



1

2

Liberty Authentication Context Specification

3

Version 1.1

4

5

15 January 2003

6

7

8

Document Description: liberty-architecture-authentication-context-v1.1

9

10

11 **Notice**

12 Copyright © 2002,2003 ActivCard; American Express Travel Related Services; America Online,
13 Inc.; Bank of America; Bell Canada; Catavault; Cingular Wireless; Cisco Systems, Inc.; Citigroup;
14 Communicator, Inc.; Consignia; Cyberun Corporation; Deloitte & Touche LLP; Earthlink, Inc.;
15 Electronic Data Systems, Inc.; Entrust, Inc.; Ericsson; Fidelity Investments; France Telecom;
16 Gemplus; General Motors; Hewlett-Packard Company; i2 Technologies, Inc.; Intuit Inc.;
17 MasterCard International; NEC Corporation; Netegrity; NeuStar; Nextel Communications; Nippon
18 Telegraph and Telephone Company; Nokia Corporation; Novell, Inc.; NTT DoCoMo, Inc.;
19 OneName Corporation; Openwave Systems Inc.; PricewaterhouseCoopers LLP; Register.com;
20 RSA Security Inc; Sabre Holdings Corporation; SAP AG; SchlumbergerSema; SK Telecom; Sony
21 Corporation; Sun Microsystems, Inc.; United Airlines; VeriSign, Inc.; Visa International;
22 Vodafone Group Plc; Wave Systems;. All rights reserved.

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

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

39 Liberty Alliance Project
40 Licensing Administrator
41 c/o IEEE-ISTO
42 445 Hoes Lane
43 Piscataway, NJ 08855-1331, USA
44 info@projectliberty.org

45 **Editors**

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

47 John Kemp, IEEE-ISTO

48 **Contributors**

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

49

50 **Revision 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 |

| | | | |
|-----------|--------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | | <p>requesting a 'better' class</p> <ul style="list-style-type: none"> - new submitted classes |
| 03 | 16-Apr-02 | Paul Madsen | <p>To reflect</p> <ul style="list-style-type: none"> - Minor edits - John Beatty's schema edits - New AuthenticationContextStatement element |
| 04 | April 29, 02 | Paul Madsen | <p>Edited after Paris meetings to include</p> <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 |
| 08 | Dec 18, 2002 | John Kemp | <ul style="list-style-type: none"> • Updated references • Updated legal text • Inserted new logo |
| 1.1 Final | Jan 15, 2003 | John Kemp | <ul style="list-style-type: none"> • Corrected typos • Revised SAML namespaces |

51 **Table of Contents**

52 1 Introduction 6
53 1.1 Notation 6
54 2 Overview 7
55 3 Authentication Context 7
56 3.1 Authentication Context Classes 8
57 3.2 Authentication Quality 10
58 3.2.1 Service Provider Request 10
59 3.2.2 Identity Provider Response 11
60 4 Previous work 11
61 4.1 PKI 11
62 4.2 SAML 12
63 5 Liberty Authentication Context Mechanisms 13
64 5.1 Authentication Context Classes 13
65 5.1.1 MobileContract 13
66 5.1.2 MobileDigitalID 15
67 5.1.3 MobileUnregistered 18
68 5.1.4 Password 20
69 5.1.5 Password- ProtectedTransport 21
70 5.1.6 Previous-Session 22
71 5.1.7 Smartcard 24
72 5.1.8 Smartcard-PKI 24
73 5.1.9 Software-PKI 26
74 5.1.10 Time-Sync-Token 28
75 5.2 Authentication Context Schema 29
76 5.2.1 XML Schema 29
77 6 References 37
78

79 1 Introduction

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

82 1.1 Notation

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

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

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

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

97
98 Listings of XML schemas appear like this.

99
100 Example code listings appear like this.

101

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

- 105 • The prefix lib: stands for the Liberty namespace
106 (<http://projectliberty.org/schemas/core/2002/12>)
- 107 • The prefix saml: stands for the SAML assertion namespace
108 (<urn:oasis:names:tc:SAML:1.0:assertion>).
- 109 • The prefix samlp: stands for the SAML request-response protocol namespace
110 (<urn:oasis:names:tc:SAML:1.0:protocol>).
- 111 • The prefix ds: stands for the W3C XML signature namespace
112 (<http://www.w3.org/2000/09/xmldsig#>).
- 113 • The prefix xsd: stands for the W3C XML schema namespace in example listings
114 (<http://www.w3.org/2001/XMLSchema>). In schema listings, this namespace is the
115 default, and no prefix is shown.

116 This specification uses the following typographical conventions in text: <Element>,
117 <ns:ForeignElement>, Attribute, **Datatype**, OtherCode.

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

119 **2 Overview**

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

136
137 *Authentication context* is defined as the information additional to the authentication assertion itself
138 that the service provider may require before it makes an entitlements decision.

139 **3 Authentication Context**

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

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

149
150 The variations and permutations in the examples above guarantee that not all authentication
151 assertions are the same; a particular authentication assertion will be characterized by the values for
152 each of these variables. A somewhat helpful model is to think of an authentication assertion as
153 defined by its coordinates in a multidimensional space. This model is demonstrated in Figure 1
154 (where only three axes are shown).
155
156

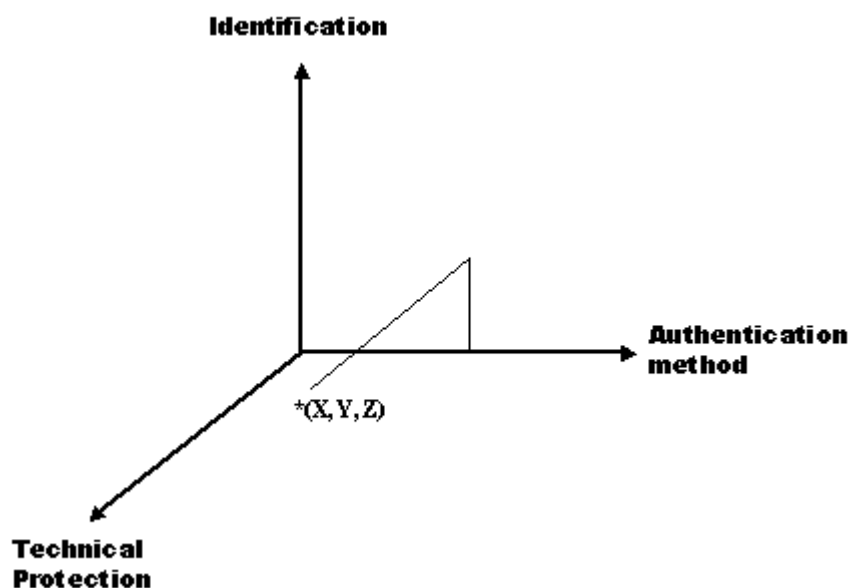


Figure 1: Authentication assertion as defined by its coordinates in multidimensional space

157
158
159
160
161

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

162
163
164
165
166
167
168
169
170
171
172
173

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.

