Liberty ID-FF Authentication Context
Specification
Version: 1.2-errata-v1.0

Editors:
Paul Madsen, Entrust, Inc.

Contributors:
Robert Aarts, Nokia Corporation
Nick Bone, Vodafone Group Plc
Scott Cantor, Internet2, The Ohio State University
Bronislav Kavsan, RSA Security Inc.
John Kemp, IEEE-ISTO
Michael Meyerstein, Vodafone Group Plc
Xavier Serret, Gemplus SA

Abstract:
If a service provider is to rely on the authentication of a Principal by an identity provider (or more generally of another provider by an authentication authority), the service provider may require information additional to the assertion itself in order to assess the level of confidence they can place in that assertion. This specification defines an XML Schema for the creation of Authentication Context statements - XML documents that allow the authentication authority to provide to the service provider this additional information. Additionally, this specification defines a number of Authentication Context classes; categories into which many Authentication Context statements will fall, thereby simplifying their interpretation.

Filename: draft-liberty-authentication-context-1.2-errata-v1.0.pdf
Notice

This document has been prepared by Sponsors of the Liberty Alliance. Permission is hereby granted to use the
document solely for the purpose of implementing the Specification. No rights are granted to prepare derivative works
of this Specification. Entities seeking permission to reproduce portions of this document for other uses must contact
the Liberty Alliance to determine whether an appropriate license for such use is available.

Implementation of certain elements of this document may require licenses under third party intellectual property
rights, including without limitation, patent rights. The Sponsors of and any other contributors to the Specification are
not, and shall not be held responsible in any manner for identifying or failing to identify any or all such third party
intellectual property rights. This Specification is provided "AS IS", and no participant in the Liberty Alliance
makes any warranty of any kind, express or implied, including any implied warranties of merchantability,
non-infringement of third party intellectual property rights, and fitness for a particular purpose. Implementors
of this Specification are advised to review the Liberty Alliance Project’s website (http://www.projectliberty.org/) for
information concerning any Necessary Claims Disclosure Notices that have been received by the Liberty Alliance
Management Board.

Copyright © 2004 ActivCard; America Online, Inc.; American Express Travel Related Services; Axalto; Bank of
America Corporation; Bell Canada; Cingular Wireless; Cisco Systems, Inc.; Communicator, Inc.; Deloitte & Touche
LLP; Earthlink, Inc.; Electronic Data Systems, Inc.; Entrust, Inc.; Epok, Inc.; Ericsson; Fidelity Investments; France
Telecom; Gemplus; General Motors; Hewlett-Packard Company; i2 Technologies, Inc.; Internet2; Intuit Inc.;
MasterCard International; NEC Corporation; Netegrity, Inc.; NeuStar, Inc.; Nextel Communications; Nippon
Telesign and Telephone Corporation; Nokia Corporation; Novell, Inc.; NTT DoCoMo, Inc.; OneName Corporation;
Openwave Systems Inc.; Phaos Technology; Ping Identity Corporation; PricewaterhouseCoopers LLP; RegistryPro,
Inc.; RSA Security Inc; Sabre Holdings Corporation; SAP AG; SchlumbergerSema; Sigaba; SK Telecom; Sony
Corporation; Sun Microsystems, Inc.; Symlabs, Inc.; Trustgenix; United Airlines; VeriSign, Inc.; Visa International;
Vodafone Group Plc; Wave Systems. All rights reserved.

Liberty Alliance Project
Licensing Administrator
c/o IEEE-ISTO
445 Hoes Lane
Piscataway, NJ 08855-1331, USA
info@projectliberty.org
1. About this Document

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

1.1. Notation and Terminology

This section specifies the notations, namespaces and terminology used throughout this specification. This specification uses schema documents conforming to W3C XML Schema (see [Schema1]) and normative text to describe the syntax and semantics of XML-encoded messages.

1.1.1. Notational Conventions

Note: Phrases and numbers in brackets [ ] refer to other documents; details of these references can be found in References (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 document are to be interpreted as described in [RFC2119].

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.

