



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36

Liberty Authentication Context Specification

Draft Version 1.1-07

November 5th 2002

37 **Document Description:** draft-liberty-architecture-authentication-context-07

38

39 **Notice**

40 Copyright © 2002 ActivCard; American Express Travel Related Services; America Online, Inc.; Bank of
41 America; Bell Canada; Cingular Wireless; Cisco Systems, Inc.; Citigroup; Communicator, Inc.; Consignia;
42 Deloitte & Touche LLP; EarthLink, Inc.; Electronic Data Systems, Inc.; Entrust, Inc.; Ericsson; Fidelity
43 Investments; France Telecom; Gemplus; General Motors; Hewlett-Packard Company; i2 Technologies, Inc.;
44 Intuit Inc.; MasterCard International; NEC Corporation; Netegrity; NeuStar; Nextel Communications; Nippon
45 Telegraph and Telephone Company; Nokia Corporation; Novell, Inc.; NTT DoCoMo, Inc.; OneName
46 Corporation; Openwave Systems Inc.; PricewaterhouseCoopers LLP; Register.com; RSA Security Inc; Sabre
47 Holdings Corporation; SAP AG; SchlumbergerSema; SK Telecom; Sony Corporation; Sun Microsystems,
48 Inc.; United Airlines; VeriSign, Inc.; Visa International; Vodafone Group Plc; Wave Systems. All rights
49 reserved.

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

55 Implementation of the Specifications may involve the use of one or more of the following United States
56 Patents claimed by AOL Time Warner, Inc.: No.5,774,670, No.6,134,592, No.5,826,242, No. 5,825,890, and
57 No.5,671,279. The Sponsors of the Specification take no position concerning the evidence, validity or scope
58 of the claimed subject matter of the aforementioned patents. Implementation of certain elements of this
59 Specification may also require licenses under third party intellectual property rights other than those identified
60 above, including without limitation, patent rights. The Sponsors of the Specification are not and shall not be
61 held responsible in any manner for identifying or failing to identify any or all such intellectual property rights
62 that may be involved in the implementation of the Specification.

63 **This Specification is provided "AS IS", and no participant in the Liberty Alliance makes any warranty**
64 **of any kind, express or implied, including any implied warranties of merchantability, non-infringement**
65 **or third party intellectual property rights, and fitness for a particular purpose.**

66 Liberty Alliance Project
67 Licensing Administrator
68 c/o IEEE-ISTO
69 445 Hoes Lane, P.O. Box 1331
70 Piscataway, NJ 08855-1331, USA
71

71
72

72 **Editors**

73 Paul Madsen, Entrust, (paul.madsen@entrust.com)

74 John Kemp, IEEE-ISTO

75 **Contributors**

ActivCard
American Express Travel Related Services
America Online, Inc.
Bank of America
Bell Canada
Cingular Wireless
Cisco Systems, Inc.
Citigroup
Communicator, Inc.
Consignia
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
NEC Corporation

Netegrity
NeuStar
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
SK Telecom
Sony Corporation
Sun Microsystems, Inc.
United Airlines
VeriSign, Inc.
Visa International
Vodafone Group Plc
Wave Systems

76
77
78

79 **Document History**

Rev	Date	By Whom	Description
00	15-Mar-02	Paul Madsen, Entrust	Initial draft
01	20-Mar-02	Paul Madsen, Entrust	Edited to reflect <ul style="list-style-type: none"> - typos/grammar - concept of authentication quality - schema mods - bindings to SAML <Request> and <Response>
02	02-Apr-02	Paul Madsen	Edited to reflect <ul style="list-style-type: none"> - removal of 'quality' - introduction of service provider requesting a 'better' class - new submitted classes

03	16-Apr-02	Paul Madsen	To reflect <ul style="list-style-type: none"> - Minor edits - John Beatty's schema edits - New AuthenticationContextStatement element
04	April 29, 02	Paul Madsen	Edited after Paris meetings to include <ul style="list-style-type: none"> • Security Considerations removed • New mobile profiles and associated schema mods • 'profile' name change to 'class' • Classes rearranged to guard against implied ranking
05	May 10, 2002	Paul Madsen	<ul style="list-style-type: none"> • Editorial review • Various CRs
06	May 16,2002	Paul Madsen	<ul style="list-style-type: none"> • Removed unused elements from schema • Converted class definitions to XML Schemas
07	Nov 5,2002	John Kemp	<ul style="list-style-type: none"> • General, format edits • Updated references

80 **Table of Contents**

81 1 Introduction 7

82 1.1 Notation 7

83 2 Overview 8

84 3 Authentication Context 8

85 3.1 Authentication Context Classes 9

86 3.2 Authentication Quality 11

87 3.2.1 Service Provider Request 11

88 3.2.2 Identity Provider Response 11

89 4 Previous work 12

90 4.1 PKI 12

91 4.2 SAML 12

92 5 Liberty Authentication Context Mechanisms 13

93 5.1 Authentication Context Classes 13

94 5.1.1 MobileContract 13

95 5.1.2 MobileDigitalID 16

96 5.1.3 MobileUnregistered 18

97 5.1.4 Password 20

98 5.1.5 Password- ProtectedTransport 21

99 5.1.6 Previous-Session 22

100 5.1.7 Smartcard 23

101 5.1.8 Smartcard-PKI 24

102 5.1.9 Software-PKI 26

103 5.1.10 Time-Sync-Token 27

104 5.2 Authentication Context Schema 29

105 5.2.1 XML Schema 29

106 6 References 37

107

108

108 1 Introduction

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

111 1.1 Notation

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

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

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

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

126
127 Listings of XML schemas appear like this.

128
129 Example code listings appear like this.

130

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

- 134 • The prefix lib: stands for the Liberty namespace (<http://projectliberty.org>)
- 135 • The prefix saml: stands for the SAML assertion namespace ([http://www.oasis-
136 open.org/committees/security/docs/draft-sstc-schema-assertion-
137 15.xsd](http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-assertion-15.xsd)).
- 138 • The prefix samlp: stands for the SAML request-response protocol namespace
139 ([http://www.oasis-open.org/committees/security/docs/draft-sstc-
140 schema-protocol-15.xsd](http://www.oasis-open.org/committees/security/docs/draft-sstc-schema-protocol-15.xsd)).
- 141 • The prefix ds: stands for the W3C XML signature namespace
142 (<http://www.w3.org/2000/09/xmldsig#>).
- 143 • The prefix xsd: stands for the W3C XML schema namespace in example listings
144 (<http://www.w3.org/2001/XMLSchema>). In schema listings, this namespace is the
145 default, and no prefix is shown.

146 Definitions for Liberty-specific terms can be found in [[LibertyGloss](#)].

147 2 Overview

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

164
165 *Authentication context* is defined as the information additional to the authentication assertion itself
166 that the service provider may require before it makes an entitlements decision.

167 3 Authentication Context

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

- 171
- 172 • Initial user identification mechanisms (for example, face-to-face, online, shared secret)
- 173 • Mechanisms for minimizing compromise of a Principal's credentials (for example, credential
174 renewal frequency, client-side key generation)
- 175 • Mechanisms for storing and protecting credentials (for example, smartcard, password rules)
- 176 • Authentication mechanism (for example, password, certificate-based SSL)

177
178 The variations and permutations in the examples above guarantee that not all authentication
179 assertions are the same; a particular authentication assertion will be characterized by the values for
180 each of these variables. A somewhat helpful model is to think of an authentication assertion as
181 defined by its coordinates in a multidimensional space. This model is demonstrated in Figure 1
182 (where only three axes are shown).