174
175
176
177
178

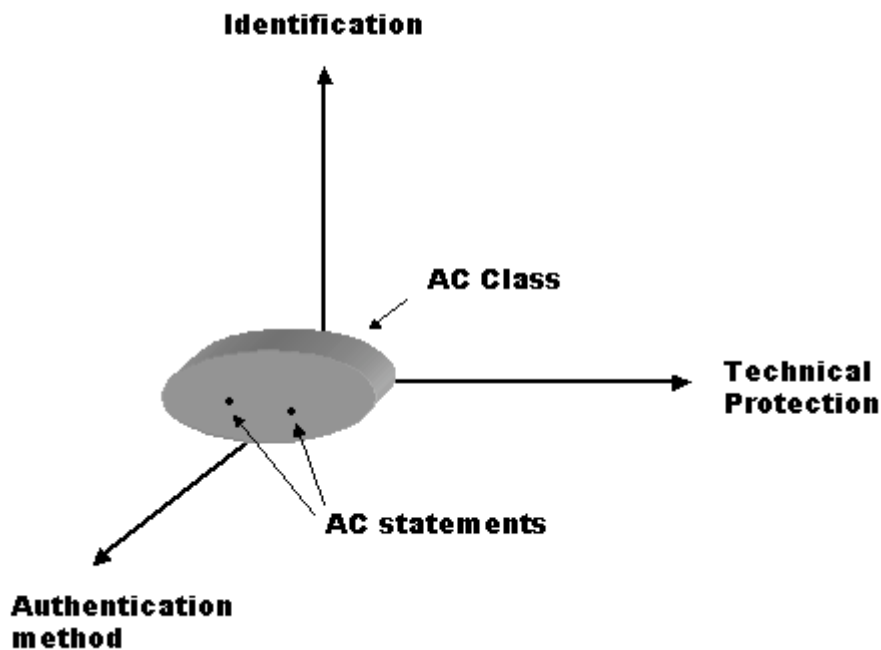
A particular Liberty authentication context class will define a list of required characteristics of the processes, procedures, and mechanisms by which the identity provider verifies the Principal before issuing an identity, protects the secrets on which subsequent authentications are based, and the mechanisms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

179
180
181
182

- **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.

- 183 • **Physical Protection** – Characteristics that specify physical controls on the facility housing
184 the identity provider’s systems (for example, site location and construction, access
185 controls).
- 186 • **Operational Protection** – Characteristics that describe procedural security controls
187 employed by the identity provider (for example, security audits, records archival).
- 188 • **Technical Protection** – Characteristics that describe how the “secret” (the knowledge or
189 possession of which allows the Principal to authenticate to the identity provider) is kept
190 secure.
- 191 • **Authentication Method** – Characteristics that define the mechanisms by which the
192 Principal authenticates to the identity provider (for example, a password versus a
193 smartcard).

194 Rather than a class being a rigid collection of these characteristics, a class will define a set of
195 conformant authentication context statements (for example, multiple and different authentication
196 context statements will satisfy the requirements of a given class). The relationship between an
197 authentication context class and particular authentication context statements is shown in Figure 2,
198 where all the authentication context statements satisfy the requirements expressed by the class.
199



200
201 **Figure 2: Relationship between authentication context class and statements**

202
203 By introducing the additional layer of classes and by defining an initial list of representative and
204 flexible classes, Liberty architecture

- 205
- 206 • Makes it easier for the identity provider and service provider to come to an agreement on
207 what are acceptable authentication contexts by giving them a framework for discussion.

- 208 • Makes it easier for service providers to indicate their preferences when requesting a step-
209 up authentication assertion from an identity provider.
- 210 • Simplifies for service providers the burden of processing authentication context statements
211 by giving them the option of being satisfied by the associated class.
- 212 • Protects service providers from impact of new authentication technologies.
- 213 • Makes it easier for identity providers to publish their authentication capabilities, for
214 example, through WSDL.

215 **3.2 Authentication Quality**

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

223 **3.2.1 Service Provider Request**

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

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

235
236 Option 1 will require that the identity provider and service provider have previously agreed on the
237 details of a particular authentication context that either does not fall into one of the Liberty-defined
238 authentication context classes or needs to be constrained more tightly.

239
240 Option 2 is expected to be the typical scenario.

241
242 For option 3, the decision as to what is better is left to the entity best qualified to make that
243 determination, the identity provider. The service provider, trusting the identity provider's
244 judgment, will accept the assertion it receives back because it will be confident the assertion meets
245 (or exceeds) the provider's requirements.
246

247 Option 4 will give the service provider greater control over the authentication context classes to
248 which the authentication assertions it receives conform. The identity provider is given no leeway
249 in providing an authentication assertion conforming to a class not on the list.
250

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

254 **3.2.2 Identity Provider Response**

255 The authentication assertion that the identity provider returns to the service provider may indicate
256 the authentication context class to which the authentication assertion conforms (if it does conform
257 to any such authentication context class), which may or may not be the same as the class
258 requested.
259

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

262 **4 Previous work**

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

264 **4.1 PKI**

265 An X.509 certificate is a signed assertion of identity just as a SAML authentication assertion is.
266 Consequently it is not surprising that the issue of authentication context has been addressed within
267 the PKI world. A number of different standards or proposals for capturing this sort of information
268 have been written:
269

- 270 • **Certificate Practice Statement (CPS)** is a statement of the practices that a certification
271 authority employs in issuing certificates. A certificate practice statement may take the
272 form of a declaration by the certification authority of the details of its trustworthy systems
273 and the practices it employs in support of its issuance of certificates.
- 274 • **Certificate Policy** is a named set of rules that indicates the applicability of a certificate to
275 a particular community and/or class of application. For example, a certificate policy might
276 indicate that a particular type of certificate is appropriate for the authentication of
277 participants in a business-to-business transaction within a given price range. The
278 fundamental difference between the certificate practice statement and the certificate policy
279 is that the former is “owned” by the issuing certification authority and the latter by the
280 entities who will use the issued certificates. Certificate users define certificate policies, and
281 certification authorities (with different certificate practice statements) attest that a
282 particular certificate is appropriate for that certificate policy. (See [[RFC2527](#)].)
- 283 • **PKI Disclosure Statement** is a supplementary instrument that discloses critical
284 information about the policies and practices of a certificate authority or PKI. A PKI
285 disclosure statement is a vehicle for disclosing and emphasizing information normally
286 covered in detail by associated certificate policy and/or certification practice statement
287 documents. Consequently, a PKI disclosure statement is not intended to replace a
288 certificate policy or practice statement. (See [[PDS](#)].)

- 289
- **Key Usage**, as defined in X.509, defines the intended use for a key contained in a certificate. These uses (or *values*) are digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment, keyAgreement, keyCertSign, CRLSign, encipherOnly, and decipherOnly.
- 290
- **Extended Key Usage**, as the name indicates, extends the possible uses for a key beyond the original nine, each use identified by an object identifier. Extended key usage is primarily used by the relying party. As part of its validation algorithm, a relying party will check for these values to determine whether a given certificate is appropriate for the application.
- 291
- 292
- 293
- 294
- 295
- 296
- 297

298 4.2 SAML

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

```
303  
304 <?xml version="1.0"?>  
305 <saml:Assertion>  
306 <saml:AuthenticationStatement AuthenticationMethod=" urn:ietf:rfc:2246">  
307 <saml:Subject>  
308 <saml:NameIdentifier  
309 Format="http://www.oasis-open.org/committees/security/docs/cs-sstc-core-  
310 28#X509SubjectName">cn=Joe User,dc=projectliberty,dc=org  
311 </saml:NameIdentifier>  
312 </saml:Subject>  
313 </saml:AuthenticationStatement>  
314 <saml:Advice>  
315 <!--additional elements in separate namespace -->  
316 </saml:Advice>  
317 </saml:Assertion>
```

318

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

- 321
- <saml:Condition> [Optional]. Conditions that MUST be taken into account in assessing the validity of the assertion.
 - <saml:Advice> [Optional]. Additional information related to the assertion that assists processing in certain situations, but MAY be ignored by applications that do not support its use.
- 322
- 323
- 324
- 325
- 326

327

328 The intent seems to be that the <saml:Condition> element protects the issuing party, and the
329 <saml:Advice> element protects the relying party.

330

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

336
337 SAML identified a list of common authentication protocols as possible values for both the
338 AuthenticationMethod attribute and the <saml:SubjectConfirmation> element,
339 including SAML Artifact, Holder of Key, Sender Vouches, Password, Kerberos, and SSL/TLS.

340 **5 Liberty Authentication Context Mechanisms**

341 **5.1 Authentication Context Classes**

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

343
344 The classes are listed in alphabetical order, no ranking is implied.

