



1

2 **Liberty Authentication Context Specification**

3 **Draft Version 1.2-05**

4 **12 April 2003**

5 **Editors:**

6 John Kemp, IEEE-ISTO

7 Paul Madsen, Entrust

8

9 **Contributors:**

10 Xavier Serret, Gemplus

11 Tom Wason, IEEE-ISTO

12

13 **Abstract:**

14 If a service provider is to rely on the authentication of a Principal by an identity provider, the service provider may
15 require information additional to the authentication itself to allow it to put the authentication in a trust context. This
16 specification defines a syntax for the definition of authentication context statements and an initial list of Liberty
17 authentication context classes.

18

19 Copyright © 2003 Liberty Alliance Project

20

20 **Notice**

21 Copyright © 2002,2003 ActivCard; American Express Travel Related Services; America Online, Inc.; Bank of
22 America; Bell Canada; Catavault; Cingular Wireless; Cisco Systems, Inc.; Citigroup; Communicator, Inc.; Consignia;
23 Cyberun Corporation; Deloitte & Touche LLP; Earthlink, Inc.; Electronic Data Systems, Inc.; Entrust, Inc.; Ericsson;
24 Fidelity Investments; France Telecom; Gemplus; General Motors; Hewlett-Packard Company; i2 Technologies, Inc.;
25 Internet2; Intuit Inc.; MasterCard International; NEC Corporation; Netegrity; NeuStar; Nextel Communications;
26 Nippon Telegraph and Telephone Company; Nokia Corporation; Novell, Inc.; NTT DoCoMo, Inc.; OneName
27 Corporation; Openwave Systems Inc.; Phaos Technology; PricewaterhouseCoopers LLP; Register.com; RSA Security
28 Inc; Sabre Holdings Corporation; SAP AG; SchlumbergerSema; SK Telecom; Sony Corporation; Sun Microsystems,
29 Inc.; Trustgenix; United Airlines; VeriSign, Inc.; Visa International; Vodafone Group Plc; Wave Systems;. All rights
30 reserved.

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

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

44
45 Liberty Alliance Project
46 Licensing Administrator
47 c/o IEEE-ISTO
48 445 Hoes Lane
49 Piscataway, NJ 08855-1331, USA
50 info@projectliberty.org

51

51

52 **Revision History**

Rev	Date	By Whom	Description
1.2 Draft 01	Feb-10-03	Xavier Serret, John Kemp	<ul style="list-style-type: none">• Added multi-IdP support• Added explicit text for Governing Agreements (extracted from the existing schema) Added AC:extension element to cover the naked <any> in many places
02	Mar-10-03	John Kemp	<ul style="list-style-type: none">• Added IAP IP Context• Corrected issue with extension element
03	Mar-10-03	John Kemp	<ul style="list-style-type: none">• Cleaned up for publication
04	Mar-21-03	John Kemp	<ul style="list-style-type: none">• Extension comes from ac: namespace now
05	Apr-12-03	Tom Wason	<ul style="list-style-type: none">• Formatted for publication• Added abstract

53

53 **Table of Contents**

54 1 Introduction 5
55 1.1 Notation 5
56 2 Overview 5
57 3 Authentication Context 6
58 3.1 Authentication Context Classes..... 7
59 3.2 Authentication Quality..... 8
60 3.2.1 Service Provider Request 8
61 3.2.2 Identity Provider Response..... 9
62 4 Previous work 9
63 4.1 PKI 9
64 4.2 SAML 10
65 5 Liberty Authentication Context Mechanisms 11
66 5.1 Authentication Context Classes..... 11
67 5.1.1 MobileContract 11
68 5.1.2 MobileDigitalID 13
69 5.1.3 MobileUnregistered 16
70 5.1.4 Password 18
71 5.1.5 Password- ProtectedTransport 19
72 5.1.6 Previous-Session 20
73 5.1.7 Smartcard 21
74 5.1.8 Smartcard-PKI 22
75 5.1.9 Software-PKI 24
76 5.1.10 Time-Sync-Token 25
77 5.1.11 Internet Protocol 26
78 5.1.12 Internet Protocol + Password 27
79 5.2 Authentication Context Schema 28
80 5.2.1 XML Schema 28
81 6 References 37
82
83

83 1 Introduction

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

86 1.1 Notation

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

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

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

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

99
100 Listings of XML schemas appear like this.

101
102 Example code listings appear like this.

103

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

- 106 • The prefix `lib:` stands for the Liberty ID-FF namespace (`urn:liberty:iff:1.2`).
- 107 • The prefix `AC:` stands for the Liberty Authentication Namespace (`urn:liberty:ac:1.2`).

108 The prefix `saml:` stands for the SAML assertion namespace (`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 (<http://www.w3.org/2000/09/xmldsig#>).

112 The prefix `xsd:` stands for the W3C XML schema namespace in example listings
113 (<http://www.w3.org/2001/XMLSchema>). In schema listings, this namespace is the default, and no prefix is
114 shown.