183

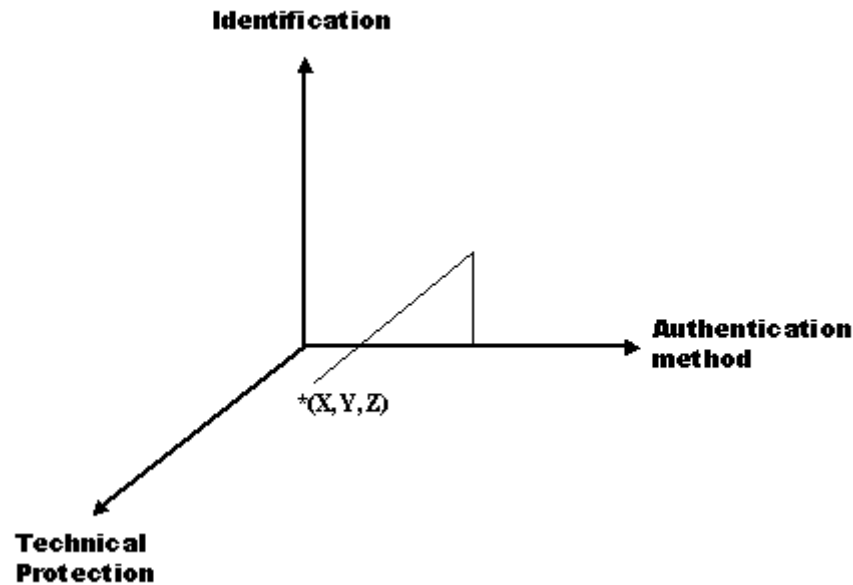


Figure 1: Authentication assertion as defined by its coordinates in multidimensional space

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

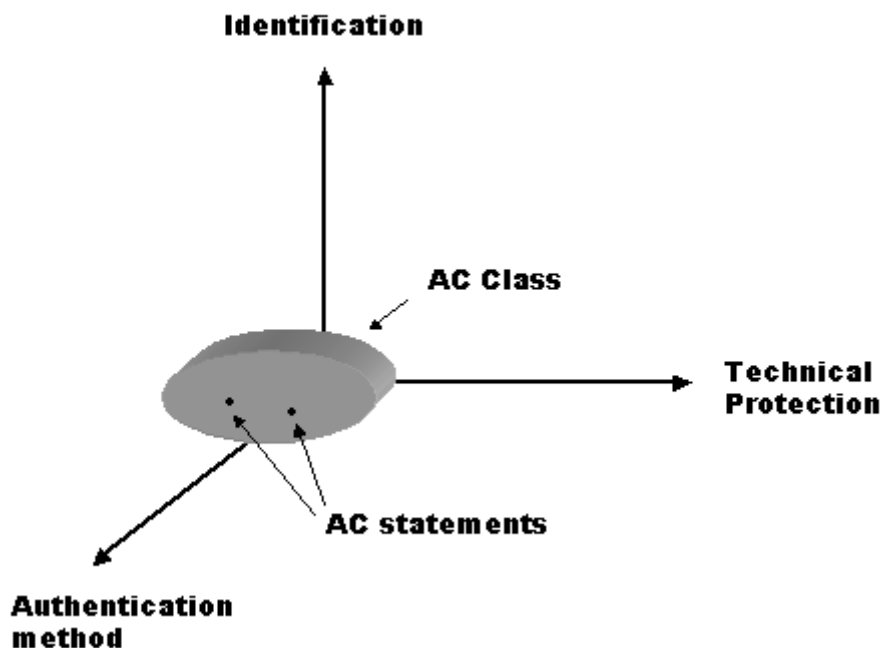
Liberty can simplify for service providers the task of assessing and comparing authentication assertions by defining particular authentication contexts that are representative of current technologies and practices among identity providers. For instance, a typical authentication context will be when a 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).

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

219 Rather than a class being a rigid collection of these characteristics, a class will define a set of
220 conformant authentication context statements (for example, multiple and different authentication
221 context statements will satisfy the requirements of a given class). The relationship between an
222 authentication context class and particular authentication context statements is shown in Figure 2,
223 where all the authentication context statements satisfy the requirements expressed by the class.
224



225
226 **Figure 2: Relationship between authentication context class and statements**

227
228 By introducing the additional layer of classes and by defining an initial list of representative and
229 flexible classes, Liberty architecture

- 230
- 231 • Makes it easier for the identity provider and service provider to come to an agreement on
232 what are acceptable authentication contexts by giving them a framework for discussion.
 - 233 • Makes it easier for service providers to indicate their preferences when requesting a step-up
234 authentication assertion from an identity provider.
 - 235 • Simplifies for service providers the burden of processing authentication context statements
236 by giving them the option of being satisfied by the associated class.
 - 237 • Protects service providers from impact of new authentication technologies.
 - 238 • Makes it easier for identity providers to publish their authentication capabilities, for example,
239 through WSDL.

240 **3.2 Authentication Quality**

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


248 **3.2.1 Service Provider Request**

249 To provide the desired flexibility without requiring Liberty to itself assess the quality of particular
250 authentication classes, the service provider will be provided a flexible mechanism by which it can
251 indicate its preferences for authentication context to the identity provider. The
252 <lib:AuthnAndFedRequest> message will allow the service provider to request any of the following:
253

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

259
260 Option 1 will require that the identity provider and service provider have previously agreed on the
261 details of a particular authentication context that either does not fall into one of the Liberty-defined
262 authentication context classes or needs to be constrained more tightly.

263
264 Option 2 is expected to be the typical scenario.

265
266 For option 3, the decision as to what is better is left to the entity best qualified to make that
267 determination, the identity provider. The service provider, trusting the identity provider's judgment,
268 will accept the assertion it receives back because it will be confident the assertion meets (or
269 exceeds) the provider's requirements. 

270
271 Option 4 will give the service provider greater control over the authentication context classes to
272 which the authentication assertions it receives conform. The identity provider is given no leeway in
273 providing an authentication assertion conforming to a class not on the list.

274
275 If the service provider does not specify any of the above options in the <lib:AuthnAndFedRequest>,
276 the identity provider will be free to provide an authentication context of its choosing.

277 **3.2.2 Identity Provider Response**

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

281
282 The returned authentication assertion will include a URI specifying the associated authentication
283 context statement.

284 4 Previous work

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

286 4.1 PKI

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

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

318 4.2 SAML

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

323
324
325
326
327
328

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

329
330
331
332
333
334
335
336
337

```
Format="http://www.oasis-open.org/committees/security/docs/draft-  
    </saml:NameIdentifier>  
    </saml:Subject>  
</saml:AuthenticationStatement>  
<saml:Advice>  
<!--additional elements in separate namespace ->  
</saml:Advice>  
</saml:Assertion>
```