345
346 Classes are identified by URIs with the initial stem:

347
348 <http://www.projectliberty.org/schemas/authctx/classes>
349

350 **5.1.1 MobileContract**

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

353 **5.1.1.1 Associated Liberty URI**

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

355 **5.1.1.2 Class Schema**

```
356 <?xml version="1.0" encoding="UTF-8"?>
357
358 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
359
360 <annotation>
361 <documentation> http://www.projectliberty.org/schemas/authctx/classes/MobileContract
362 </documentation>
363 </annotation>
364
365 <xs:element name="AuthenticationContextStatement">
366 <xs:complexType>
367 <xs:sequence>
368 <xs:element minOccurs="1" maxOccurs="1"
369 ref="Identification"/>
370 <xs:element minOccurs="1" maxOccurs="1"
371 ref="TechnicalProtection"/>
372 <xs:element minOccurs="1" maxOccurs="1"
373 ref="AuthenticationMethod"/>
374 <xs:element minOccurs="1" maxOccurs="1"
375 ref="OperationalProtection"/>
376 <xs:element minOccurs="1" maxOccurs="1"
377 ref="GoverningAgreements"/>
378 <xs:any namespace="##any" minOccurs="0"
379 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
380 </xs:complexType>
381 </xs:element>
382 <xs:element name="AuthenticationMethod">
383 <xs:complexType>
```

```

384         <xs:sequence>
385             <xs:element minOccurs="1" maxOccurs="1" ref="Authenticator"/>
386             <xs:element minOccurs="1" maxOccurs="1"
387 ref="AuthenticatorTransportProtocol"/>
388             <xs:any namespace="##any" minOccurs="0"
389 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
390         </xs:complexType>
391     </xs:element>
392
393     <xs:element name="Authenticator">
394         <xs:complexType>
395             <xs:sequence>
396                 <xs:element minOccurs="1" maxOccurs="1"
397 ref="SharedSecretChallengeResponse"/>
398                 <xs:any namespace="##any" minOccurs="0"
399 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
400             </xs:complexType>
401         </xs:element>
402
403     <xs:element name="AuthenticatorTransportProtocol">
404         <xs:complexType>
405             <xs:sequence>
406                 <xs:element minOccurs="1" maxOccurs="1" ref="MobileNetwork"/>
407                 <xs:any namespace="##any" minOccurs="0"
408 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
409             </xs:complexType>
410         </xs:element>
411
412     <xs:element name="DeactivationCallCenter">
413         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
414 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
415     </xs:element>
416
417     <xs:element name="GoverningAgreementRef">
418         <xs:complexType>
419             <xs:attribute name="ref"
420 fixed="http://SomeMobileforum.org/namespaces/authcontext/classes/Mobile-Class2.pdf"/>
421         </xs:complexType>
422     </xs:element>
423     <xs:element name="GoverningAgreements">
424         <xs:complexType>
425             <xs:sequence>
426                 <xs:element minOccurs="1" maxOccurs="1"
427 ref="GoverningAgreementRef"/>
428                 <xs:any namespace="##any" minOccurs="0"
429 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
430             </xs:complexType>
431         </xs:element>
432     <xs:element name="Identification">
433         <xs:complexType>
434             <xs:sequence>
435                 <xs:element minOccurs="1" maxOccurs="1"
436 ref="PhysicalVerification"/>
437                 <xs:any namespace="##any" minOccurs="0"
438 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
439                 <xs:attribute name="nym" type="xs:string" use="required"/>
440             </xs:complexType>
441         </xs:element>
442     <xs:element name="MobileAuthCard">
443         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
444 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
445     </xs:element>
446     <xs:element name="MobileDevice">
447         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
448 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>

```

```

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

```

507 5.1.2 MobileDigitalID

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

510 5.1.2.1 Associated Liberty URI

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

512 5.1.2.2 Class Schema

```
513 <?xml version="1.0" encoding="UTF-8"?>
514
515 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
516
517 <annotation>
518 <documentation> http://www.projectliberty.org/schemas/authctx/classes/MobileDigitalID
519 </documentation>
520 </annotation>
521
522
523 <xs:element name="AuthenticationContextStatement">
524 <xs:complexType>
525 <xs:sequence>
526 <xs:element ref="Identification"/>
527 <xs:element ref="TechnicalProtection"/>
528 <xs:element ref="AuthenticationMethod"/>
529 <xs:element ref="OperationalProtection"/>
530 <xs:element ref="GoverningAgreements"/>
531 <xs:any namespace="##any" minOccurs="0"
532 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
533 </xs:complexType>
534 </xs:element>
535 <xs:element name="AuthenticationMethod">
536 <xs:complexType>
537 <xs:sequence>
538 <xs:element ref="Authenticator"/>
539 <xs:element ref="AuthenticatorTransportProtocol"/>
540 <xs:any namespace="##any" minOccurs="0"
541 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
542 </xs:complexType>
543 </xs:element>
544 <xs:element name="Authenticator">
545 <xs:complexType>
546 <xs:choice>
547 <xs:element ref="Dig-sig"/>
548 <xs:element ref="ZeroKnowledge"/>
549 </xs:choice>
550 </xs:complexType>
551 </xs:element>
552 <xs:element name="AuthenticatorTransportProtocol">
553 <xs:complexType>
554 <xs:choice>
555 <xs:element ref="MobileNetwork"/>
556 <xs:element ref="SSL"/>
557 <xs:element ref="WTLS"/>
558 <xs:element ref="IPSec"/>
559 </xs:choice>
560 </xs:complexType>
561 </xs:element>
562 <xs:element name="DeactivationCallCenter">
563 <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
564 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
565 </xs:element>
566 <xs:element name="Dig-sig">
567 <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
568 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
569 </xs:element>
```



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

```
635     <xs:complexType>
636         <xs:sequence>
637             <xs:element ref="SwitchAudit"/>
638             <xs:any namespace="##any" minOccurs="0"
639 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
640         </xs:complexType>
641     </xs:element>
642     <xs:element name="SwitchAudit">
643         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
644 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
645     </xs:element>
646     <xs:element name="TechnicalProtection">
647         <xs:complexType>
648             <xs:sequence>
649                 <xs:element ref="PrivateKeyProtection"/>
650                 <xs:any namespace="##any" minOccurs="0"
651 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
652             </xs:complexType>
653         </xs:element>
654     <xs:element name="WTLS">
655         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
656 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
657     </xs:element>
658     <xs:element name="WrittenConsent">
659         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
660 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
661     </xs:element>
662     <xs:element name="ZeroKnowledge">
663         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
664 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
665     </xs:element>
666 </xs:schema>
```

667 **5.1.3 MobileUnregistered**

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

670 **5.1.3.1 Associated Liberty URI**

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

672 **5.1.3.2 Class Schema**

```
673 <?xml version="1.0" encoding="UTF-8"?>
674 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
675 <annotation>
676 <documentation> http://www.projectliberty.org/schemas/authctx/classes/MobileUnregistered
677 </documentation>
678 </annotation>
679 <xs:element name="AuthenticationContextStatement">
680     <xs:complexType>
681         <xs:sequence>
682             <xs:element ref="TechnicalProtection"/>
683             <xs:element ref="AuthenticationMethod"/>
684             <xs:element ref="OperationalProtection"/>
685             <xs:element ref="GoverningAgreements"/>
686         </xs:sequence>
687     </xs:complexType>
688 </xs:element>
689 </xs:schema>
```

```

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