Listings of XML schemas appear like this.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:liberty:ac:2003-08"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="urn:liberty:ac:2003-08">
    <!-- Add Stuff Here -->
</xs:schema>
```

1.1.2. Namespaces

The following namespaces are referred to in this document:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>ac</td>
<td>urn:liberty:ac:1.2</td>
</tr>
<tr>
<td>lib</td>
<td>urn:liberty:iff:1.2</td>
</tr>
<tr>
<td>xs</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
</tr>
<tr>
<td>xsi</td>
<td><a href="http://www.w3.org/2001/XMLSchema-instance">http://www.w3.org/2001/XMLSchema-instance</a></td>
</tr>
</tbody>
</table>

This specification uses the following typographical conventions in text: <Element>, <ns:ForeignElement>, Attribute, Datatype, OtherCode.
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 with respect to an authentication assertion.
3. Authentication Context

If a relying party is to rely on the authentication of another entity by an authentication authority, the relying party may require information additional to the authentication itself to allow it to put the authentication into a risk-management context. This information could include:

- What were the initial user identification mechanisms (for example, face-to-face, online, shared secret).
- What are the mechanisms for minimizing compromise of credentials (for example, credential renewal frequency, client-side key generation).
- What are the mechanisms for storing and protecting credentials (for example, smartcard, password rules).
- What was the authentication mechanism (for example, password, certificate-based SSL).

The variations and permutations in the characteristics listed above guarantee that not all authentication assertions will be the same with respect to the confidence that a relying party can place in it; a particular authentication assertion will be characterized by the values for each of these (and other) variables.
4. Authentication Context Statement

A Liberty authentication authority will deliver to a relying party the additional authentication context information in the form of an Authentication Context Statement, an XML document either inserted or referenced within the <AuthnResponse> message the authentication authority returns to the relying party.

4.1. Authentication Context Statement Data Model

A particular Liberty authentication context statement will capture the characteristics of the processes, procedures, and mechanisms by which the authentication verified the subject before issuing an identity, protects the secrets on which subsequent authentications are based, and the mechanisms used for this authentication. These characteristics are categorized in the Liberty Authentication Context schema as follows:

- Identification - Characteristics that describe the processes and mechanism the authentication authority uses to initially create an association between a subject and the identity (or name) by which the subject will be known.
- Technical Protection - Characteristics that describe how the "secret" (the knowledge or possession of which allows the subject to authenticate to the authentication authority) is kept secure.
- Operational Protection - Characteristics that describe procedural security controls employed by the authentication authority (for example, security audits, records archival).
- Authentication Method - Characteristics that define the mechanisms by which the subject of the issued assertion authenticates to the authentication authority (for example, a password versus a smartcard).
- Governing Agreements - Characteristics that describe the legal framework (e.g. liability constraints and contractual obligations) underlying the authentication event and/or its associated technical authentication infrastructure.

4.2. Authentication Context Statement Schema

This section lists the complete Authentication Context XML Schema.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:liberty:ac:2003-08"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="urn:liberty:ac:2003-08">
  <!-- added to get the Extension element -->
  <xs:include schemaLocation="liberty-utility-v1.0.xsd"/>
  <xs:annotation>
    <xs:documentation> ### IMPORTANT NOTICE ###
    The source code in this XSD file was excerpted verbatim from:
    Liberty Authentication Context Specification
    Version 1.2-errata-v1.0
    12 September 2004
    Copyright (c) 2004 Liberty Alliance participants, see
    http://www.projectliberty.org/specs/idff_copyrights.html
    </xs:documentation>
  </xs:annotation>
  <xs:element name="AuthenticationContextStatement" type="AuthenticationContextStatementType"/>
  <xs:annotation>
    <xs:documentation>
      A particular assertion on an identity provider's part with respect to the authentication
    </xs:documentation>
  </xs:annotation>
</xs:schema>
```
context associated with an authentication assertion.
</xs:documentation>
</xs:element>
</xs:element>  
<xs:element name="Identification" type="IdentificationType">
  <xs:annotation>
    <xs:documentation>
      Refers to those characteristics that describe the processes and mechanisms
      the Authentication Authority uses to initially create an association between a Principal
      and the identity (or name) by which the Principal will be known.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="PhysicalVerification">
  <xs:annotation>
    <xs: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.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="WrittenConsent">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="TechnicalProtection" type="TechnicalProtectionType">
  <xs:annotation>
    <xs:documentation>
      Refers to those characteristics that describe how the ‘secret’ (the knowledge or
      possession of which allows the Principal to authenticate to the Authentication
      Authority) is kept secure.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="SecretKeyProtection" type="SecretKeyProtectionType">
  <xs:annotation>
    <xs: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.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="PrivateKeyProtection" type="PrivateKeyProtectionType">
  <xs:annotation>
    <xs: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.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="KeyActivation" type="KeyActivationType">
  <xs:annotation>
    <xs:documentation>
The actions that must be performed before the private key can be used.
    </xs:documentation>
  </xs:annotation>
</xs:element>

Liberty Alliance Project
Whether or not the private key is shared with the certificate authority.

In which medium is the key stored.
memory - the key is stored in memory.
smartcard - the key is stored in a smartcard.
token - the key is stored in a hardware token.
MobileDevice - the key is stored in a mobile device.
MobileAuthCard - the key is stored in a mobile authentication card.

This element indicates that a password (or passphrase) has been used to authenticate the Principal to a remote system.

This element indicates that a Pin (Personal Identification Number) has been used to authenticate the Principal to some local system in order to activate a key.

This element indicates that a hardware or software token is used as a method of identifying the Principal.

This element indicates that a time synchronization token is used to identify the Principal.

This element indicates that a smartcard is used to identify the Principal.
<xs:complexType>
  <xs:element name="Length" type="LengthType">
    <xs:annotation>
      <xs: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.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="ActivationLimit" type="ActivationLimitType">
    <xs:annotation>
      <xs:documentation>
        This element indicates the length of time for which an PIN-based authentication is valid.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="Generation">
    <xs:annotation>
      <xs:documentation>
        Indicates whether the password was chosen by the Principal or auto-supplied by the Authentication Authority.
        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.
      </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:attribute name="mechanism" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="principalchosen"/>
            <xs:enumeration value="automatic"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
  <xs:element name="AuthenticationMethod" type="AuthenticationMethodType">
    <xs:annotation>
      <xs:documentation>
        Refers to those characteristics that define the mechanisms by which the Principal authenticates to the Authentication Authority.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="PrincipalAuthenticationMechanism" type="PrincipalAuthenticationMechanismType">
    <xs:annotation>
      <xs:documentation>
        The method that a Principal employs to perform authentication to local system components.
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:element name="Authenticator" type="AuthenticatorType">
    <xs:annotation>
      <xs:documentation>
        The method applied to validate a principal’s authentication across a network
      </xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:complexType>
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.

Rather like PreviousSession but using stronger security. A secret that was established in a previous session with the Authentication Authority has been cached by the local system and is now re-used (e.g. a Master Secret is used to derive new session keys in TLS, SSL, WTLS).

This element indicates that the Principal has been authenticated by a zero knowledge technique as specified in ISO/IEC 9798-5.

This element indicates that the Principal has been authenticated by a challenge-response protocol utilizing shared secret keys and symmetric cryptography.