338

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

341

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

346

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

349

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

354

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

358 5 Liberty Authentication Context Mechanisms

359 5.1 Authentication Context Classes

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

361

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

363

364 Classes are identified by URIs with the initial stem:

365

366 <http://www.projectliberty.org/schemas/authctx/classes>

367

368 5.1.1 MobileContract

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

371 5.1.1.1 Associated Liberty URI

372 <http://www.projectliberty.org/schemas/authctx/classes/MobileContract>

373 **5.1.1.2 Class Schema**

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

```
<?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>
      </xs:element>
    </xs:element>
  </xs:element>
</xs:schema>
```

```
440     <xs:element name="GoverningAgreements">
441         <xs:complexType>
442             <xs:sequence>
443                 <xs:element minOccurs="1" maxOccurs="1"
444 ref="GoverningAgreementRef"/>
445                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
446 processContents="lax" /></xs:sequence>
447             </xs:complexType>
448         </xs:element>
449     <xs:element name="Identification">
450         <xs:complexType>
451             <xs:sequence>
452                 <xs:element minOccurs="1" maxOccurs="1"
453 ref="PhysicalVerification"/>
454                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
455 processContents="lax" /></xs:sequence>
456                 <xs:attribute name="nym" type="xs:string" use="required"/>
457             </xs:complexType>
458         </xs:element>
459     <xs:element name="MobileAuthCard">
460         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
461 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
462     </xs:element>
463     <xs:element name="MobileDevice">
464         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
465 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
466     </xs:element>
467     <xs:element name="MobileNetwork">
468         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
469 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
470     </xs:element>
471     <xs:element name="OperationalProtection">
472         <xs:complexType>
473             <xs:sequence>
474                 <xs:element minOccurs="1" maxOccurs="1" ref="SecurityAudit"/>
475                 <xs:element minOccurs="1" maxOccurs="1"
476 ref="DeactivationCallCenter"/>
477                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
478 processContents="lax" /></xs:sequence>
479             </xs:complexType>
480         </xs:element>
481
482     <xs:element name="PhysicalVerification">
483         <xs:complexType>
484             <xs:attribute name="credentialLevel" type="xs:string" use="required"/>
485         </xs:complexType>
486     </xs:element>
487
488     <xs:element name="SecurityAudit">
489         <xs:complexType>
490             <xs:sequence>
491                 <xs:element minOccurs="1" maxOccurs="1" ref="SwitchAudit"/>
492                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
493 processContents="lax" /></xs:sequence>
494             </xs:complexType>
495         </xs:element>
496     <xs:element name="SharedKeyProtection">
497         <xs:complexType>
498             <xs:choice>
499                 <xs:element minOccurs="1" maxOccurs="1" ref="MobileAuthCard"/>
500                 <xs:element minOccurs="1" maxOccurs="1" ref="MobileDevice"/>
501             </xs:choice>
502         </xs:complexType>
503     </xs:element>
504     <xs:element name="SharedSecretChallengeResponse">
505         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
506 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
507     </xs:element>
```

```
508     <xs:element name="SwitchAudit">
509         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
510 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
511     </xs:element>
512     <xs:element name="TechnicalProtection">
513         <xs:complexType>
514             <xs:sequence>
515                 <xs:element minOccurs="1" maxOccurs="1"
516 ref="SharedKeyProtection"/>
517                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
518 processContents="lax" /></xs:sequence>
519             </xs:complexType>
520         </xs:element>
521 </xs:schema>
```

522 5.1.2 MobileDigitalID

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

525 5.1.2.1 Associated Liberty URI

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

527 5.1.2.2 Class Schema

```
528 <?xml version="1.0" encoding="UTF-8"?>
529 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
530 <annotation>
531 <documentation> http://www.projectliberty.org/schemas/authctx/classes/MobileDigitalID
532 </documentation>
533 </annotation>
534     <xs:element name="AuthenticationContextStatement">
535         <xs:complexType>
536             <xs:sequence>
537                 <xs:element ref="Identification"/>
538                 <xs:element ref="TechnicalProtection"/>
539                 <xs:element ref="AuthenticationMethod"/>
540                 <xs:element ref="OperationalProtection"/>
541                 <xs:element ref="GoverningAgreements"/>
542                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
543 processContents="lax" /></xs:sequence>
544             </xs:complexType>
545         </xs:element>
546     <xs:element name="AuthenticationMethod">
547         <xs:complexType>
548             <xs:sequence>
549                 <xs:element ref="Authenticator"/>
550                 <xs:element ref="AuthenticatorTransportProtocol"/>
551                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
552 processContents="lax" /></xs:sequence>
553             </xs:complexType>
554         </xs:element>
555     <xs:element name="Authenticator">
556         <xs:complexType>
557             <xs:choice>
558                 <xs:element ref="Dig-sig"/>
559                 <xs:element ref="ZeroKnowledge"/>
560             </xs:choice>
561         </xs:complexType>
562     </xs:element>
```



```
567 <xs:element name="AuthenticatorTransportProtocol">
568   <xs:complexType>
569     <xs:choice>
570       <xs:element ref="MobileNetwork"/>
571       <xs:element ref="SSL"/>
572       <xs:element ref="WTLS"/>
573       <xs:element ref="IPSec"/>
574     </xs:choice>
575   </xs:complexType>
576 </xs:element>
577 <xs:element name="DeactivationCallCenter">
578   <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
579 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
580 </xs:element>
581 <xs:element name="Dig-sig">
582   <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
583 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
584 </xs:element>
585 <xs:element name="GoverningAgreementRef">
586   <xs:complexType>
587     <xs:attribute name="ref"
588 fixed="http://SomeMobileforum.org/namespaces/authcontext/classes/Mobile-Class3.pdf"/>
589   </xs:complexType>
590 </xs:element>
591 <xs:element name="GoverningAgreements">
592   <xs:complexType>
593     <xs:sequence>
594       <xs:element ref="GoverningAgreementRef"/>
595       <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
596 processContents="lax" /></xs:sequence>
597     </xs:complexType>
598 </xs:element>
599 <xs:element name="IPSec">
600   <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
601 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
602 </xs:element>
603 <xs:element name="Identification">
604   <xs:complexType>
605     <xs:sequence>
606       <xs:element ref="PhysicalVerification"/>
607       <xs:element ref="WrittenConsent"/>
608       <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
609 processContents="lax" /></xs:sequence>
610     <xs:attribute name="nym" type="xs:string" use="required"/>
611   </xs:complexType>
612 </xs:element>
613 <xs:element name="KeyStorage">
614   <xs:complexType>
615     <xs:attribute name="medium" type="xs:string" use="required"/>
616   </xs:complexType>
617 </xs:element>
618 <xs:element name="MobileNetwork">
619   <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
620 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
621 </xs:element>
622 <xs:element name="OperationalProtection">
623   <xs:complexType>
624     <xs:sequence>
625       <xs:element ref="SecurityAudit"/>
626       <xs:element ref="DeactivationCallCenter"/>
627       <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
628 processContents="lax" /></xs:sequence>
629     </xs:complexType>
630 </xs:element>
631 <xs:element name="PhysicalVerification">
632   <xs:complexType>
633     <xs:attribute name="credentialLevel" type="xs:string" use="required"/>
634 </xs:complexType>
```

```
635     </xs:element>
636     <xs:element name="PrivateKeyProtection">
637         <xs:complexType>
638             <xs:sequence>
639                 <xs:element ref="KeyStorage"/>
640                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
642             </xs:complexType>
643         </xs:element>
644         <xs:element name="SSL">
645             <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
647         </xs:element>
648         <xs:element name="SecurityAudit">
649             <xs:complexType>
650                 <xs:sequence>
651                     <xs:element ref="SwitchAudit"/>
652                     <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
654                 </xs:complexType>
655             </xs:element>
656             <xs:element name="SwitchAudit">
657                 <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
659             </xs:element>
660             <xs:element name="TechnicalProtection">
661                 <xs:complexType>
662                     <xs:sequence>
663                         <xs:element ref="PrivateKeyProtection"/>
664                         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
666                     </xs:complexType>
667                 </xs:element>
668                 <xs:element name="WTLS">
669                     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
671                 </xs:element>
672                 <xs:element name="WrittenConsent">
673                     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
675                 </xs:element>
676                 <xs:element name="ZeroKnowledge">
677                     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
679                 </xs:element>
680 </xs:schema>
```