```

```

754         <xs:any namespace="##any" minOccurs="0"
755 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
756     </xs:complexType>
757 </xs:element>
758 <xs:element name="SecurityAudit">
759     <xs:complexType>
760         <xs:sequence>
761             <xs:element ref="SwitchAudit"/>
762             <xs:any namespace="##any" minOccurs="0"
763 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
764         </xs:complexType>
765     </xs:element>
766 <xs:element name="SharedKeyProtection">
767     <xs:complexType>
768         <xs:choice>
769             <xs:element ref="MobileAuthCard"/>
770             <xs:element ref="MobileDevice"/>
771         </xs:choice>
772     </xs:complexType>
773 </xs:element>
774 <xs:element name="SharedSecretChallengeResponse">
775     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
776 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
777 </xs:element>
778 <xs:element name="SwitchAudit">
779     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
780 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
781 </xs:element>
782 <xs:element name="TechnicalProtection">
783     <xs:complexType>
784         <xs:sequence>
785             <xs:element ref="SharedKeyProtection"/>
786             <xs:any namespace="##any" minOccurs="0"
787 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
788         </xs:complexType>
789     </xs:element>
790 </xs:schema>

```

791 **5.1.4 Password**

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

794 **5.1.4.1 Associated Liberty URI**

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

796 **5.1.4.2 Class Schema**

```

797 <?xml version="1.0" encoding="UTF-8"?>
798
799 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
800
801 <annotation>
802 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Password
803 </documentation>
804 </annotation>
805
806     <xs:element name="AuthenticationContextStatement">
807         <xs:complexType>
808             <xs:sequence>
809                 <xs:element ref="AuthenticationMethod"/>

```

```

810         <xs:any namespace="##any" minOccurs="0"
811 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
812     </xs:complexType>
813 </xs:element>
814 <xs:element name="AuthenticationMethod">
815     <xs:complexType>
816         <xs:all>
817             <xs:element ref="PrincipalAuthenticationMechanism"/>
818             <xs:element ref="AuthenticatorTransportProtocol"/>
819         </xs:all>
820     </xs:complexType>
821 </xs:element>
822 <xs:element name="AuthenticatorTransportProtocol">
823     <xs:complexType>
824         <xs:sequence>
825             <xs:element ref="HTTP"/>
826             <xs:any namespace="##any" minOccurs="0"
827 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
828         </xs:complexType>
829     </xs:element>
830 <xs:element name="HTTP">
831     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
832 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
833 </xs:element>
834 <xs:element name="Length">
835     <xs:complexType>
836         <xs:attribute name="min" fixed="3"/>
837     </xs:complexType>
838 </xs:element>
839 <xs:element name="Password">
840     <xs:complexType>
841         <xs:sequence>
842             <xs:element ref="Length"/>
843             <xs:any namespace="##any" minOccurs="0"
844 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
845         </xs:complexType>
846     </xs:element>
847 <xs:element name="PrincipalAuthenticationMechanism">
848     <xs:complexType>
849         <xs:sequence>
850             <xs:element ref="Password"/>
851             <xs:any namespace="##any" minOccurs="0"
852 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
853         </xs:complexType>
854     </xs:element>
855 </xs:schema>

```

856 5.1.5 Password- ProtectedTransport

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

859 5.1.5.1 Associated Liberty URI

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

861 5.1.5.2 Class Schema

```

862 <?xml version="1.0" encoding="UTF-8"?>
863 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
864
865
866

```

```

867 <annotation>
868 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Password-
869 ProtectedTransport </documentation>
870 </annotation>
871
872     <xs:element name="AuthenticationContextStatement">
873         <xs:complexType>
874             <xs:sequence>
875                 <xs:element ref="AuthenticationMethod"/>
876                 <xs:any namespace="##any" minOccurs="0"
877 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
878             </xs:complexType>
879         </xs:element>
880     <xs:element name="AuthenticationMethod">
881         <xs:complexType>
882             <xs:all>
883                 <xs:element ref="PrincipalAuthenticationMechanism"/>
884                 <xs:element ref="AuthenticatorTransportProtocol"/>
885             </xs:all>
886         </xs:complexType>
887     </xs:element>
888     <xs:element name="AuthenticatorTransportProtocol">
889         <xs:complexType>
890             <xs:sequence>
891                 <xs:element ref="SSL"/>
892                 <xs:any namespace="##any" minOccurs="0"
893 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
894             </xs:complexType>
895         </xs:element>
896         <xs:element name="Length">
897             <xs:complexType>
898                 <xs:attribute name="min" fixed="3"/>
899             </xs:complexType>
900         </xs:element>
901         <xs:element name="Password">
902             <xs:complexType>
903                 <xs:sequence>
904                     <xs:element ref="Length"/>
905                     <xs:any namespace="##any" minOccurs="0"
906 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
907                 </xs:complexType>
908             </xs:element>
909             <xs:element name="PrincipalAuthenticationMechanism">
910                 <xs:complexType>
911                     <xs:sequence>
912                         <xs:element ref="Password"/>
913                         <xs:any namespace="##any" minOccurs="0"
914 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
915                     </xs:complexType>
916                 </xs:element>
917                 <xs:element name="SSL">
918                     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
919 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
920                 </xs:element>
921 </xs:schema>

```

922 5.1.6 Previous-Session

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

928
929 The context for the previously authenticated session is explicitly not included in this context class
930 because the user has not authenticated during this session, and so the mechanism that the user
931 employed to authenticate in a previous session should not be used as part of a decision on whether
932 to *now* allow access to a resource.

933 **5.1.6.1 Associated Liberty URI**

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

935 **5.1.6.2 Class Schema**

```
936 <?xml version="1.0" encoding="UTF-8"?>
937 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
938 <annotation>
939 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Previous-Session
940 </documentation>
941 </annotation>
942
943     <xs:element name="AuthenticationContextStatement">
944         <xs:complexType>
945             <xs:sequence>
946                 <xs:element minOccurs="1" maxOccurs="1"
947 ref="AuthenticationMethod"/>
948                 <xs:any namespace="##any" minOccurs="0"
949 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
950             </xs:complexType>
951         </xs:element>
952
953     <xs:element name="AuthenticationMethod">
954         <xs:complexType>
955             <xs:sequence>
956                 <xs:element ref="Authenticator" minOccurs="0" maxOccurs="1"/>
957                 <xs:any namespace="##any" minOccurs="0"
958 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
959             </xs:complexType>
960         </xs:element>
961
962     <xs:element name="Authenticator">
963         <xs:complexType>
964             <xs:sequence>
965                 <xs:element minOccurs="1" maxOccurs="1"
966 ref="PreviousSession"/>
967                 <xs:any namespace="##any" minOccurs="0"
968 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
969             </xs:complexType>
970         </xs:element>
971
972     <xs:element name="PreviousSession">
973         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
974 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
975     </xs:element>
976 </xs:schema>
```

982 **5.1.7 Smartcard**

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

985 **5.1.7.1 Associated Liberty URI**

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

987 **5.1.7.2 Class Schema**

```
988 <?xml version="1.0" encoding="UTF-8"?>
989
990 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
991
992 <annotation>
993 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Smartcard
994 </documentation>
995 </annotation>
996
997 <xs:element name="AuthenticationContextStatement">
998 <xs:complexType>
999 <xs:sequence>
1000 <xs:element minOccurs="1" maxOccurs="1"
1001 ref="AuthenticationMethod"/>
1002 <xs:any namespace="##any" minOccurs="0"
1003 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1004 </xs:complexType>
1005 </xs:element>
1006 <xs:element name="AuthenticationMethod">
1007 <xs:complexType>
1008 <xs:sequence>
1009 <xs:element minOccurs="1" maxOccurs="1"
1010 ref="PrincipalAuthenticationMechanism"/>
1011 <xs:any namespace="##any" minOccurs="0"
1012 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1013 </xs:complexType>
1014 </xs:element>
1015 <xs:element name="PrincipalAuthenticationMechanism">
1016 <xs:complexType>
1017 <xs:sequence>
1018 <xs:element minOccurs="1" maxOccurs="1" ref="Smartcard"/>
1019 <xs:any namespace="##any" minOccurs="0"
1020 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1021 </xs:complexType>
1022 </xs:element>
1023 <xs:element name="Smartcard">
1024 <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1025 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1026 </xs:element>
1027 </xs:schema>
1028
```

1029 **5.1.8 Smartcard-PKI**

1030 The Smartcard-PKI class is identified when a Principal authenticates to an identity provider
1031 through a two-factor authentication mechanism using a smartcard with enclosed private key and a
1032 PIN.