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.
<xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="IPAddress">
  <xs:annotation>
    <xs:documentation>
    This element indicates that the Principal has been authenticated through connection from a particular IP address.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="AsymmetricDecryption">
  <xs:annotation>
    <xs:documentation>
The local system has a private key but it is used in decryption mode, rather than signature mode. For example, the Authentication Authority generates a secret and encrypts it using the local system’s public key; the local system then proves it has decrypted the secret.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="AsymmetricKeyAgreement">
  <xs:annotation>
    <xs:documentation>
The local system has a private key and uses it for shared secret key agreement with the Authentication Authority (e.g. via Diffie Heiman).
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="SharedSecretDynamicPlaintext">
  <xs:annotation>
    <xs:documentation>
The local system and Authentication Authority share a secret key. The local system uses this to encrypt a randomized string to pass to the Authentication Authority.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<xs:element name="AuthenticatorTransportProtocol" type="AuthenticatorTransportProtocolType">
  <xs:annotation>
    <xs:documentation>
The protocol across which Authenticator information is transferred to an Authentication Authority verifier.
    </xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
This element indicates that the Authenticator has been transmitted using bare HTTP utilizing no additional security protocols.

This element indicates that the Authenticator has been transmitted using a transport mechanism protected by an IPSEC session.

This element indicates that the Authenticator has been transmitted using a transport mechanism protected by a WTLS session.

This element indicates that the Authenticator has been transmitted solely across a mobile network using no additional security mechanism.
This element indicates that the Authenticator has been transmitted using a transport mechanism protected by an SSL or TLS session.

Refers to those characteristics that describe procedural security controls employed by the Authentication Authority.

Provides a mechanism for linking to external (likely human readable) documents in which additional business agreements, (e.g. liability constraints, obligations, etc) can be placed.

The Authority that originally authenticated the Principal.

This attribute indicates whether or not the Identification mechanisms allow the actions of the Principal to be linked to an actual end user.
<xs:simpleType>
  <xs:restriction base="xs:NMTOKEN">
    <xs:enumeration value="anonymity"/>
    <xs:enumeration value="verinymity"/>
    <xs:enumeration value="pseudonymity"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="GoverningAgreementsType">
  <xs:sequence>
    <xs:element ref="GoverningAgreementRef" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="GoverningAgreementRefType">
  <xs:attribute name="governingAgreementRef" type="xs:anyURI" use="required"/>
</xs:complexType>

<xs:complexType name="AuthenticatingAuthorityType">
  <xs:sequence>
    <xs:element ref="GoverningAgreements"/>
  </xs:sequence>
  <xs:attribute name="ID" type="xs:anyURI" use="required"/>
</xs:complexType>

<xs:complexType name="AuthenticatorTransportProtocolType">
  <xs:choice>
    <xs:element ref="HTTP"/>
    <xs:element ref="SSL"/>
    <xs:element ref="MobileNetworkNoEncryption"/>
    <xs:element ref="MobileNetworkRadioEncryption"/>
    <xs:element ref="MobileNetworkEndToEndEncryption"/>
    <xs:element ref="WTLS"/>
    <xs:element ref="IPSec"/>
    <xs:element ref="Extension" maxOccurs="unbounded"/>
  </xs:choice>
</xs:complexType>

<xs:complexType name="PrincipalAuthenticationMechanismType">
  <xs:choice>
    <xs:element ref="Password"/>
    <xs:element ref="Token"/>
    <xs:element ref="Smartcard"/>
    <xs:element ref="ActivationPin" maxOccurs="unbounded"/>
  </xs:choice>
</xs:complexType>

<xs:complexType name="AuthenticationMethodType">
  <xs:sequence>
    <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="Authenticator" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="AuthenticationContextStatementType">
  <xs:sequence>
    <xs:element ref="Identification" minOccurs="0" maxOccurs="0"/>
    <xs:element ref="TechnicalProtection" minOccurs="0" maxOccurs="0"/>
    <xs:element ref="OperationalProtection" minOccurs="0" maxOccurs="0"/>
    <xs:element ref="AuthenticationMethod" minOccurs="0" maxOccurs="0"/>
    <xs:element ref="GoverningAgreements" minOccurs="0" maxOccurs="0"/>
    <xs:element ref="AuthenticatingAuthority" minOccurs="0" maxOccurs="0"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="ID" type="xs:ID"/>
</xs:complexType>