681 5.1.3 MobileUnregistered

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

684 5.1.3.1 Associated Liberty URI

685 <http://www.projectliberty.org/schemas/authctx/classes/MobileUnregistered>

686 5.1.3.2 Class Schema

```
687 <?xml version="1.0" encoding="UTF-8"?>
688 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
691 <annotation>
```

```
693 <documentation> http://www.projectliberty.org/schemas/authctx/classes/MobileUnregistered
694 </documentation>
695 </annotation>
696
697     <xs:element name="AuthenticationContextStatement">
698         <xs:complexType>
699             <xs:sequence>
700                 <xs:element ref="TechnicalProtection"/>
701                 <xs:element ref="AuthenticationMethod"/>
702                 <xs:element ref="OperationalProtection"/>
703                 <xs:element ref="GoverningAgreements"/>
704                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
705 processContents="lax" /></xs:sequence>
706             </xs:complexType>
707         </xs:element>
708     <xs:element name="AuthenticationMethod">
709         <xs:complexType>
710             <xs:sequence>
711                 <xs:element ref="Authenticator"/>
712                 <xs:element ref="AuthenticatorTransportProtocol"/>
713                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
714 processContents="lax" /></xs:sequence>
715             </xs:complexType>
716         </xs:element>
717     <xs:element name="Authenticator">
718         <xs:complexType>
719             <xs:sequence>
720                 <xs:element ref="SharedSecretChallengeResponse"/>
721                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
722 processContents="lax" /></xs:sequence>
723             </xs:complexType>
724         </xs:element>
725     <xs:element name="AuthenticatorTransportProtocol">
726         <xs:complexType>
727             <xs:sequence>
728                 <xs:element ref="MobileNetwork"/>
729                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
730 processContents="lax" /></xs:sequence>
731             </xs:complexType>
732         </xs:element>
733     <xs:element name="DeactivationCallCenter">
734         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
735 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
736     </xs:element>
737     <xs:element name="GoverningAgreementRef">
738         <xs:complexType>
739             <xs:attribute name="ref"
740 fixed="http://SomeMobileforum.org/namespaces/authcontext/classes/Mobile-Class1.pdf"/>
741         </xs:complexType>
742     </xs:element>
743     <xs:element name="GoverningAgreements">
744         <xs:complexType>
745             <xs:sequence>
746                 <xs:element ref="GoverningAgreementRef"/>
747                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
748 processContents="lax" /></xs:sequence>
749             </xs:complexType>
750         </xs:element>
751     <xs:element name="MobileAuthCard">
752         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
753 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
754     </xs:element>
755     <xs:element name="MobileDevice">
756         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
757 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
758     </xs:element>
759     <xs:element name="MobileNetwork">
```

```
760     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"  
761 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>  
762   </xs:element>  
763   <xs:element name="OperationalProtection">  
764     <xs:complexType>  
765       <xs:sequence>  
766         <xs:element ref="SecurityAudit"/>  
767         <xs:element ref="DeactivationCallCenter"/>  
768         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"  
769 processContents="lax" /></xs:sequence>  
770       </xs:complexType>  
771     </xs:element>  
772     <xs:element name="SecurityAudit">  
773       <xs:complexType>  
774         <xs:sequence>  
775           <xs:element ref="SwitchAudit"/>  
776           <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"  
777 processContents="lax" /></xs:sequence>  
778         </xs:complexType>  
779       </xs:element>  
780     <xs:element name="SharedKeyProtection">  
781       <xs:complexType>  
782         <xs:choice>  
783           <xs:element ref="MobileAuthCard"/>  
784           <xs:element ref="MobileDevice"/>  
785         </xs:choice>  
786       </xs:complexType>  
787     </xs:element>  
788     <xs:element name="SharedSecretChallengeResponse">  
789       <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"  
790 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>  
791     </xs:element>  
792     <xs:element name="SwitchAudit">  
793       <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"  
794 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>  
795     </xs:element>  
796     <xs:element name="TechnicalProtection">  
797       <xs:complexType>  
798         <xs:sequence>  
799           <xs:element ref="SharedKeyProtection"/>  
800           <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"  
801 processContents="lax" /></xs:sequence>  
802         </xs:complexType>  
803       </xs:element>  
804 </xs:schema>
```

805 5.1.4 Password

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

808 5.1.4.1 Associated Liberty URI

809 <http://www.projectliberty.org/schemas/authctx/classes/Password>

810 5.1.4.2 Class Schema

```
811 <?xml version="1.0" encoding="UTF-8"?>  
812  
813 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">  
814  
815 <annotation>  
816 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Password  
817 </documentation>  
818 </annotation>  
819
```

```

820     <xs:element name="AuthenticationContextStatement">
821         <xs:complexType>
822             <xs:sequence>
823                 <xs:element ref="AuthenticationMethod"/>
824                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
826             </xs:complexType>
827         </xs:element>
828     <xs:element name="AuthenticationMethod">
829         <xs:complexType>
830             <xs:all>
831                 <xs:element ref="PrincipalAuthenticationMechanism"/>
832                 <xs:element ref="AuthenticatorTransportProtocol"/>
833             </xs:all>
834         </xs:complexType>
835     </xs:element>
836     <xs:element name="AuthenticatorTransportProtocol">
837         <xs:complexType>
838             <xs:sequence>
839                 <xs:element ref="HTTP"/>
840                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
842             </xs:complexType>
843         </xs:element>
844         <xs:element name="HTTP">
845             <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
846 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
847         </xs:element>
848         <xs:element name="Length">
849             <xs:complexType>
850                 <xs:attribute name="min" fixed="3"/>
851             </xs:complexType>
852         </xs:element>
853         <xs:element name="Password">
854             <xs:complexType>
855                 <xs:sequence>
856                     <xs:element ref="Length"/>
857                     <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
859                 </xs:complexType>
860             </xs:element>
861             <xs:element name="PrincipalAuthenticationMechanism">
862                 <xs:complexType>
863                     <xs:sequence>
864                         <xs:element ref="Password"/>
865                         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
867                     </xs:complexType>
868                 </xs:element>
869 </xs:schema>

```

870 5.1.5 Password- ProtectedTransport

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

873 5.1.5.1 Associated Liberty URI

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

875 5.1.5.2 Class Schema

```

876 <?xml version="1.0" encoding="UTF-8"?>
877 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
878
879

```

```

880 <annotation>
881 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Password-
882 ProtectedTransport </documentation>
883 </annotation>
884
885
886 <xs:element name="AuthenticationContextStatement">
887 <xs:complexType>
888 <xs:sequence>
889 <xs:element ref="AuthenticationMethod"/>
890 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
891 processContents="lax" /></xs:sequence>
892 </xs:complexType>
893 </xs:element>
894 <xs:element name="AuthenticationMethod">
895 <xs:complexType>
896 <xs:all>
897 <xs:element ref="PrincipalAuthenticationMechanism"/>
898 <xs:element ref="AuthenticatorTransportProtocol"/>
899 </xs:all>
900 </xs:complexType>
901 </xs:element>
902 <xs:element name="AuthenticatorTransportProtocol">
903 <xs:complexType>
904 <xs:sequence>
905 <xs:element ref="SSL"/>
906 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
907 processContents="lax" /></xs:sequence>
908 </xs:complexType>
909 </xs:element>
910 <xs:element name="Length">
911 <xs:complexType>
912 <xs:attribute name="min" fixed="3"/>
913 </xs:complexType>
914 </xs:element>
915 <xs:element name="Password">
916 <xs:complexType>
917 <xs:sequence>
918 <xs:element ref="Length"/>
919 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
920 processContents="lax" /></xs:sequence>
921 </xs:complexType>
922 </xs:element>
923 <xs:element name="PrincipalAuthenticationMechanism">
924 <xs:complexType>
925 <xs:sequence>
926 <xs:element ref="Password"/>
927 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
928 processContents="lax" /></xs:sequence>
929 </xs:complexType>
930 </xs:element>
931 <xs:element name="SSL">
932 <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
933 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
934 </xs:element>
935 </xs:schema>

```

936 5.1.6 Previous-Session

937 The Previous-Session class is identified when a Principal had authenticated to an identity provider at
938 some point in the past using any authentication context supported by that identity provider.
939 Consequently, a subsequent authentication event that the identity provider will assert to the service
940 provider may be significantly separated in time from the Principal's current resource access request.
941