1033 **5.1.8.1 Associated Liberty URI**

1034 <http://www.projectliberty.org/schemas/authctx/classes/Smartcard-PKI>

1035 **5.1.8.2 Class Schema**

```
1036 <?xml version="1.0" encoding="UTF-8"?>
1037
1038 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
1039
1040 <annotation>
1041 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Smartcard-
1042 PKI</documentation>
1043 </annotation>
1044
1045 <xs:element name="AuthenticationContextStatement">
1046 <xs:complexType>
1047 <xs:sequence>
1048 <xs:element minOccurs="1" maxOccurs="1"
1049 ref="TechnicalProtection"/>
1050 <xs:element minOccurs="1" maxOccurs="1"
1051 ref="AuthenticationMethod"/>
1052 <xs:any namespace="##any" minOccurs="0"
1053 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1054 </xs:complexType>
1055 </xs:element>
1056 <xs:element name="AuthenticationMethod">
1057 <xs:complexType>
1058 <xs:sequence>
1059 <xs:element minOccurs="1" maxOccurs="1"
1060 ref="PrincipalAuthenticationMechanism"/>
1061 <xs:element minOccurs="1" maxOccurs="1" ref="Authenticator"/>
1062 <xs:element minOccurs="1" maxOccurs="1"
1063 ref="AuthenticatorTransportProtocol"/>
1064 <xs:any namespace="##any" minOccurs="0"
1065 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1066 </xs:complexType>
1067 </xs:element>
1068 <xs:element name="Authenticator">
1069 <xs:complexType>
1070 <xs:sequence>
1071 <xs:element minOccurs="1" maxOccurs="1" ref="Dig-sig"/>
1072 <xs:any namespace="##any" minOccurs="0"
1073 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1074 </xs:complexType>
1075 </xs:element>
1076 <xs:element name="AuthenticatorTransportProtocol">
1077 <xs:complexType>
1078 <xs:sequence>
1079 <xs:element minOccurs="1" maxOccurs="1" ref="SSL"/>
1080 <xs:any namespace="##any" minOccurs="0"
1081 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1082 </xs:complexType>
1083 </xs:element>
1084 <xs:element name="Dig-sig">
1085 <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1086 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1087 </xs:element>
1088 <xs:element name="KeyActivation">
1089 <xs:complexType>
1090 <xs:sequence>
1091 <xs:element minOccurs="1" maxOccurs="1" ref="Password"/>
1092 <xs:any namespace="##any" minOccurs="0"
1093 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
```

```
1094         </xs:complexType>
1095     </xs:element>
1096     <xs:element name="Length">
1097         <xs:complexType>
1098             <xs:attribute name="min" type="xs:byte" use="required"/>
1099         </xs:complexType>
1100     </xs:element>
1101     <xs:element name="Password">
1102         <xs:complexType>
1103             <xs:sequence>
1104                 <xs:element minOccurs="1" maxOccurs="1" ref="Length"/>
1105                 <xs:any namespace="##any" minOccurs="0"
1106 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1107             </xs:complexType>
1108     </xs:element>
1109     <xs:element name="PrincipalAuthenticationMechanism">
1110         <xs:complexType>
1111             <xs:sequence>
1112                 <xs:element minOccurs="1" maxOccurs="1" ref="Smartcard"/>
1113                 <xs:any namespace="##any" minOccurs="0"
1114 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1115             </xs:complexType>
1116     </xs:element>
1117     <xs:element name="PrivateKeyProtection">
1118         <xs:complexType>
1119             <xs:sequence>
1120                 <xs:element minOccurs="1" maxOccurs="1" ref="KeyActivation"/>
1121                 <xs:any namespace="##any" minOccurs="0"
1122 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1123             </xs:complexType>
1124     </xs:element>
1125     <xs:element name="SSL">
1126         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1127 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1128     </xs:element>
1129     <xs:element name="Smartcard">
1130         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1131 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1132     </xs:element>
1133     <xs:element name="TechnicalProtection">
1134         <xs:complexType>
1135             <xs:sequence>
1136                 <xs:element minOccurs="1" maxOccurs="1"
1137 ref="PrivateKeyProtection"/>
1138                 <xs:any namespace="##any" minOccurs="0"
1139 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1140             </xs:complexType>
1141     </xs:element>
1142 </xs:schema>
1143
```

1144 5.1.9 Software-PKI

1145 The Software-PKI class is identified when a Principal uses an X.509 certificate stored in software
1146 to authenticate to the identity provider over an SSL protected session.
1147

1148 5.1.9.1 Associated Liberty URI

1149
1150 <http://www.projectliberty.org/schemas/authctx/classes/Software-PKI>

1151 **5.1.9.2 Class Schema**

1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <annotation>
  <documentation> http://www.projectliberty.org/schemas/authctx/classes/Software-PKI
  </documentation>
  </annotation>
  <xs:element name="AuthenticationContextStatement">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticationMethod"/>
        <xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="AuthenticationMethod">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="1" maxOccurs="1"
ref="PrincipalAuthenticationMechanism"/>
          <xs:element minOccurs="1" maxOccurs="1" ref="Authenticator"/>
          <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticatorTransportProtocol"/>
          <xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Authenticator">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" ref="Dig-sig"/>
            <xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="AuthenticatorTransportProtocol">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="1" maxOccurs="1" ref="SSL"/>
              <xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="Dig-sig">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
          </xs:element>
          <xs:element name="Password">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
          </xs:element>
          <xs:element name="PrincipalAuthenticationMechanism">
            <xs:complexType>
              <xs:sequence>
                <xs:element minOccurs="1" maxOccurs="1" ref="Password"/>
                <xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:element>
        </xs:element>
      </xs:element>
    </xs:element>
  </xs:schema>
```

```
1214     </xs:element>
1215     <xs:element name="SSL">
1216         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1217 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1218     </xs:element>
1219 </xs:schema>
1220
```

1221 5.1.10 Time-Sync-Token

1222 The Time-Sync-Token class is identified when a Principal authenticates through a time
1223 synchronization token.

1224 5.1.10.1 Associated Liberty URI

1225 <http://www.projectliberty.org/schemas/authctx/classes/Time-Sync-Token>

1226 5.1.10.2 Class Schema

```
1227 <?xml version="1.0" encoding="UTF-8"?>
1228 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
1229
1230 <annotation>
1231 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Time-Sync-Token
1232 </documentation>
1233 </annotation>
1234
1235     <xs:element name="AuthenticationContextStatement">
1236         <xs:complexType>
1237             <xs:sequence>
1238                 <xs:element minOccurs="1" maxOccurs="1"
1239 ref="AuthenticationMethod"/>
1240                 <xs:any namespace="##any" minOccurs="0"
1241 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1242             </xs:complexType>
1243         </xs:element>
1244         <xs:element name="AuthenticationMethod">
1245             <xs:complexType>
1246                 <xs:sequence>
1247                     <xs:element minOccurs="1" maxOccurs="1"
1248 ref="PrincipalAuthenticationMechanism"/>
1249                     <xs:any namespace="##any" minOccurs="0"
1250 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1251                 </xs:complexType>
1252             </xs:element>
1253             <xs:element name="Generation">
1254                 <xs:complexType>
1255                     <xs:attribute name="mechanism" fixed="principalchosen" />
1256                 </xs:complexType>
1257             </xs:element>
1258
1259     <xs:element name="PrincipalAuthenticationMechanism">
1260         <xs:complexType>
1261             <xs:sequence>
1262                 <xs:element minOccurs="1" maxOccurs="1" ref="Token"/>
1263                 <xs:any namespace="##any" minOccurs="0"
1264 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1265             </xs:complexType>
1266         </xs:element>
1267         <xs:element name="TimeSyncToken">
1268             <xs:complexType>
1269
```