<xs:complexType name="TechnicalProtectionType">
  <xs:choice>
    <xs:element ref="Extension" maxOccurs="unbounded"/>
  </xs:choice>
</xs:complexType>
<xs:element ref="PrivateKeyProtection" minOccurs="0"/>
<xs:element ref="SecretKeyProtection" minOccurs="0"/>
<xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
</xs:choice>
</xs:complexType>
<xs:complexType name="OperationalProtectionType">
<xs:sequence>
<xs:element ref="SecurityAudit" minOccurs="0"/>
<xs:element ref="DeactivationCallCenter" minOccurs="0"/>
<xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="AuthenticatorType">
<xs:choice>
<xs:element ref="PreviousSession"/>
<xs:element ref="ResumeSession"/>
<xs:element ref="DigSig"/>
<xs:element ref="Password"/>
<xs:element ref="ZeroKnowledge"/>
<xs:element ref="SharedSecretChallengeResponse"/>
<xs:element ref="SharedSecretDynamicPlaintext"/>
<xs:element ref="IPAddress"/>
<xs:element ref="AsymmetricDecryption"/>
<xs:element ref="AsymmetricKeyAgreement"/>
<xs:element ref="Extension" maxOccurs="unbounded"/>
</xs:choice>
</xs:complexType>
<xs:complexType name="KeyActivationType">
<xs:choice>
<xs:element ref="ActivationPin"/>
<xs:element ref="Extension" maxOccurs="unbounded"/>
</xs:choice>
</xs:complexType>
<xs:complexType name="KeySharingType">
<xs:attribute name="sharing" type="xs:boolean" use="required"/>
</xs:complexType>
<xs:complexType name="PrivateKeyProtectionType">
<xs:sequence>
<xs:element ref="KeyActivation" minOccurs="0"/>
<xs:element ref="KeyStorage" minOccurs="0"/>
<xs:element ref="KeySharing" minOccurs="0"/>
<xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="PasswordType">
<xs:sequence>
<xs:element ref="Length" minOccurs="0"/>
<xs:element ref="Alphabet" minOccurs="0"/>
<xs:element ref="Generation" minOccurs="0"/>
<xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:complexType name="ActivationPinType">
<xs:sequence>
<xs:element ref="Length" minOccurs="0"/>
<xs:element ref="Alphabet" minOccurs="0"/>
<xs:element ref="Generation" minOccurs="0"/>
<xs:element ref="ActivationLimit" minOccurs="0"/>
<xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:element name="Alphabet" type="AlphabetType"/>
<xs:complexType name="AlphabetType">
<xs:complexType name="TokenType">
  <xs:sequence>
    <xs:element ref="TimeSyncToken"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="TimeSyncTokenType">
  <xs:attribute name="DeviceType" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="hardware"/>
        <xs:enumeration value="software"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="SeedLength" type="xs:integer" use="required"/>
  <xs:attribute name="DeviceInHand" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="true"/>
        <xs:enumeration value="false"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

<xs:complexType name="ActivationLimitType">
  <xs:choice>
    <xs:element ref="ActivationLimitDuration"/>
    <xs:element ref="ActivationLimitUsages"/>
    <xs:element ref="ActivationLimitSession"/>
  </xs:choice>
</xs:complexType>

<xs:complexType name="ActivationLimitDurationType">
  <xs:attribute name="duration" type="xs:duration" use="required"/>
</xs:complexType>

<xs:element name="ActivationLimitDuration" type="ActivationLimitDurationType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Key Activation Limit is defined as a specific duration of time.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimitUsages" type="ActivationLimitUsagesType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Key Activation Limit is defined as a number of usages.
    </xs:documentation>
  </xs:annotation>
</xs:element>

<xs:element name="ActivationLimitSession" type="ActivationLimitSessionType">
  <xs:annotation>
    <xs:documentation>
      This element indicates that the Key Activation Limit is the session.
    </xs:documentation>
  </xs:annotation>
</xs:element>
<xs:complexType name="ActivationLimitUsagesType">
    <xs:attribute name="number" type="xs:integer" use="required"/>
</xs:complexType>

<xs:complexType name="ActivationLimitSessionType"/>

<xs:complexType name="LengthType">
    <xs:attribute name="min" type="xs:integer" use="required"/>
    <xs:attribute name="max" type="xs:integer" use="optional"/>
</xs:complexType>

<xs:complexType name="KeyStorageType">
    <xs:attribute name="medium" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="memory"/>
                <xs:enumeration value="smartcard"/>
                <xs:enumeration value="token"/>
                <xs:enumeration value="MobileDevice"/>
                <xs:enumeration value="MobileAuthCard"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>

<xs:complexType name="SecretKeyProtectionType">
    <xs:sequence>
        <xs:element ref="KeyActivation" minOccurs="0"/>
        <xs:element ref="KeyStorage" minOccurs="0"/>
        <xs:element ref="Extension" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="SecurityAuditType">
    <xs:sequence>
        <xs:element ref="SwitchAudit" minOccurs="0"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>

4.3. Authentication Context Statement Extensibility

The Authentication Context Statement schema has well-defined extensibility points through the <Extension> element. Authentication authorities can use this element to insert additional authentication context details for the SAML assertions they issue (assuming that the consuming relying party will be able to understand these extensions). These additional elements MUST be in a separate XML Namespace to that of the base Authentication Context Statement schema.

4.4. Authentication Context Statement Processing Rules

The processing rules for Authentication Context Statements are listed in [LibertyProtSchema].
5. Authentication Context Classes

The number of permutations of the different authentication context characteristics ensure that there are a theoretically infinite number of unique authentication contexts. The implication is that in theory any particular relying party would be expected to be able to parse arbitrary authentication context statements and, more importantly, to analyze the statement in order to assess the ‘quality’ of the associated authentication assertion. Making such an assessment is non-trivial.

Fortunately, an optimization is possible. While theoretically infinite, in practice many authentication contexts will fall into categories - these categories determined by industry practices and technology. For instance, many B2C Web browser authentication contexts will be (partially) defined by the Principal authenticating to the identity provider through the presentation of a password over an SSL protected session. In the enterprise world, certificate-based authentication will be more common. Of course, the full authentication context is not limited to the specifics of how the Principal authenticated. Nevertheless, the authentication method is often the most visible characteristic and as such, can serve as a useful classifier for a class of related authentication contexts.

Liberty normalizes this concept through the definition of a number of Authentication Context Classes. Each class will define a proper subset of the full set of authentication contexts. Classes have been chosen as representative of the current practices and technologies for authentication technologies. Classes will provide identity and service providers with a convenient shorthand when referring to authentication context issues. For instance, an identity provider, may include with the complete authentication context statement it provides to a service provider an assertion that the authentication context also belongs to one of the Liberty defined authentication classes. For some service providers, this assertion will be sufficient detail for it to be able to assign an appropriate level of confidence to the associated authentication assertion. Other service providers might prefer to examine the complete authentication context statement itself. Likewise, the ability to refer to an authentication context class rather than being required to list the complete details of a specific authentication content will simplify how the service provider expresses its desires and/or requirements to an identity provider.