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

946 **5.1.6.1 Associated Liberty URI**

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

948 **5.1.6.2 Class Schema**

```
949 <?xml version="1.0" encoding="UTF-8"?>
950 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
951 <annotation>
952 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Previous-Session
953 </documentation>
954 </annotation>
955
956     <xs:element name="AuthenticationContextStatement">
957         <xs:complexType>
958             <xs:sequence>
959                 <xs:element minOccurs="1" maxOccurs="1"
960 ref="AuthenticationMethod"/>
961                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
962 processContents="lax" /></xs:sequence>
963             </xs:complexType>
964         </xs:element>
965
966     <xs:element name="AuthenticationMethod">
967         <xs:complexType>
968             <xs:sequence>
969                 <xs:element ref="Authenticator" minOccurs="0" maxOccurs="1"/>
970                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
971 processContents="lax" /></xs:sequence>
972             </xs:complexType>
973         </xs:element>
974
975     <xs:element name="Authenticator">
976         <xs:complexType>
977             <xs:sequence>
978                 <xs:element minOccurs="1" maxOccurs="1" ref="PreviousSession"/>
979                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
980 processContents="lax" /></xs:sequence>
981             </xs:complexType>
982         </xs:element>
983
984     <xs:element name="PreviousSession">
985         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
986 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
987     </xs:element>
988 </xs:schema>
```

994 **5.1.7 Smartcard**

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

997 **5.1.7.1 Associated Liberty URI**

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

999 **5.1.7.2 Class Schema**

```
1000 <?xml version="1.0" encoding="UTF-8"?>
1001
1002 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
1003
1004 <annotation>
1005 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Smartcard
1006 </documentation>
1007 </annotation>
1008
1009     <xs:element name="AuthenticationContextStatement">
1010         <xs:complexType>
1011             <xs:sequence>
1012                 <xs:element minOccurs="1" maxOccurs="1"
1013 ref="AuthenticationMethod"/>
1014                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1015 processContents="lax" /></xs:sequence>
1016             </xs:complexType>
1017         </xs:element>
1018         <xs:element name="AuthenticationMethod">
1019             <xs:complexType>
1020                 <xs:sequence>
1021                     <xs:element minOccurs="1" maxOccurs="1"
1022 ref="PrincipalAuthenticationMechanism"/>
1023                     <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1024 processContents="lax" /></xs:sequence>
1025                 </xs:complexType>
1026             </xs:element>
1027             <xs:element name="PrincipalAuthenticationMechanism">
1028                 <xs:complexType>
1029                     <xs:sequence>
1030                         <xs:element minOccurs="1" maxOccurs="1" ref="Smartcard"/>
1031                         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1032 processContents="lax" /></xs:sequence>
1033                     </xs:complexType>
1034                 </xs:element>
1035                 <xs:element name="Smartcard">
1036                     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1037 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1038                 </xs:element>
1039 </xs:schema>
1040
```

1041 **5.1.8 Smartcard-PKI**

1042 The Smartcard-PKI class is identified when a Principal authenticates to an identity provider through
1043 a two-factor authentication mechanism using a smartcard with enclosed private key and a PIN.

1044 **5.1.8.1 Associated Liberty URI**

1045 <http://www.projectliberty.org/schemas/authctx/classes/Smartcard-PKI>

1046 **5.1.8.2 Class Schema**

```
1047 <?xml version="1.0" encoding="UTF-8"?>
1048
1049 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
1050
1051 <annotation>
```



```
1052 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Smartcard-
1053 PKI</documentation>
1054 </annotation>
1055
1056     <xs:element name="AuthenticationContextStatement">
1057         <xs:complexType>
1058             <xs:sequence>
1059                 <xs:element minOccurs="1" maxOccurs="1"
1060 ref="TechnicalProtection"/>
1061                 <xs:element minOccurs="1" maxOccurs="1"
1062 ref="AuthenticationMethod"/>
1063                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1064 processContents="lax" /></xs:sequence>
1065             </xs:complexType>
1066         </xs:element>
1067     <xs:element name="AuthenticationMethod">
1068         <xs:complexType>
1069             <xs:sequence>
1070                 <xs:element minOccurs="1" maxOccurs="1"
1071 ref="PrincipalAuthenticationMechanism"/>
1072                 <xs:element minOccurs="1" maxOccurs="1" ref="Authenticator"/>
1073                 <xs:element minOccurs="1" maxOccurs="1"
1074 ref="AuthenticatorTransportProtocol"/>
1075                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1076 processContents="lax" /></xs:sequence>
1077             </xs:complexType>
1078         </xs:element>
1079     <xs:element name="Authenticator">
1080         <xs:complexType>
1081             <xs:sequence>
1082                 <xs:element minOccurs="1" maxOccurs="1" ref="Dig-sig"/>
1083                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1084 processContents="lax" /></xs:sequence>
1085             </xs:complexType>
1086         </xs:element>
1087     <xs:element name="AuthenticatorTransportProtocol">
1088         <xs:complexType>
1089             <xs:sequence>
1090                 <xs:element minOccurs="1" maxOccurs="1" ref="SSL"/>
1091                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1092 processContents="lax" /></xs:sequence>
1093             </xs:complexType>
1094         </xs:element>
1095     <xs:element name="Dig-sig">
1096         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1097 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1098     </xs:element>
1099     <xs:element name="KeyActivation">
1100         <xs:complexType>
1101             <xs:sequence>
1102                 <xs:element minOccurs="1" maxOccurs="1" ref="Password"/>
1103                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1104 processContents="lax" /></xs:sequence>
1105             </xs:complexType>
1106         </xs:element>
1107     <xs:element name="Length">
1108         <xs:complexType>
1109             <xs:attribute name="min" type="xs:byte" use="required"/>
1110         </xs:complexType>
1111     </xs:element>
1112     <xs:element name="Password">
1113         <xs:complexType>
1114             <xs:sequence>
1115                 <xs:element minOccurs="1" maxOccurs="1" ref="Length"/>
1116                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1117 processContents="lax" /></xs:sequence>
1118             </xs:complexType>
1119         </xs:element>
```

```

1120     <xs:element name="PrincipalAuthenticationMechanism">
1121         <xs:complexType>
1122             <xs:sequence>
1123                 <xs:element minOccurs="1" maxOccurs="1" ref="Smartcard"/>
1124                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
1126             </xs:complexType>
1127         </xs:element>
1128     <xs:element name="PrivateKeyProtection">
1129         <xs:complexType>
1130             <xs:sequence>
1131                 <xs:element minOccurs="1" maxOccurs="1" ref="KeyActivation"/>
1132                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
1134             </xs:complexType>
1135         </xs:element>
1136     <xs:element name="SSL">
1137         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1139     </xs:element>
1140     <xs:element name="Smartcard">
1141         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1143     </xs:element>
1144     <xs:element name="TechnicalProtection">
1145         <xs:complexType>
1146             <xs:sequence>
1147                 <xs:element minOccurs="1" maxOccurs="1"
ref="PrivateKeyProtection"/>
1149                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
1151             </xs:complexType>
1152         </xs:element>
1153 </xs:schema>
1154

```

1155 5.1.9 Software-PKI

1156 Th Software-PKI class is identified when a Principal uses an X.509 certificate stored in software to
1157 authenticate to the identity provider over an SSL protected session.