```

1271     <xs:attribute name="deviceType" fixed="hardware" />
1272     <xs:attribute name="seedLength" fixed="64" />
1273     <xs:attribute name="deviceInHand" fixed="true" />
1274   </xs:complexType>
1275 </xs:element>
1276 <xs:element name="Token">
1277   <xs:complexType>
1278     <xs:sequence>
1279       <xs:element minOccurs="1" maxOccurs="1" ref="TimeSyncToken"/>
1280       <xs:any namespace="##any" minOccurs="0"
1281 maxOccurs="unbounded" processContents="lax" /></xs:sequence>
1282     </xs:complexType>
1283   </xs:element>
1284 </xs:schema>

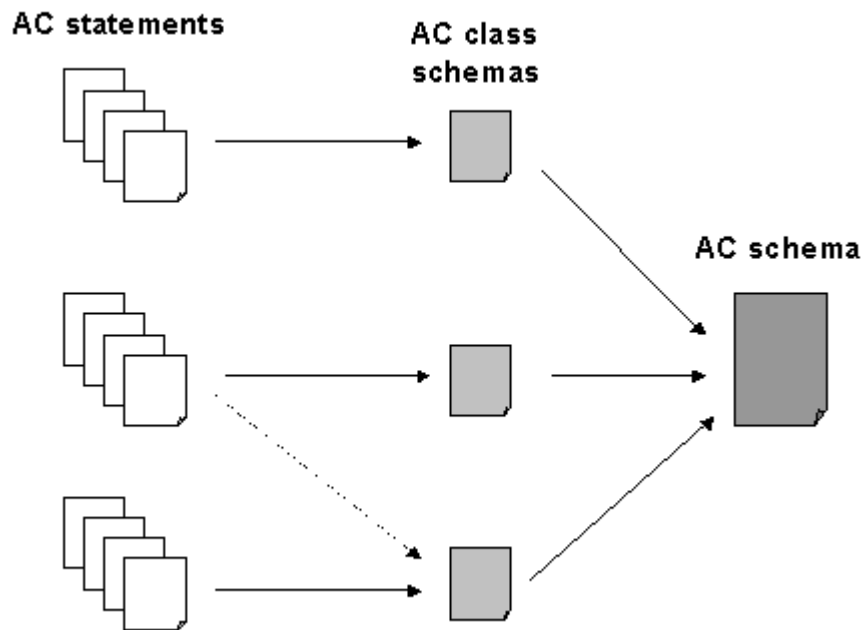
```

1285
1286

1287 5.2 Authentication Context Schema

1288 The relationship between authentication context statements, authentication context classes, and the
1289 authentication context XML schema is shown in Figure 3.

1290



1291

1292 **Figure 3: Relationship between authentication context statements, classes, and XML schema**

1293

1294 Authentication context statements may conform to authentication context classes, which are
1295 themselves logical subsets of the authentication context XML schema.

1296

1297

1298 5.2.1 XML Schema

1299

```

1300 <?xml version="1.0" encoding="UTF-8"?>
1301
1302 <schema targetNamespace="http://www.projectliberty.org/schemas/authctx/2002/05"
1303 xmlns:xsd="http://www.w3.org/2001/XMLSchema "
1304 xmlns:AC="http://www.projectliberty.org/schemas/authctx/2002/05"
1305 xmlns="http://www.w3.org/2001/XMLSchema" version="1.0">
1306
1307 <annotation>
1308 <documentation> http://www.projectliberty.org/schemas/authctx/2002/05/
1309 </documentation>
1310 </annotation>
1311
1312 <element name="AuthenticationContextStatement">
1313 <annotation>
1314 <documentation>A claim made by an identity provider with respect to
1315 the authentication context associated with an authentication assertion. </documentation>
1316 </annotation>
1317 <complexType>
1318 <sequence>
1319 <element ref="AC:Identification" minOccurs="0"/>
1320 <element ref="AC:TechnicalProtection" minOccurs="0"/>
1321 <element ref="AC:OperationalProtection" minOccurs="0"/>
1322 <element ref="AC:AuthenticationMethod" minOccurs="0"/>
1323 <element ref="AC:GoverningAgreements" minOccurs="0"/>
1324 <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1325 processContents="lax" />
1326 </sequence>
1327 <attribute name="ID" type="ID"/>
1328 </complexType>
1329 </element>
1330
1331 <element name="Identification">
1332 <annotation>
1333 <documentation>Refers to those characteristics that describe the
1334 processes and mechanisms the
1335 identity provider uses to initially create an association between a
1336 Principal and the identity
1337 (or name) by which the Principal will be known</documentation>
1338 </annotation>
1339 <complexType>
1340 <sequence>
1341 <element ref="AC:PhysicalVerification" minOccurs="0"/>
1342 <element ref="AC:WrittenConsent" minOccurs="0"/>
1343 <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1344 processContents="lax" />
1345 </sequence>
1346 <attribute name="nym">
1347 <annotation>
1348 <documentation>This attribute indicates whether or not
1349 the Identification mechanisms allow the
1350 actions of the Principal to be linked to an actual end
1351 user.</documentation>
1352 </annotation>
1353 <simpleType>
1354 <restriction base="NMTOKEN">
1355 <enumeration value="anonymity"/>
1356 <enumeration value="verinyimity"/>
1357 <enumeration value="pseudonymity"/>
1358 </restriction>
1359 </simpleType>
1360 </attribute>
1361 </complexType>
1362 </element>
1363 <element name="PhysicalVerification">
1364 <annotation>

```

```

1365         <documentation>This element indicates that identification has been
1366 performed in a physical
1367         face-to-face meeting with the principal and not in an online manner.
1368     </documentation>
1369     </annotation>
1370     <complexType>
1371         <attribute name="credentialLevel">
1372             <simpleType>
1373                 <restriction base="NMTOKEN">
1374                     <enumeration value="primary"/>
1375                     <enumeration value="secondary"/>
1376                 </restriction>
1377             </simpleType>
1378         </attribute>
1379     </complexType>
1380 </element>
1381 <element name="WrittenConsent">
1382     <complexType><sequence><any namespace="##any" minOccurs="0"
1383 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1384 </element>
1385 <element name="TechnicalProtection">
1386     <annotation>
1387         <documentation>Refers to those characterstics that describe how the
1388 'secret' (the knowledge or possession of which allows the Principal to authenticate to
1389 the identity provider) is kept secure</documentation>
1390     </annotation>
1391     <complexType>
1392         <sequence>
1393             <element ref="AC:PrivateKeyProtection" minOccurs="0"/>
1394             <element ref="AC:SharedKeyProtection" minOccurs="0"/>
1395             <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1396 processContents="lax" />
1397         </sequence>
1398     </complexType>
1399 </element>
1400 <element name="SharedKeyProtection">
1401     <annotation>
1402         <documentation>This element indicates the types and strengths of
1403 facilities
1404         of a UA used to protect a shared secret key from unauthorized access
1405 and/or use.</documentation>
1406     </annotation>
1407     <complexType>
1408         <choice minOccurs="0">
1409             <element ref="AC:MobileDevice"/>
1410             <element ref="AC:MobileAuthCard"/>
1411         </choice>
1412     </complexType>
1413 </element>
1414 <element name="MobileDevice">
1415     <annotation>
1416         <documentation>This element idicates that the shared secret key is
1417 securely maintained in a mobile device
1418         (as opposed to being stored in a mobile authentication
1419 card).</documentation>
1420     </annotation>
1421     <complexType><sequence><any namespace="##any" minOccurs="0"
1422 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1423 </element>
1424 <element name="MobileAuthCard">
1425     <annotation>
1426         <documentation>This element indicates that the shared secret key is
1427 securely maintained in a mobile authentication card (e.g., a SIM card).</documentation>
1428     </annotation>
    