5.1. Advantages of Authentication Context Classes

The introduction of the additional layer of classes and the definition of an initial list of representative and flexible classes are expected to:

• Make 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.

• Make it easier for service providers to indicate their preferences when requesting a step-up authentication assertion from an identity provider.

• Simplify for service providers the burden of processing authentication context statements by giving them the option of being satisfied by the associated class.

• Protect service providers from impact of new authentication technologies.

• Make it easier for identity providers to publish their authentication capabilities, for example, through WSDL.
5.2. Authentication Context Class Schemas

The initial Liberty authentication context classes are listed in the following sub-sections.

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

Classes are identified by URIs with the initial stem: http://www.projectliberty.org/schemas/authctx/classes

The class schemas are defined as extension by restriction of the base Authentication Context schema. Consequently, any XML instances that satisfy the schema constraints of one of the class schemas will also conform to the base Authentication Context schema.

5.2.1. Internet Protocol

The Internet Protocol class is identified when a Principal is authenticated through the use of a provided IP address.

5.2.1.1. Associated Liberty URI

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

5.2.1.2. Class Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:liberty:ac:2003-08"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="urn:liberty:ac:2003-08"
version="1.2-08" finalDefault="extension">
<xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
<xs:annotation>
<xs:documentation>
http://www.projectliberty.org/schemas/authctx/classes/InternetProtocol</xs:documentation>
</xs:annotation>
<xs:complexType name="InternetProtocolAuthenticatorType">
<xs:complexContent>
<xs:restriction base="AuthenticatorType">
<xs:choice>
<xs:element ref="IPAddress"/>
</xs:choice>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
</xs:schema>
```

5.2.2. InternetProtocolPassword

The Internet Protocol Password class is identified when a Principal is authenticated through the use of a provided IP address, in addition to username/password.

5.2.2.1. Associated Liberty URI

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

5.2.2.2. Class Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:liberty:ac:2003-08"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="urn:liberty:ac:2003-08""
5.2.3. MobileOneFactorUnregistered

Reflects no mobile customer registration procedures and an authentication of the mobile device without requiring explicit end-user interaction. Again, this context authenticates only the device and never the user, it is useful when services other than the mobile operator want to add a secure device authentication to their authentication process.

5.2.3.1. Associated Liberty URI

http://www.projectliberty.org/schemas/authctx/classes/MobileOneFactorUnregistered
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
  <xs:annotation>
    <xs:documentation>
      http://www.projectliberty.org/schemas/authctx/classes/MobileOneFactorUnregistered</xs:documentation>
  </xs:annotation>
  <xs:complexType name="MobileOneFactorUnregisteredAuthenticatorType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorType">
        <xs:choice>
          <xs:element ref="DigSig"/>
          <xs:element ref="ZeroKnowledge"/>
          <xs:element ref="SharedSecretChallengeResponse"/>
          <xs:element ref="AsymmetricDecryption"/>
          <xs:element ref="AsymmetricKeyAgreement"/>
          <xs:element ref="SharedSecretDynamicPlaintext"/>
        </xs:choice>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="MobileOneFactorUnregisteredAuthenticatorTransportProtocolType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorTransportProtocolType">
        <xs:choice>
          <xs:element ref="MobileNetworkNoEncryption"/>
          <xs:element ref="MobileNetworkRadioEncryption"/>
          <xs:element ref="MobileNetworkEndToEndEncryption"/>
          <xs:element ref="WTLS"/>
        </xs:choice>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="MobileOneFactorUnregisteredOperationalProtectionType">
    <xs:complexContent>
      <xs:restriction base="OperationalProtectionType">
        <xs:sequence>
          <xs:element ref="SecurityAudit"/>
          <xs:element ref="DeactivationCallCenter"/>  
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="MobileOneFactorUnregisteredTechnicalProtectionType">
    <xs:complexContent>
      <xs:restriction base="TechnicalProtectionType">
        <xs:choice>
          <xs:element ref="PrivateKeyProtection"/>
          <xs:element ref="SecretKeyProtection"/>
        </xs:choice>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="MobileOneFactorUnregisteredPrivateKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="PrivateKeyProtectionType">
        <xs:sequence>
          <xs:element ref="KeyStorage"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="MobileOneFactorUnregisteredSecretKeyProtectionType">
    <xs:complexContent>
      <xs:restriction base="SecretKeyProtectionType">
        <xs:sequence>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
<xs:sequence>
  <xs:element ref="KeyStorage"/>
  <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>

<xs:complexType>
  <xs:complexContent>
    <xs:restriction base="KeyStorageType">
      <xs:attribute name="medium" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="MobileDevice"/>
            <xs:enumeration value="MobileAuthCard"/>
            <xs:enumeration value="smartcard"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="MobileOneFactorUnregisteredSecurityAuditType">
  <xs:complexContent>
    <xs:restriction base="SecurityAuditType">
      <xs:sequence>
        <xs:element ref="SwitchAudit"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="MobileOneFactorUnregisteredIdentificationType">
  <xs:complexContent>
    <xs:restriction base="IdentificationType">
      <xs:attribute name="nym">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="anonymity"/>
            <xs:enumeration value="pseudonymity"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:schema targetNamespace="urn:liberty:ac:2003-08">
  <xs:element name="MobileTwoFactorUnregistered">