1159 5.1.9.1 Associated Liberty URI

1160
1161 <http://www.projectliberty.org/schemas/authctx/classes/Software-PKI>

1162 5.1.9.2 Class Schema

```

1163 <?xml version="1.0" encoding="UTF-8"?>
1164 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
1165 <annotation>
1166 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Software-PKI
1167 </documentation>
1168 </annotation>
1169 <xs:element name="AuthenticationContextStatement">
1170     <xs:complexType>
1171         <xs:sequence>
1172             <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticationMethod"/>
1173         </xs:sequence>
1174     </xs:complexType>
1175 </xs:element>
1176 </xs:schema>
1177

```

```
1178         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1179 processContents="lax" /></xs:sequence>
1180     </xs:complexType>
1181 </xs:element>
1182 <xs:element name="AuthenticationMethod">
1183     <xs:complexType>
1184         <xs:sequence>
1185             <xs:element minOccurs="1" maxOccurs="1"
1186 ref="PrincipalAuthenticationMechanism"/>
1187             <xs:element minOccurs="1" maxOccurs="1" ref="Authenticator"/>
1188             <xs:element minOccurs="1" maxOccurs="1"
1189 ref="AuthenticatorTransportProtocol"/>
1190         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1191 processContents="lax" /></xs:sequence>
1192     </xs:complexType>
1193 </xs:element>
1194 <xs:element name="Authenticator">
1195     <xs:complexType>
1196         <xs:sequence>
1197             <xs:element minOccurs="1" maxOccurs="1" ref="Dig-sig"/>
1198             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1199 processContents="lax" /></xs:sequence>
1200     </xs:complexType>
1201 </xs:element>
1202 <xs:element name="AuthenticatorTransportProtocol">
1203     <xs:complexType>
1204         <xs:sequence>
1205             <xs:element minOccurs="1" maxOccurs="1" ref="SSL"/>
1206             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1207 processContents="lax" /></xs:sequence>
1208     </xs:complexType>
1209 </xs:element>
1210 <xs:element name="Dig-sig">
1211     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1212 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1213 </xs:element>
1214 <xs:element name="Password">
1215     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1216 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1217 </xs:element>
1218 <xs:element name="PrincipalAuthenticationMechanism">
1219     <xs:complexType>
1220         <xs:sequence>
1221             <xs:element minOccurs="1" maxOccurs="1" ref="Password"/>
1222             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1223 processContents="lax" /></xs:sequence>
1224     </xs:complexType>
1225 </xs:element>
1226 <xs:element name="SSL">
1227     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1228 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1229 </xs:element>
1230 </xs:schema>
```

1232 5.1.10 Time-Sync-Token

1233 The Time-Sync-Token class is identified when a Principal authenticates through a time
1234 synchronization token.

1235 5.1.10.1 Associated Liberty URI

1236 <http://www.projectliberty.org/schemas/authctx/classes/Time-Sync-Token>

1237 **5.1.10.2 Class Schema**

1238

1239

1240

1241

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295

1296

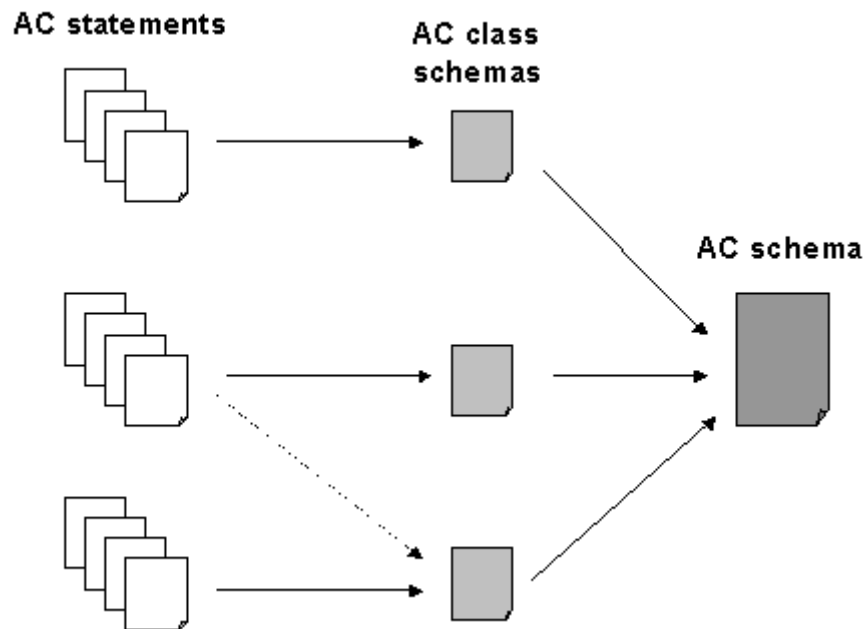
1297

```
<?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>
```

1298 **5.2 Authentication Context Schema**

1299 The relationship between authentication context statements, authentication context classes, and the
1300 authentication context XML schema is shown in Figure 3.

1301



1302

1303 **Figure 3: Relationship between authentication context statements, classes, and XML schema**

1304

1305 Authentication context statements may conform to authentication context classes, which are
1306 themselves logical subsets of the authentication context XML schema.

1307

1308

1309 **5.2.1 XML Schema**

1310

```

1311 <?xml version="1.0" encoding="UTF-8"?>
1312
1313 <schema targetNamespace="http://www.projectliberty.org/schemas/authctx/2002/05"
1314 xmlns:xsd="http://www.w3.org/2001/XMLSchema "
1315 xmlns:AC="http://www.projectliberty.org/schemas/authctx/2002/05"
1316 xmlns="http://www.w3.org/2001/XMLSchema" version="1.0">
1317
1318 <annotation>
1319 <documentation> http://www.projectliberty.org/schemas/authctx/2002/05/
1320 </documentation>
1321 </annotation>
1322
1323 <element name="AuthenticationContextStatement">
1324 <annotation>
1325 <documentation>A claim made by an identity provider with respect to
1326 the authentication context associated with an authentication assertion. </documentation>
1327 </annotation>
1328 <complexType>
1329 <sequence>
1330 <element ref="AC:Identification" minOccurs="0"/>

```

```

1331         <element ref="AC:TechnicalProtection" minOccurs="0"/>
1332         <element ref="AC:OperationalProtection" minOccurs="0"/>
1333         <element ref="AC:AuthenticationMethod" minOccurs="0"/>
1334         <element ref="AC:GoverningAgreements" minOccurs="0"/>
1335         <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1336 processContents="lax" />
1337     </sequence>
1338     <attribute name="ID" type="ID"/>
1339 </complexType>
1340 </element>
1341
1342     <element name="Identification">
1343         <annotation>
1344             <documentation>Refers to those characteristics that describe the
1345 processes and mechanisms the
1346 identity provider uses to initially create an association between a
1347 Principal and the identity
1348 (or name) by which the Principal will be known</documentation>
1349         </annotation>
1350         <complexType>
1351             <sequence>
1352                 <element ref="AC:PhysicalVerification" minOccurs="0"/>
1353                 <element ref="AC:WrittenConsent" minOccurs="0"/>
1354                 <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1355 processContents="lax" />
1356             </sequence>
1357             <attribute name="nym">
1358                 <annotation>
1359                     <documentation>This attribute indicates whether or not
1360 the Identification mechanisms allow the
1361 actions of the Principal to be linked to an actual end
1362 user.</documentation>
1363                 </annotation>
1364                 <simpleType>
1365                     <restriction base="NMTOKEN">
1366                         <enumeration value="anonymity"/>
1367                         <enumeration value="verinymity"/>
1368                         <enumeration value="pseudonymity"/>
1369                     </restriction>
1370                 </simpleType>
1371             </attribute>
1372         </complexType>
1373     </element>
1374     <element name="PhysicalVerification">
1375         <annotation>
1376             <documentation>This element indicates that identification has been
1377 performed in a physical
1378 face-to-face meeting with the principal and not in an online manner.
1379 </documentation>
1380         </annotation>
1381         <complexType>
1382             <attribute name="credentialLevel">
1383                 <simpleType>
1384                     <restriction base="NMTOKEN">
1385                         <enumeration value="primary"/>
1386                         <enumeration value="secondary"/>
1387                     </restriction>
1388                 </simpleType>
1389             </attribute>
1390         </complexType>
1391     </element>
1392     <element name="WrittenConsent">
1393         <complexType><sequence><any namespace="##any" minOccurs="0"
1394 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1395     </element>
1396     <element name="TechnicalProtection">
1397         <annotation>