```

```

1429     <complexType><sequence><any namespace="##any" minOccurs="0"
1430 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1431     </element>
1432     <element name="PrivateKeyProtection">
1433         <annotation>
1434             <documentation>This element indicates the types and strengths of
1435 facilities
1436             of a UA used to protect a private key from unauthorized access
1437 and/or use.</documentation>
1438         </annotation>
1439         <complexType>
1440             <sequence>
1441                 <element ref="AC:KeyActivation" minOccurs="0"/>
1442                 <element ref="AC:KeyStorage" minOccurs="0"/>
1443                 <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1444 processContents="lax" />
1445             </sequence>
1446         </complexType>
1447     </element>
1448     <element name="KeyActivation">
1449         <annotation>
1450             <documentation>The actions that must be performed before the private
1451 key can be used. </documentation>
1452         </annotation>
1453         <complexType>
1454             <choice>
1455                 <element ref="AC:Password"/>
1456             </choice>
1457         </complexType>
1458     </element>
1459     <element name="KeyStorage">
1460         <annotation>
1461             <documentation>In which medium is the private key stored.
1462
1463             memory - the private key is stored in memory.
1464
1465             smartcard - the private key is stored in a smartcard.
1466
1467             token - the private key is stored in a hardware token.
1468
1469             MobileAuthCard - the private key is stored in a mobile
1470 authentication card (e.g., SIM card).
1471         </documentation>
1472     </annotation>
1473     <complexType>
1474         <attribute name="medium" use="required">
1475             <simpleType>
1476                 <restriction base="NMTOKEN">
1477                     <enumeration value="memory"/>
1478                     <enumeration value="smartcard"/>
1479                     <enumeration value="token"/>
1480                     <enumeration value="MobileAuthCard"/>
1481                 </restriction>
1482             </simpleType>
1483         </attribute>
1484     </complexType>
1485 </element>
1486
1487     <element name="Password">
1488         <annotation>
1489             <documentation>This element indicates that a password (or PIN or
1490 passphrase) has been used to authenticate the Principal or
1491 to gain access to some resource (for example, to gain access to the
1492 private key).</documentation>
1493

```



```

1494         </annotation>
1495         <complexType>
1496             <sequence>
1497                 <element ref="AC:Length" minOccurs="0"/>
1498                 <element ref="AC:Generation" minOccurs="0"/>
1499                 <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1500 processContents="lax" />
1501             </sequence>
1502         </complexType>
1503     </element>
1504     <element name="Token">
1505         <annotation>
1506             <documentation>This element indicates that a hardware or software
1507 token is
1508             used as a method of identifying the Principal.</documentation>
1509         </annotation>
1510         <complexType>
1511             <sequence>
1512                 <element ref="AC:TimeSyncToken"/>
1513                 <any namespace="##any" minOccurs="0"
1514 maxOccurs="unbounded" processContents="lax" />
1515             </sequence>
1516         </complexType>
1517     </element>
1518     <element name="TimeSyncToken">
1519         <annotation>
1520             <documentation>This element indicates that a time synchronization
1521 token is used to identify the Principal.
1522             hardware - the time synchronization token has been implemented in
1523 hardware.
1524             software - the time synchronization token has been implemented in
1525 software.
1526             SeedLength - the length, in bits, of the random seed used in the
1527 time synchronization token.
1528             </documentation>
1529         </annotation>
1530         <complexType>
1531             <attribute name="DeviceType" use="required">
1532                 <simpleType>
1533                     <restriction base="NMTOKEN">
1534                         <enumeration value="hardware"/>
1535                         <enumeration value="software"/>
1536                     </restriction>
1537                 </simpleType>
1538             </attribute>
1539             <attribute name="SeedLength" type="integer" use="required"/>
1540             <attribute name="DeviceInHand" use="required">
1541                 <simpleType>
1542                     <restriction base="NMTOKEN">
1543                         <enumeration value="true"/>
1544                         <enumeration value="false"/>
1545                     </restriction>
1546                 </simpleType>
1547             </attribute>
1548         </complexType>
1549     </element>
1550     <complexType>
1551         <sequence>
1552             <element name="Smartcard">
1553                 <annotation>
1554                     <documentation>This element indicates that a smartcard is used to
1555 identity the Principal.</documentation>
1556                 </annotation>

```

```

1558     <complexType><sequence><any namespace="##any" minOccurs="0"
1559 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1560     </element>
1561     <element name="Length">
1562         <annotation>
1563             <documentation>This element indicates the minimum and/or maximum
1564 ASCII
1565             length of the password which is enforced (by the UA or the IdP). In
1566 other words,
1567             this is the minimum and/or maximum number of ASCII characters
1568 required to represent a valid password.
1569
1570             min - the minimum number of ASCII characters required in a valid
1571 password, as enforced by the UA or the IdP.
1572
1573             max - the maximum number of ASCII characters required in a valid
1574 password, as enforced by the UA or the IdP.
1575         </documentation>
1576     </annotation>
1577     <complexType>
1578         <attribute name="min" type="integer" use="required"/>
1579         <attribute name="max" type="integer" use="optional"/>
1580     </complexType>
1581 </element>
1582 <element name="Generation">
1583     <annotation>
1584         <documentation>Indicates whether the password was chosen by the
1585 Principal or auto-supplied by the identity provider.
1586
1587         principalchosen - the Principal is allowed to choose the value of
1588 the password. This is true even if the initial password is chosen at
1589 random by the UA or the IdP and the Principal is then free to change
1590 the password.
1591
1592         automatic - the password is chosen by the UA or the IdP to be
1593 cryptographically strong in some sense, or to satisfy certain
1594 password rules, and that the Principal is not free to change it or
1595 to choose a new password.
1596
1597     </documentation>
1598 </annotation>
1599 <complexType>
1600     <attribute name="mechanism" use="required">
1601         <simpleType>
1602             <restriction base="NMTOKEN">
1603                 <enumeration value="principalchosen"/>
1604                 <enumeration value="automatic"/>
1605             </restriction>
1606         </simpleType>
1607     </attribute>
1608 </complexType>
1609 </element>
1610 <element name="AuthenticationMethod">
1611     <annotation>
1612         <documentation>Refers to those characteristics that define the
1613 mechanisms by which the Principal authenticates to the identity
1614 provider.</documentation>
1615     </annotation>
1616 <complexType>
1617     <sequence>
1618         <element ref="AC:PrincipalAuthenticationMechanism"/>
1619         <element ref="AC:Authenticator" minOccurs="0"/>
1620         <element ref="AC:AuthenticatorTransportProtocol"/>
1621         <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1622 processContents="lax" />

```

```

1623         </sequence>
1624     </complexType>
1625 </element>
1626 <element name="PrincipalAuthenticationMechanism">
1627     <annotation>
1628         <documentation>The method that a Principal employs to perform
1629 authentication to local system components.</documentation>
1630     </annotation>
1631     <complexType>
1632         <choice minOccurs="0" maxOccurs="unbounded">
1633             <element ref="AC:Password"/>
1634             <element ref="AC:Token"/>
1635             <element ref="AC:Smartcard"/>
1636         </choice>
1637     </complexType>
1638 </element>
1639 <element name="Authenticator">
1640     <annotation>
1641         <documentation>The method applied to validate a principal's
1642 authentication across a network </documentation>
1643     </annotation>
1644     <complexType>
1645         <choice minOccurs="0" maxOccurs="unbounded">
1646             <element ref="AC:PreviousSession"/>
1647             <element ref="AC:Dig-sig"/>
1648             <element ref="AC:ZeroKnowledge"/>
1649             <element ref="AC:SharedSecretChallengeResponse"/>
1650         </choice>
1651     </complexType>
1652 </element>
1653 <element name="PreviousSession">
1654     <annotation>
1655         <documentation>Indicates that the Principal has been strongly
1656 authenticated in a previous session during which
1657         the IdP has set a cookie in the UA. During the present session the
1658 Principal has only been authenticated by
1659         the UA returning the cookie to the IdP.</documentation>
1660     </annotation>
1661     <complexType><sequence><any namespace="##any" minOccurs="0"
1662 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1663 </element>
1664 <element name="ZeroKnowledge">
1665     <annotation>
1666         <documentation>This element indicates that the Principal has been
1667 authenticated by a zero knowledge
1668         technique as specified in ISO/IEC 9798-5.</documentation>
1669     </annotation>
1670     <complexType><sequence><any namespace="##any" minOccurs="0"
1671 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1672 </element>
1673 <element name="SharedSecretChallengeResponse">
1674     <annotation>
1675         <documentation>This element indicates that the Principal has been
1676 authenticated by a challenge-response
1677         protocol utilizing shared secret keys and symmetric
1678 cryptography.</documentation>
1679     </annotation>
1680     <complexType><sequence><any namespace="##any" minOccurs="0"
1681 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1682 </element>
1683 <element name="Dig-sig">
1684     <annotation>
1685         <documentation>This element indicates that the Principal has been
1686 authenticated by a mechanism which involves the Principal

```