    <xs:complexType>
      <xs:complexContent>
        <xs:restriction base="IdentificationType">
          <xs:sequence>
            <xs:element ref="SwitchAudit"/>
            <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>

5.2.4. MobileTwoFactorUnregistered

Reflects no mobile customer registration procedures and a two-factor based authentication, such as secure device and user PIN. This context class is useful when a service other than the mobile operator wants to link their customer ID to a mobile supplied two-factor authentication service by capturing mobile phone data at enrollment.

5.2.4.1. Associated Liberty URI

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

5.2.4.2. Class Schema

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:liberty:ac:2003-08"
<xs:complexType name="MobileTwoFactorUnregisteredPrivateKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="PrivateKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyActivation" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="KeyStorage" minOccurs="0"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="MobileTwoFactorUnregisteredSecretKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="SecretKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyActivation"/>
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="MobileTwoFactorUnregisteredKeyActivationType">
  <xs:complexContent>
    <xs:restriction base="KeyActivationType">
      <xs:sequence>
        <xs:element ref="ActivationPin"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="MobileTwoFactorUnregisteredKeyStorageType">
  <xs:complexContent>
    <xs:restriction base="KeyStorageType">
      <xs:attribute name="medium" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="MobileDevice"/>
            <xs:enumeration value="MobileAuthCard"/>
            <xs:enumeration value="smartcard"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="MobileTwoFactorUnregisteredSecurityAuditType">
  <xs:complexContent>
    <xs:restriction base="SecurityAuditType">
      <xs:sequence>
        <xs:element ref="SwitchAudit"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="MobileTwoFactorUnregisteredIdentificationType">
  <xs:complexContent>
    <xs:restriction base="IdentificationType">
      <xs:attribute name="nym"/>
      <xs:attribute name="nym"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
5.2.5. MobileOneFactorContract

Reflects mobile contract customer registration procedures and a single factor authentication. For example, a digital signing device with tamper resistant memory for key storage, such as the mobile MSISDN, but no required PIN or biometric for real-time user authentication.

5.2.5.1. Associated Liberty URI

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

5.2.5.2. Class Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:liberty:ac:2003-08"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="urn:liberty:ac:2003-08"
version="1.2-08" finalDefault="extension">
<xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
<xs:annotation>
<xs:complexType name="MobileOneFactorContractAuthenticatorType">
<xs:complexContent>
<xs:restriction base="AuthenticatorType">
<xs:choice maxOccurs="1">
<xs:element ref="DigSig"/>
<xs:element ref="ZeroKnowledge"/>
<xs:element ref="SharedSecretChallengeResponse"/>
<xs:element ref="AsymmetricDecryption"/>
<xs:element ref="AsymmetricKeyAgreement"/>
<xs:element ref="SharedSecretDynamicPlaintext"/>
</xs:choice>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="MobileOneFactorContractAuthenticatorTransportProtocolType">
<xs:complexContent>
<xs:restriction base="AuthenticatorTransportProtocolType">
<xs:choice>
<xs:element ref="MobileNetworkNoEncryption"/>
<xs:element ref="MobileNetworkRadioEncryption"/>
<xs:element ref="MobileNetworkEndToEndEncryption"/>
<xs:element ref="WTLS"/>
</xs:choice>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="MobileOneFactorContractOperationalProtectionType">
<xs:complexContent>
</xs:complexContent>
</xs:complexType>
```
<xs:restriction base="OperationalProtectionType">
  <xs:sequence>
    <xs:element ref="SecurityAudit"/>
    <xs:element ref="DeactivationCallCenter"/>
    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:restriction>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="MobileOneFactorContractTechnicalProtectionType">
  <xs:complexContent>
    <xs:restriction base="TechnicalProtectionType">
      <xs:choice>
        <xs:element ref="PrivateKeyProtection"/>
        <xs:element ref="SecretKeyProtection"/>
      </xs:choice>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="MobileOneFactorContractPrivateKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="PrivateKeyProtectionType">
      <xs:sequence maxOccurs="1">
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="MobileOneFactorContractSecretKeyProtectionType">
  <xs:complexContent>
    <xs:restriction base="SecretKeyProtectionType">
      <xs:sequence>
        <xs:element ref="KeyStorage"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="MobileOneFactorContractKeyStorageType">
  <xs:complexContent>
    <xs:restriction base="KeyStorageType">
      <xs:attribute name="medium" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="MobileDevice"/>
            <xs:enumeration value="MobileAuthCard"/>
            <xs:enumeration value="smartcard"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="MobileOneFactorContractSecurityAuditType">
  <xs:complexContent>
    <xs:restriction base="SecurityAuditType">
      <xs:sequence>
        <xs:element ref="SwitchAudit"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="MobileOneFactorContractIdentificationType">
  <xs:complexContent>
    <xs:restriction base="IdentificationType">
      <xs:sequence>
        <xs:element ref="PhysicalVerification"/>
        <xs:element ref="WrittenConsent"/>
        <xs:element ref="GoverningAgreements"/>
        <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="nym">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="anonymity"/>
            <xs:enumeration value="verinymity"/>
            <xs:enumeration value="pseudonymity"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

5.2.6. MobileTwoFactorContract

Reflects mobile contract customer registration procedures and a two-factor based authentication. For example, a digital signing device with tamper resistant memory for key storage, such as a GSM SIM, that requires explicit proof of user identity and intent, such as a PIN or biometric.