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

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

118 2 Overview

119 Liberty will not prescribe a single technology, protocol, or policy for the processes by which identity providers issue
120 identities to Principals and by which those Principals subsequently authenticate themselves to the identity provider.
121 Different identity providers will choose different technologies, follow different processes, and be bound by different

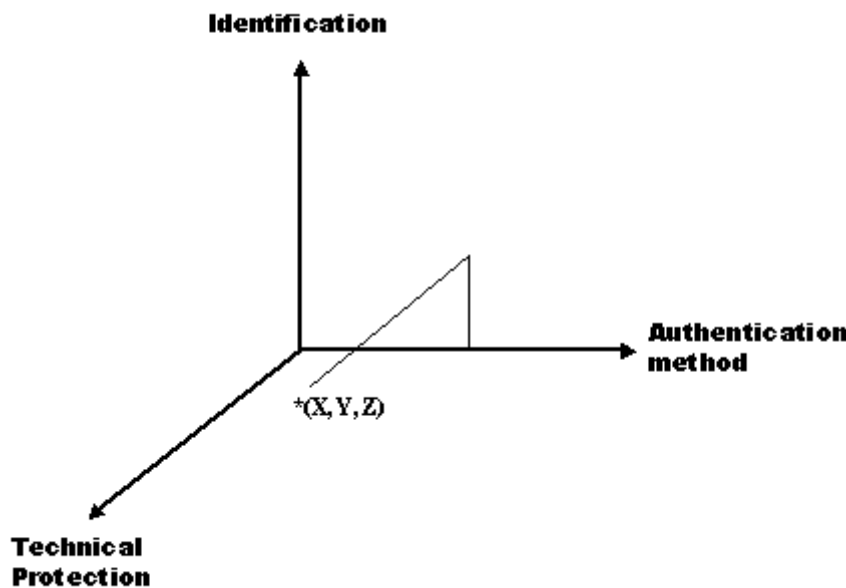
122 legal obligations with respect to how they authenticate Principals. The choices that an identity provider makes here
123 will be driven in large part by the requirements of the service providers with which the identity provider has affiliated
124 into a circle of trust. These requirements themselves will be determined by the nature of the service (that is, the
125 sensitivity of any information exchanged, the associated financial value, the service providers risk tolerance, etc.) that
126 the service provider will be providing to the Principal. Consequently, for anything other than trivial services, if the
127 service provider is to place sufficient confidence in the authentication assertions it receives from an identity provider,
128 it will be necessary for the service provider to know which technologies, protocols, and processes were used or
129 followed for the original authentication mechanism on which the authentication assertion is based. Armed with this
130 information and trusting the origin of the actual assertion, the service provider will be better able to make an informed
131 entitlements decision regarding what services the subject of the authentication assertion should be allowed to access.
132
133 *Authentication context* is defined as the information additional to the authentication assertion itself that the service
134 provider may require before it makes an entitlements decision.

135 **3 Authentication Context**

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

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

144
145 The variations and permutations in the examples above guarantee that not all authentication assertions are the same; a
146 particular authentication assertion will be characterized by the values for each of these variables. A somewhat helpful
147 model is to think of an authentication assertion as defined by its coordinates in a multidimensional space. This model
148 is demonstrated in Figure 1 (where only three axes are shown).
149
150



151
152 **Figure 1: Authentication assertion as defined by its coordinates in multidimensional space**

153
154 A particular authentication context statement will be characterized by its values along the different axes and
155 consequently by its position in this space.

156 3.1 Authentication Context Classes

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

166
167 A particular Liberty authentication context class will define a list of required characteristics of the processes,
168 procedures, and mechanisms by which the identity provider verifies the Principal before issuing an identity, protects
169 the secrets on which subsequent authentications are based, and the mechanisms used for this authentication. These
170 characteristics can be categorized as

171
172 **Identification** – Characteristics that describe the processes and mechanism the identity provider uses to initially create
173 an association between a Principal and the identity (or name) by which the Principal will be known.

174 **Physical Protection** – Characteristics that specify physical controls on the facility housing the identity provider's
175 systems (for example, site location and construction, access controls).

176 **Operational Protection** – Characteristics that describe procedural security controls employed by the identity provider
177 (for example, security audits, records archival).

178 **Technical Protection** – Characteristics that describe how the “secret” (the knowledge or possession of which allows
179 the Principal to authenticate to the identity provider) is kept secure.

180 **Authentication Method** – Characteristics that define the mechanisms by which the Principal authenticates to the
181 identity provider (for example, a password versus a smartcard).

182 **Governing Agreements**- Provide a mechanism for linking to external (likely human readable) documents in which
183 the identity provider can define business level authentication context, for example, liability constraints or contractual
184 obligations. Governing Agreements are normally profiled on a Authentication Class-level basis but can be specific to a
185 given Authentication Context statement.

186 **Authenticating IdP**- When Principal authentication is relayed between different IdPs to provide for a seamless cross-
187 IdP SSO experience the relaying IdP **MAY** include this element with a reference to the originating IdP. The relaying
188 IdP **MAY** include as well extra specific inter-IdP Governing Agreements that may affect those specified at the
189 Authentication Class-level.