```
1687         computing a digital signature over at least challenge data provided
1688 by the IdP.</documentation>
1689     </annotation>
1690     <complexType><sequence><any namespace="##any" minOccurs="0"
1691 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1692 </element>
1693 <element name="AuthenticatorTransportProtocol">
1694     <annotation>
1695         <documentation>The protocol across which Authenticator information
1696 is transferred to an identity provider verifier.</documentation>
1697     </annotation>
1698     <complexType>
1699         <choice minOccurs="0" maxOccurs="unbounded">
1700             <element ref="AC:HTTP"/>
1701             <element ref="AC:SSL"/>
1702             <element ref="AC:MobileNetwork"/>
1703             <element ref="AC:WTLS"/>
1704             <element ref="AC:IPSec"/>
1705         </choice>
1706     </complexType>
1707 </element>
1708 <element name="HTTP">
1709     <annotation>
1710         <documentation>This element indicates that the Authenticator has
1711 been transmitted
1712         using bare HTTP utilizing no additional security
1713 protocols.</documentation>
1714     </annotation>
1715     <complexType><sequence><any namespace="##any" minOccurs="0"
1716 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1717 </element>
1718 <element name="IPSec">
1719     <annotation>
1720         <documentation>This element indicates that the Authenticator has
1721 been transmitted
1722         using a transport mechanism protected by an IPSEC
1723 session.</documentation>
1724     </annotation>
1725     <complexType><sequence><any namespace="##any" minOccurs="0"
1726 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1727 </element>
1728 <element name="WTLS">
1729     <annotation>
1730         <documentation>This element indicates that the Authenticator has
1731 been transmitted
1732         using a transport mechanism protected by a WTLS
1733 session.</documentation>
1734     </annotation>
1735     <complexType><sequence><any namespace="##any" minOccurs="0"
1736 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1737 </element>
1738 <element name="MobileNetwork">
1739     <annotation>
1740         <documentation>This element indicates that the Authenticator has
1741 been transmitted
1742         solely across a mobile network using no additional security
1743 mechanism.</documentation>
1744     </annotation>
1745     <complexType><sequence><any namespace="##any" minOccurs="0"
1746 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1747 </element>
1748 <element name="SSL">
1749     <annotation>
1750         <documentation>This element indicates that the Authenticator has
1751 been transmitted
```

```

1752         using a transport mechanism protected by an SSL or TLS
1753 session.</documentation>
1754         </annotation>
1755         <complexType><sequence><any namespace="##any" minOccurs="0"
1756 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1757         </element>
1758         <element name="OperationalProtection">
1759             <annotation>
1760                 <documentation>Refers to those characteristics that describe
1761 procedural security controls employed by the identity provider.</documentation>
1762             </annotation>
1763             <complexType>
1764                 <sequence>
1765                     <element ref="AC:SecurityAudit" minOccurs="0"/>
1766                     <element ref="AC:DeactivationCallCenter" minOccurs="0"/>
1767                     <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1768 processContents="lax" />
1769                 </sequence>
1770             </complexType>
1771         </element>
1772         <element name="SecurityAudit">
1773             <complexType>
1774                 <sequence>
1775                     <element ref="AC:SwitchAudit" minOccurs="0"/>
1776                     <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1777 processContents="lax" />
1778                 </sequence>
1779             </complexType>
1780         </element>
1781         <element name="SwitchAudit">
1782             <complexType><sequence><any namespace="##any" minOccurs="0"
1783 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1784         </element>
1785         <element name="DeactivationCallCenter">
1786             <complexType><sequence><any namespace="##any" minOccurs="0"
1787 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1788         </element>
1789         <element name="GoverningAgreements">
1790             <annotation>
1791                 <documentation>Provides a mechanism for linking to external (likely
1792 human readable) documents in which the identity provider can define business level
1793 authentication context, e.g. liability constraints, contractual
1794 obligations.</documentation>
1795             </annotation>
1796             <complexType>
1797                 <sequence>
1798                     <element ref="AC:GoverningAgreementRef"/>
1799                 </sequence>
1800             </complexType>
1801         </element>
1802         <element name="GoverningAgreementRef">
1803             <complexType>
1804                 <attribute name="governingAgreementRef" type="anyURI"
1805 use="required"/>
1806             </complexType>
1807         </element>
1808 </schema>

```

1809 6 References

1810 [LibertyGloss] Mauldin, H., & Wason, T., eds. (January 2003). "Liberty Architecture
1811 Glossary," Version 1.1. Liberty Alliance Project,
1812 <<http://www.projectliberty.org/specs/>>.

- 1813 [LibertyProtSchema] Beatty, J., & Kemp, J., eds. (January 2003). “Liberty Protocols and
1814 Schema Specification,” Version 1.1. Liberty Alliance Project,
1815 <<http://www.projectliberty.org/specs/>>.
- 1816 [PDS] Santesson, S. & Baum, M., (May 2000). “Internet X.509 Public Key
1817 Infrastructure PKI Disclosure Statement,” Internet Draft . The Internet
1818 Engineering Task Force, <<http://www.verisign.com/repository/pds.txt>>
1819 [20 December 2002].
- 1820 [RFC2119] Bradner, S. (March 1997). “Key words for use in RFCs to Indicate
1821 Requirement Levels,” RFC 2119. The Internet Engineering Task Force,
1822 <<http://www.rfc-editor.org/rfc/rfc2119.txt>> [18 December 2002].
- 1823 [RFC2527] Chokhani, S., Ford, W. (March 1999). “Internet X.509 Public Key
1824 Infrastructure Certificate Policy and Certification Practices Framework,”
1825 RFC 2527. The Internet Engineering Task Force,
1826 <<http://www.ietf.org/rfc/rfc2527.txt?number=2527>> [20 December
1827 2002].
- 1828 [SAMLBind] Mishra, P., ed. (05 Nov. 2002). “Bindings and Profiles for the OASIS
1829 Security Assertion Markup Language (SAML),” Version 1.0, OASIS
1830 Standard. Organization for the Advancement of Structured Information
1831 Standards, <[http://www.oasis-
1832 open.org/committees/security/#documents](http://www.oasis-open.org/committees/security/#documents)> [18 December 2002].
- 1833 [SAMLCore] Hallam-Baker, P., Maler, E., eds. (05 Nov. 2002). “Assertions and
1834 Protocol for the OASIS Security Assertion Markup Language (SAML),”
1835 Version 1.0, OASIS Standard. Organization for the Advancement of
1836 Structured Information Standards, <[http://www.oasis-
1837 open.org/committees/security/#documents](http://www.oasis-open.org/committees/security/#documents)> [18 December 2002].
- 1838 [Schema1] Thompson, H. S., Beech, D., Maloney, M., & Mendleson, N., eds. (May
1839 2002). “XML Schema Part 1: Structures,” Recommendation. World
1840 Wide Web Consortium, <<http://www.w3.org/TR/xmlschema-1/>> [18
1841 December 2002].
- 1842