5.2.6.1. Associated Liberty URI

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

5.2.6.2. Class Schema

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:liberty:ac:2003-08"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:liberty:ac:2003-08"
  version="1.2-08"
  finalDefault="extension">
  <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
  <xs:complexType name="MobileTwoFactorContractAuthenticatorType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorType">
        <xs:choice>
          <xs:element ref="DigSig"/>
          <xs:element ref="ZeroKnowledge"/>
          <xs:element ref="SharedSecretChallengeResponse"/>
          <xs:element ref="AsymmetricDecryption"/>
          <xs:element ref="AsymmetricKeyAgreement"/>
          <xs:element ref="SharedSecretDynamicPlaintext"/>
          <xs:sequence>
            <xs:element ref="Password" minOccurs="1"/>
          </xs:choice>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
  </xs:complexType>
</xs:schema>
<xs:element ref="SharedSecretDynamicPlaintext"/>
<xs:element ref="SharedSecretChallengeResponse"/>
</xs:choice>
<xs:element ref="Extension" maxOccurs="unbounded"/>
</xs:sequence>
</xs:choice>
</xs:restriction>
</xs:complexType>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="MobileTwoFactorContractOperationalProtectionType">
<xs:complexContent>
<xs:restriction base="OperationalProtectionType">
<xs:sequence>
<xs:element ref="SecurityAudit"/>
<xs:element ref="DeactivationCallCenter"/>
<xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="MobileTwoFactorContractTechnicalProtectionType">
<xs:complexContent>
<xs:restriction base="TechnicalProtectionType">
<xs:choice>
<xs:element ref="PrivateKeyProtection"/>
<xs:element ref="SecretKeyProtection"/>
</xs:choice>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="MobileTwoFactorContractPrivateKeyProtectionType">
<xs:complexContent>
<xs:restriction base="PrivateKeyProtectionType">
<xs:sequence>
<xs:element ref="KeyActivation" minOccurs="1" maxOccurs="1"/>
<xs:element ref="KeyStorage" minOccurs="0"/>
<xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="MobileTwoFactorContractSecretKeyProtectionType">
<xs:complexContent>
<xs:restriction base="SecretKeyProtectionType">
<xs:sequence>
<xs:element ref="KeyActivation"/>
<xs:element ref="KeyStorage"/>
<xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
5.2.7. 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.
5.2.7.1. Associated Liberty URI

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

5.2.7.2. Class Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:liberty:ac:2003-08"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns="urn:liberty:ac:2003-08"
    version="1.2-08"
    finalDefault="extension">
    <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
    <xs:annotation>
        <xs:documentation>
            http://www.projectliberty.org/schemas/authctx/classes/Password
        </xs:documentation>
    </xs:annotation>
    <xs:complexType name="PasswordAuthenticatorType">
        <xs:complexContent>
            <xs:restriction base="AuthenticatorType">
                <xs:choice>
                    <xs:element ref="Password"/>
                </xs:choice>
            </xs:restriction>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="PasswordPasswordType">
        <xs:complexContent>
            <xs:restriction base="PasswordType">
                <xs:sequence>
                    <xs:element ref="Length" minOccurs="1"/>
                    <xs:element ref="Generation" minOccurs="0"/>
                    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:restriction>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="PasswordLengthType">
        <xs:complexContent>
            <xs:restriction base="LengthType">
                <xs:attribute name="min" use="required">
                    <xs:simpleType>
                        <xs:restriction base="xs:integer">
                            <xs:minInclusive value="3"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
                <xs:attribute type="xs:integer" use="optional"/>
            </xs:restriction>
        </xs:complexContent>
    </xs:complexType>
</xs:schema>
```

5.2.8. PasswordProtectedTransport

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

5.2.8.1. Associated Liberty URI
5.2.8.2. Class Schema

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:liberty:ac:2003-08"
    xmlns="urn:liberty:ac:2003-08"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    version="1.2-08"
    finalDefault="extension">
    <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
    <xs:annotation>
        <xs:documentation>
            http://www.projectliberty.org/schemas/authctx/classes/PasswordProtectedTransport</xs:documentation>
    </xs:annotation>
    <xs:complexType name="PasswordProtectedTransportAuthenticatorType">
        <xs:complexContent>
            <xs:restriction base="AuthenticatorType">
                <xs:choice>
                    <xs:element ref="Password"/>
                </xs:choice>
            </xs:restriction>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="PasswordProtectedTransportPasswordType">
        <xs:complexContent>
            <xs:restriction base="PasswordType">
                <xs:sequence>
                    <xs:element ref="Length"/>
                    <xs:element ref="Generation" minOccurs="0"/>
                    <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
            </xs:restriction>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="PasswordProtectedTransportLengthType">
        <xs:complexContent>
            <xs:restriction base="LengthType">
                <xs:attribute name="min" use="required">
                    <xs:simpleType>
                        <xs:restriction base="xs:integer">
                            <xs:minInclusive value="3"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="max" type="xs:integer" use="optional"/>
            </xs:restriction>
        </xs:complexContent>
    </xs:complexType>
    <xs:complexType name="PasswordProtectedTransportAuthenticatorTransportProtocolType">
        <xs:complexContent>
            <xs:restriction base="AuthenticatorTransportProtocolType">
                <xs:choice>
                    <xs:element ref="SSL"/>
                </xs:choice>
            </xs:restriction>
        </xs:complexContent>
    </xs:complexType>
</xs:schema>
```
5.2.9. PreviousSession

The PreviousSession 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 Principals 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.

5.2.9.1. Associated Liberty URI

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

5.2.9.2. Class Schema

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:liberty:ac:2003-08"
xmlns="urn:liberty:ac:2003-08"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
version="1.2-08"
finalDefault="extension">
<xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
<xs:annotation>
<xs:documentation> http://www.projectliberty.org/schemas/authctx/classes/PreviousSession</xs:documentation>
</xs:annotation>
<xs:complexType name="PreviousSessionAuthenticatorType">
<xs:complexContent>
<xs:restriction base="AuthenticatorType">
<xs:choice>
<xs:element ref="PreviousSession"/>
</xs:choice>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
</xs:schema>

5.2.10. Smartcard

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

5.2.10.1. Associated Liberty URI

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

5.2.10.2. Class Schema

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:liberty:ac:2003-08"
xmlns="urn:liberty:ac:2003-08"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
version="1.2-08"
finalDefault="extension">
<xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
<xs:annotation>
<xs:documentation> http://www.projectliberty.org/schemas/authctx/classes/Smartcard</xs:documentation>
</xs:annotation>
5.2.11. SmartcardPKI

The SmartcardPKI 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.

