookie in the UA.  During the present session the
Principal has only been authenticated by
                    the UA returning the cookie to the IdP.</documentation>
             </annotation>
```

```
            <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
      </element>
      <element name="ZeroKnowledge">
            <annotation>
                  <documentation>This element indicates that the Principal has been
authenticated by a zero knowledge
                  technique as specified in ISO/IEC 9798-5.</documentation>
            </annotation>
            <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
      </element>
      <element name="SharedSecretChallengeResponse">
            <annotation>
                  <documentation>This element indicates that the Principal has been
authenticated by a challenge-response
                  protocol utilizing shared secret keys and symmetric
cryptography.</documentation>
            </annotation>
            <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
      </element>
      <element name="Dig-sig">
            <annotation>
                  <documentation>This element indicates that the Principal has been
authenticated by a mechanism which involves the Principal
                  computing a digital signature over at least challenge data provided by
the IdP.</documentation>
            </annotation>
            <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
      </element>
      <element name="AuthenticatorTransportProtocol">
            <annotation>
                  <documentation>The protocol across which Authenticator information is
transferred to an identity provider verifier.</documentation>
            </annotation>
            <complexType>
                  <choice minOccurs="0" maxOccurs="unbounded">
                        <element ref="AC:HTTP"/>
                        <element ref="AC:SSL"/>
                        <element ref="AC:MobileNetwork"/>
                        <element ref="AC:WTLS"/>
                        <element ref="AC:IPSec"/>
                  </choice>
            </complexType>
      </element>
      <element name="HTTP">
            <annotation>
                  <documentation>This element indicates that the Authenticator has been
transmitted
                  using bare HTTP utilizing no additional security
protocols.</documentation>
            </annotation>
            <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
      </element>
      <element name="IPSec">
            <annotation>
                  <documentation>This element indicates that the Authenticator has been
transmitted
                  using a transport mechanism protected by an IPSEC
session.</documentation>
            </annotation>
            <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
      </element>
      <element name="WTLS">
```

```
            <annotation>
                  <documentation>This element indicates that the Authenticator has been
transmitted
                  using a transport mechanism protected by a WTLS
session.</documentation>
            </annotation>
            <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
      </element>
      <element name="MobileNetwork">
            <annotation>
                  <documentation>This element indicates that the Authenticator has been
transmitted
                  solely across a mobile network using no additional security
mechanism.</documentation>
            </annotation>
            <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
      </element>
      <element name="SSL">
            <annotation>
                  <documentation>This element indicates that the Authenticator has been
transmitted
                  using a transport mechnanism protected by an SSL or TLS
session.</documentation>
            </annotation>
            <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
      </element>
      <element name="OperationalProtection">
            <annotation>
                  <documentation>Refers to those characteristics that describe
procedural security controls employed by the identity provider.</documentation>
            </annotation>
            <complexType>
                  <sequence>
                        <element ref="AC:SecurityAudit" minOccurs="0"/>
                        <element ref="AC:DeactivationCallCenter" minOccurs="0"/>
                        <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" />
                  </sequence>
            </complexType>
      </element>
      <element name="SecurityAudit">
            <complexType>
                  <sequence>
                        <element ref="AC:SwitchAudit" minOccurs="0"/>
                        <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" />
                  </sequence>
            </complexType>
      </element>
      <element name="SwitchAudit">
            <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
      </element>
      <element name="DeactivationCallCenter">
            <complexType><sequence><any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
      </element>
      <element name="GoverningAgreements">
            <annotation>
                  <documentation>Provides a mechanism for linking to external (likely
human readable) documents in which the identity provider can define business level
authentication context, e.g. liability constraints, contrctual
obligations.</documentation>
            </annotation>
            <complexType>
```

```
                    <sequence>
                          <element ref="AC:GoverningAgreementRef"/>
                    </sequence>
              </complexType>
        </element>
        <element name="GoverningAgreementRef">
              <complexType>
                    <attribute name="governingAgreementRef" type="anyURI" use="required"/>
              </complexType>
        </element>
</schema>
```

6 References

[LibertyGloss] Mauldin, H., "Liberty Glossary," https://66.34.4.93/members/technology expert group/draft-liberty-tech-glossary-01.doc, April 2002.

[LibertyProtSchema] Beatty, J., "Liberty Protocols and Schemas Specification," https://66.34.4.93/members/technology 20expert group/architecture/draft-liberty-architecture-protocols-schemas-04.doc, April 2002.

[PDS] Santesson S. et al, "Internet X.509 Public Key Infrastructure PKI Disclosure Statement," http://www.verisign.com/repository/pds.txt.

[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.

[RFC2527] Chokhani S. et al, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework," http://www.ietf.org/rfc/rfc2527.txt?number=2527.

[SAMLBind] P. Mishra et al., "Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML)," http://www.oasis-open.org/committees/security/docs/draft-sstc-bindings-model-11.pdf, OASIS, January 2002.

[SAMLCore] Hallam-Baker, P., et al., "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)," http://www.oasis-open.org/committees/security/docs/draft-sstc-core-31.pdf, OASIS, April 2002.

[Schema1] H. S. Thompson et al., "XML Schema Part 1: Structures," http://www.w3.org/TR/xmlschema-1/, World Wide Web Consortium Recommendation, May 2001.