```

```
1398         <documentation>Refers to those characteristics that describe how the
1399 'secret' (the knowledge or possession of which allows the Principal to authenticate to the
1400 identity provider) is kept secure</documentation>
1401         </annotation>
1402         <complexType>
1403             <sequence>
1404                 <element ref="AC:PrivateKeyProtection" minOccurs="0"/>
1405                 <element ref="AC:SharedKeyProtection" minOccurs="0"/>
1406                 <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1407 processContents="lax" />
1408             </sequence>
1409         </complexType>
1410     </element>
1411     <element name="SharedKeyProtection">
1412         <annotation>
1413             <documentation>This element indicates the types and strengths of
1414 facilities
1415 of a UA used to protect a shared secret key from unauthorized access
1416 and/or use.</documentation>
1417         </annotation>
1418         <complexType>
1419             <choice minOccurs="0">
1420                 <element ref="AC:MobileDevice"/>
1421                 <element ref="AC:MobileAuthCard"/>
1422             </choice>
1423         </complexType>
1424     </element>
1425     <element name="MobileDevice">
1426         <annotation>
1427             <documentation>This element indicates that the shared secret key is
1428 securely maintained in a mobile device
1429 (as opposed to being stored in a mobile authentication
1430 card).</documentation>
1431         </annotation>
1432         <complexType><sequence><any namespace="##any" minOccurs="0"
1433 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1434     </element>
1435     <element name="MobileAuthCard">
1436         <annotation>
1437             <documentation>This element indicates that the shared secret key is
1438 securely maintained in a mobile authentication card (e.g., a SIM card).</documentation>
1439         </annotation>
1440         <complexType><sequence><any namespace="##any" minOccurs="0"
1441 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1442     </element>
1443     <element name="PrivateKeyProtection">
1444         <annotation>
1445             <documentation>This element indicates the types and strengths of
1446 facilities
1447 of a UA used to protect a private key from unauthorized access and/or
1448 use.</documentation>
1449         </annotation>
1450         <complexType>
1451             <sequence>
1452                 <element ref="AC:KeyActivation" minOccurs="0"/>
1453                 <element ref="AC:KeyStorage" minOccurs="0"/>
1454                 <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1455 processContents="lax" />
1456             </sequence>
1457         </complexType>
1458     </element>
1459     <element name="KeyActivation">
1460         <annotation>
1461             <documentation>The actions that must be performed before the private
1462 key can be used. </documentation>
1463         </annotation>
1464         <complexType>
1465             <choice>
```

```
1466         <element ref="AC:Password"/>
1467     </choice>
1468     </complexType>
1469 </element>
1470 <element name="KeyStorage">
1471     <annotation>
1472         <documentation>In which medium is the private key stored.
1473
1474         memory - the private key is stored in memory.
1475
1476         smartcard - the private key is stored in a smartcard.
1477
1478         token - the private key is stored in a hardware token.
1479
1480         MobileAuthCard - the private key is stored in a mobile authentication
1481 card (e.g., SIM card).
1482     </documentation>
1483     </annotation>
1484     <complexType>
1485         <attribute name="medium" use="required">
1486             <simpleType>
1487                 <restriction base="NMTOKEN">
1488                     <enumeration value="memory"/>
1489                     <enumeration value="smartcard"/>
1490                     <enumeration value="token"/>
1491                     <enumeration value="MobileAuthCard"/>
1492                 </restriction>
1493             </simpleType>
1494         </attribute>
1495     </complexType>
1496 </element>
1497
1498
1499 <element name="Password">
1500     <annotation>
1501         <documentation>This element indicates that a password (or PIN or
1502 passphrase) has been used to authenticate the Principal or
1503         to gain access to some resource (for example, to gain access to the
1504 private key).</documentation>
1505     </annotation>
1506     <complexType>
1507         <sequence>
1508             <element ref="AC:Length" minOccurs="0"/>
1509             <element ref="AC:Generation" minOccurs="0"/>
1510             <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1511 processContents="lax" />
1512         </sequence>
1513     </complexType>
1514 </element>
1515 <element name="Token">
1516     <annotation>
1517         <documentation>This element indicates that a hardware or software
1518 token is
1519         used as a method of identifying the Principal.</documentation>
1520     </annotation>
1521     <complexType>
1522         <sequence>
1523             <element ref="AC:TimeSyncToken"/>
1524             <any namespace="##any" minOccurs="0"
1525 maxOccurs="unbounded" processContents="lax" />
1526         </sequence>
1527     </complexType>
1528 </element>
1529 <element name="TimeSyncToken">
1530     <annotation>
1531         <documentation>This element indicates that a time synchronization
1532 token is used to identify the Principal.
1533
```



```
1534         hardware - the time synchronization token has been implemented in
1535 hardware.
1536
1537         software - the time synchronization token has been implemented in
1538 software.
1539
1540         SeedLength - the length, in bits, of the random seed used in the time
1541 synchronization token.
1542         </documentation>
1543     </annotation>
1544     <complexType>
1545         <attribute name="DeviceType" use="required">
1546             <simpleType>
1547                 <restriction base="NMTOKEN">
1548                     <enumeration value="hardware"/>
1549                     <enumeration value="software"/>
1550                 </restriction>
1551             </simpleType>
1552         </attribute>
1553         <attribute name="SeedLength" type="integer" use="required"/>
1554         <attribute name="DeviceInHand" use="required">
1555             <simpleType>
1556                 <restriction base="NMTOKEN">
1557                     <enumeration value="true"/>
1558                     <enumeration value="false"/>
1559                 </restriction>
1560             </simpleType>
1561         </attribute>
1562     </complexType>
1563 </element>
1564 <element name="Smartcard">
1565     <annotation>
1566         <documentation>This element indicates that a smartcard is used to
1567 identity the Principal.</documentation>
1568     </annotation>
1569     <complexType><sequence><any namespace="##any" minOccurs="0"
1570 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1571 </element>
1572 <element name="Length">
1573     <annotation>
1574         <documentation>This element indicates the minimum and/or maximum ASCII
1575 length of the password which is enforced (by the UA or the IdP). In
1576 other words,
1577         this is the minimum and/or maximum number of ASCII characters required
1578 to represent a valid password.
1579
1580         min - the minimum number of ASCII characters required in a valid
1581 password, as enforced by the UA or the IdP.
1582
1583         max - the maximum number of ASCII characters required in a valid
1584 password, as enforced by the UA or the IdP.
1585     </documentation>
1586 </annotation>
1587     <complexType>
1588         <attribute name="min" type="integer" use="required"/>
1589         <attribute name="max" type="integer" use="optional"/>
1590     </complexType>
1591 </element>
1592 <element name="Generation">
1593     <annotation>
1594         <documentation>Indicates whether the password was chosen by the
1595 Principal or auto-supplied by the identity provider.
1596
1597         principalchosen - the Principal is allowed to choose the value of the
1598 password. This is true even if the initial password is chosen at
1599 random by the UA or the IdP and the Principal is then free to change
1600 the password.
1601
```

```
1602         automatic - the password is chosen by the UA or the IdP to be
1603 cryptographically strong in some sense, or to satisfy certain
1604         password rules, and that the Principal is not free to change it or to
1605 choose a new password.
1606
1607         </documentation>
1608     </annotation>
1609     <complexType>
1610         <attribute name="mechanism" use="required">
1611             <simpleType>
1612                 <restriction base="NMTOKEN">
1613                     <enumeration value="principalchosen"/>
1614                     <enumeration value="automatic"/>
1615                 </restriction>
1616             </simpleType>
1617         </attribute>
1618     </complexType>
1619 </element>
1620 <element name="AuthenticationMethod">
1621     <annotation>
1622         <documentation>Refers to those characteristics that define the
1623 mechanisms by which the Principal authenticates to the identity provider.</documentation>
1624     </annotation>
1625     <complexType>
1626         <sequence>