5.2.11.1. Associated Liberty URI

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

5.2.11.2. Class Schema

```xml
<xs:schema targetNamespace="urn:liberty:ac:2003-08"
  xmlns="urn:liberty:ac:2003-08"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  version="1.2-08"
  finalDefault="extension">
  <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
  <xs:annotation>
    <xs:documentation>
      http://www.projectliberty.org/schemas/authctx/classes/SmartcardPKI
    </xs:documentation>
  </xs:annotation>

  <xs:complexType name="SmartCardPKIPrincipalAuthenticationMechanismType">
    <xs:complexContent>
      <xs:restriction base="PrincipalAuthenticationMechanismType">
        <xs:sequence>
          <xs:element ref="ActivationPin"/>
          <xs:element ref="Smartcard"/>
          <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SmartCardPKIAuthenticatorType">
    <xs:complexContent>
      <xs:restriction base="AuthenticatorType">
        <xs:choice>
          <xs:element ref="AsymmetricDecryption"/>
          <xs:element ref="AsymmetricKeyAgreement"/>
          <xs:element ref="DigSig"/>
        </xs:choice>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="SmartCardPKIKeyActivationType">
    <xs:complexContent>
      <xs:restriction base="KeyActivationType">
        <xs:sequence>
          <xs:element ref="PrivateKey"/>
          <xs:element ref="ActivationPin"/>
        </xs:sequence>
      </xs:restriction>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```
5.2.12. SoftwarePKI

The Software-PKI class is identified when a Principal uses an X.509 certificate stored in software to authenticate to the identity provider.

5.2.12.1. Associated Liberty URI

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

5.2.12.2. Class Schema
<xs:restriction base="PrincipalAuthenticationMechanismType">
  <xs:element ref="ActivationPin"/>
  <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/></xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="SoftwarePKIAuthenticatorType">
  <xs:complexContent>
    <xs:restriction base="AuthenticatorType">
      <xs:choice>
        <xs:element ref="AsymmetricDecryption"/>
        <xs:element ref="AsymmetricKeyAgreement"/>
        <xs:element ref="DigSig"/>
      </xs:choice>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="SoftwarePKIKeyActivationType">
  <xs:complexContent>
    <xs:restriction base="KeyActivationType">
      <xs:choice>
        <xs:element ref="ActivationPin"/>
      </xs:choice>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="SoftwarePKIKeyStorageType">
  <xs:complexContent>
    <xs:restriction base="KeyStorageType">
      <xs:attribute name="medium" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="memory"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>

5.2.13. TimeSyncToken

The TimeSyncToken class is identified when a Principal authenticates through a time synchronization token.

5.2.13.1. Associated Liberty URI
5.2.13.2. Class Schema

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:liberty:ac:2003-08"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="urn:liberty:ac:2003-08"
finalDefault="extension">
<xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
<xs:annotation>
</xs:annotation>
<xs:complexType name="TimeSyncTokenPrincipalAuthenticationMechanismType">
<xs:complexContent>
<xs:restriction base="PrincipalAuthenticationMechanismType">
<xs:choice>
<xs:element ref="Token"/>
</xs:choice>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="TimeSyncTokenTokenType">
<xs:complexContent>
<xs:restriction base="TokenType">
<xs:sequence>
<xs:element ref="TimeSyncToken"/>
<xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="TimeSyncTokenTimeSyncTokenType">
<xs:complexContent>
<xs:restriction base="TimeSyncTokenType">
<xs:attribute name="DeviceType" use="required">
<xs:simpleType>
<xs:restriction base="xs:NMTOKEN">
<xs:enumeration value="hardware"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="SeedLength" use="required">
<xs:simpleType>
<xs:restriction base="xs:integer">
<xs:enumeration value="64"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:attribute name="DeviceInHand" use="required">
<xs:simpleType>
<xs:restriction base="xs:NMTOKEN">
<xs:enumeration value="true"/>
</xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:restriction>
</xs:complexContent>
</xs:complexType>
</xs:schema>

5.3. Authentication Context Classes Extensibility
1999 As did the core Authentication Context Statement schema, the separate Authentication Context Classes schemas allow
2000 the <Extension> element in certain locations of the tree structure. In general, where the <Extension> element occurred
2001 as a child of a <Choice> element, this option was removed in creating the appropriate class schema definition as an
2002 extension of the base type. When the <Extension> element occurred as an optional child of a <Sequence> element,
2003 the <Extension> element was allowed to remain in addition to any required elements.

2004 Consequently, authentication context statements can include the <Extension> element (with additional elements in
2005 different namespaces) and still conform to authentication context class schemas (if they meet the other requirements
2006 of the schema of course)

2007 The Authentication Context Class schemas extend (as restrictions) appropriate type definitions in the core Authentica-
2008 tion Context Statement schema. As an extension point, the Authentication Context Classes schemas themselves can be
2009 extended - their type definitions serving as base types in some other schema (potentially defined by some community
2010 wishing a more tightly defined authentication context class). To prevent logical inconsistencies, any such extensions
2011 can only further constrain the type definitions of the core Authentication Context Statement schema. To enforce this
2012 constraint, the Authentication Context Class schemas are defined with the finalDefault="extension" attribute on the
2013 <schema> element to prevent this type of extension derivation.

2014 **5.4. Authentication Context Classes Processing Rules**

2015 The processing rules for both Service and Identity Provider for Authentication Context Classes are listed in [Libe-
2016 ryProtSchema].
References