1627             <element ref="AC:PrincipalAuthenticationMechanism"/>
1628             <element ref="AC:Authenticator" minOccurs="0"/>
1629             <element ref="AC:AuthenticatorTransportProtocol"/>
1630             <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1631 processContents="lax" />
1632         </sequence>
1633     </complexType>
1634 </element>
1635 <element name="PrincipalAuthenticationMechanism">
1636     <annotation>
1637         <documentation>The method that a Principal employs to perform
1638 authentication to local system components.</documentation>
1639     </annotation>
1640     <complexType>
1641         <choice minOccurs="0" maxOccurs="unbounded">
1642             <element ref="AC>Password"/>
1643             <element ref="AC:Token"/>
1644             <element ref="AC:Smartcard"/>
1645         </choice>
1646     </complexType>
1647 </element>
1648 <element name="Authenticator">
1649     <annotation>
1650         <documentation>The method applied to validate a principal's
1651 authentication across a network </documentation>
1652     </annotation>
1653     <complexType>
1654         <choice minOccurs="0" maxOccurs="unbounded">
1655             <element ref="AC:PreviousSession"/>
1656             <element ref="AC:Dig-sig"/>
1657             <element ref="AC:ZeroKnowledge"/>
1658             <element ref="AC:SharedSecretChallengeResponse"/>
1659         </choice>
1660     </complexType>
1661 </element>
1662 <element name="PreviousSession">
1663     <annotation>
1664         <documentation>Indicates that the Principal has been strongly
1665 authenticated in a previous session during which
1666         the IdP has set a cookie in the UA. During the present session the
1667 Principal has only been authenticated by
1668         the UA returning the cookie to the IdP.</documentation>
1669     </annotation>
```

```
1670     <complexType><sequence><any namespace="##any" minOccurs="0"
1671 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1672     </element>
1673     <element name="ZeroKnowledge">
1674         <annotation>
1675             <documentation>This element indicates that the Principal has been
1676 authenticated by a zero knowledge
1677 technique as specified in ISO/IEC 9798-5.</documentation>
1678         </annotation>
1679         <complexType><sequence><any namespace="##any" minOccurs="0"
1680 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1681     </element>
1682     <element name="SharedSecretChallengeResponse">
1683         <annotation>
1684             <documentation>This element indicates that the Principal has been
1685 authenticated by a challenge-response
1686 protocol utilizing shared secret keys and symmetric
1687 cryptography.</documentation>
1688         </annotation>
1689         <complexType><sequence><any namespace="##any" minOccurs="0"
1690 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1691     </element>
1692     <element name="Dig-sig">
1693         <annotation>
1694             <documentation>This element indicates that the Principal has been
1695 authenticated by a mechanism which involves the Principal
1696 computing a digital signature over at least challenge data provided by
1697 the IdP.</documentation>
1698         </annotation>
1699         <complexType><sequence><any namespace="##any" minOccurs="0"
1700 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1701     </element>
1702     <element name="AuthenticatorTransportProtocol">
1703         <annotation>
1704             <documentation>The protocol across which Authenticator information is
1705 transferred to an identity provider verifier.</documentation>
1706         </annotation>
1707         <complexType>
1708             <choice minOccurs="0" maxOccurs="unbounded">
1709                 <element ref="AC:HTTP"/>
1710                 <element ref="AC:SSL"/>
1711                 <element ref="AC:MobileNetwork"/>
1712                 <element ref="AC:WTLS"/>
1713                 <element ref="AC:IPSec"/>
1714             </choice>
1715         </complexType>
1716     </element>
1717     <element name="HTTP">
1718         <annotation>
1719             <documentation>This element indicates that the Authenticator has been
1720 transmitted
1721 using bare HTTP utilizing no additional security
1722 protocols.</documentation>
1723         </annotation>
1724         <complexType><sequence><any namespace="##any" minOccurs="0"
1725 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1726     </element>
1727     <element name="IPSec">
1728         <annotation>
1729             <documentation>This element indicates that the Authenticator has been
1730 transmitted
1731 using a transport mechanism protected by an IPSEC
1732 session.</documentation>
1733         </annotation>
1734         <complexType><sequence><any namespace="##any" minOccurs="0"
1735 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1736     </element>
1737     <element name="WTLS">
```

```
1738         <annotation>
1739             <documentation>This element indicates that the Authenticator has been
1740 transmitted
1741             using a transport mechanism protected by a WTLS
1742 session.</documentation>
1743         </annotation>
1744         <complexType><sequence><any namespace="##any" minOccurs="0"
1745 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1746     </element>
1747     <element name="MobileNetwork">
1748         <annotation>
1749             <documentation>This element indicates that the Authenticator has been
1750 transmitted
1751             solely across a mobile network using no additional security
1752 mechanism.</documentation>
1753         </annotation>
1754         <complexType><sequence><any namespace="##any" minOccurs="0"
1755 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1756     </element>
1757     <element name="SSL">
1758         <annotation>
1759             <documentation>This element indicates that the Authenticator has been
1760 transmitted
1761             using a transport mechanism protected by an SSL or TLS
1762 session.</documentation>
1763         </annotation>
1764         <complexType><sequence><any namespace="##any" minOccurs="0"
1765 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1766     </element>
1767     <element name="OperationalProtection">
1768         <annotation>
1769             <documentation>Refers to those characteristics that describe
1770 procedural security controls employed by the identity provider.</documentation>
1771         </annotation>
1772         <complexType>
1773             <sequence>
1774                 <element ref="AC:SecurityAudit" minOccurs="0"/>
1775                 <element ref="AC:DeactivationCallCenter" minOccurs="0"/>
1776                 <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1777 processContents="lax" />
1778             </sequence>
1779         </complexType>
1780     </element>
1781     <element name="SecurityAudit">
1782         <complexType>
1783             <sequence>
1784                 <element ref="AC:SwitchAudit" minOccurs="0"/>
1785                 <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1786 processContents="lax" />
1787             </sequence>
1788         </complexType>
1789     </element>
1790     <element name="SwitchAudit">
1791         <complexType><sequence><any namespace="##any" minOccurs="0"
1792 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1793     </element>
1794     <element name="DeactivationCallCenter">
1795         <complexType><sequence><any namespace="##any" minOccurs="0"
1796 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1797     </element>
1798     <element name="GoverningAgreements">
1799         <annotation>
1800             <documentation>Provides a mechanism for linking to external (likely
1801 human readable) documents in which the identity provider can define business level
1802 authentication context, e.g. liability constraints, contractual
1803 obligations.</documentation>
1804         </annotation>
1805     </complexType>
```

```
1806         <sequence>
1807             <element ref="AC:GoverningAgreementRef"/>
1808         </sequence>
1809     </complexType>
1810 </element>
1811 <element name="GoverningAgreementRef">
1812     <complexType>
1813         <attribute name="governingAgreementRef" type="anyURI" use="required"/>
1814     </complexType>
1815 </element>
1816 </schema>
```

1817 6 References

- 1818 [LibertyGloss] H. Mauldin, "Liberty Glossary," <http://projectliberty.org/specs/liberty-tech-glossary-v1.0.pdf>, October 2002.
- 1819
- 1820 [LibertyProtSchema] J.Beatty, "Liberty Protocols and Schemas Specification," <http://projectliberty.org/specs/liberty-architecture-protocols-schemas-v1.0.pdf>, October 2002.
- 1821
- 1822
- 1823 [PDS] S. Santesson et al, "Internet X.509 Public Key Infrastructure PKI Disclosure Statement," <http://www.verisign.com/repository/pds.txt>.
- 1824
- 1825 [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.
- 1826
- 1827 [RFC2527] S. Chokhani et al, "Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework," <http://www.ietf.org/rfc/rfc2527.txt?number=2527>.
- 1828
- 1829
- 1830 [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.
- 1831
- 1832
- 1833
- 1834 [SAMLCore] P. Hallam-Baker 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.
- 1835
- 1836
- 1837
- 1838 [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.
- 1839
- 1840
- 1841