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

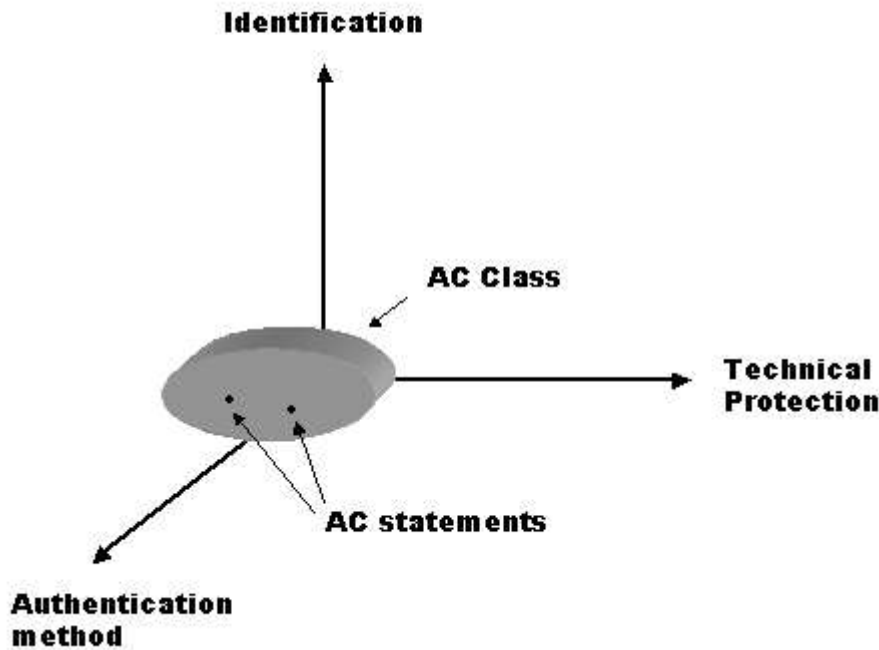


Figure 2: Relationship between authentication context class and statements

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

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

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

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

Protects service providers from impact of new authentication technologies.

Makes it easier for identity providers to publish their authentication capabilities, for example, through WSDL.

3.2 Authentication Quality

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

3.2.1 Service Provider Request

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

1. A match on a particular authentication context statement

- 224 2. A match within a specific authentication context class
225 3. A match or better on a particular authentication context class
226 4. A match within an ordered list (which is designated by the service provider) of authentication context classes
227 5. A match on the originating IdP within an ordered list.

228
229 Option 1 will require that the identity provider and service provider have previously agreed on the details of a
230 particular authentication context that either does not fall into one of the Liberty-defined authentication context classes
231 or needs to be constrained more tightly.

232
233 Option 2 is expected to be the typical scenario.

234
235 For option 3, the decision as to what is better is left to the entity best qualified to make that determination, the identity
236 provider. The service provider, trusting the identity provider's judgment, will accept the assertion it receives back
237 because it will be confident the assertion meets (or exceeds) the provider's requirements.

238
239 Option 4 will give the service provider greater control over the authentication context classes to which the
240 authentication assertions it receives conform. The identity provider is given no leeway in providing an authentication
241 assertion conforming to a class not on the list.

242
243 Option 5 will give the service provider control on which is the original IdP authenticating the Principal. A list containing
244 zero elements **MUST** be interpreted as the a non-delegable authentication request.

245
246 If the service provider does not specify any of the above options in the `<lib:AuthnRequest>`, the identity
247 provider will be free to provide an authentication context of its choosing.

248 **3.2.2 Identity Provider Response**

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

252
253 The returned authentication assertion will include a URI specifying the associated authentication context statement.

254 **4 Previous work**

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

256 **4.1 PKI**

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

260
261 **Certificate Practice Statement (CPS)** is a statement of the practices that a certification authority employs in issuing
262 certificates. A certificate practice statement may take the form of a declaration by the certification authority of the
263 details of its trustworthy systems and the practices it employs in support of its issuance of certificates.

264 **Certificate Policy** is a named set of rules that indicates the applicability of a certificate to a particular community
265 and/or class of application. For example, a certificate policy might indicate that a particular type of certificate is
266 appropriate for the authentication of participants in a business-to-business transaction within a given price range. The
267 fundamental difference between the certificate practice statement and the certificate policy is that the former is
268 "owned" by the issuing certification authority and the latter by the entities who will use the issued certificates.
269 Certificate users define certificate policies, and certification authorities (with different certificate practice statements)
270 attest that a particular certificate is appropriate for that certificate policy. (See [[RFC2527](#)].)

271 **PKI Disclosure Statement** is a supplementary instrument that discloses critical information about the policies and
272 practices of a certificate authority or PKI. A PKI disclosure statement is a vehicle for disclosing and emphasizing
273 information normally covered in detail by associated certificate policy and/or certification practice statement
274 documents. Consequently, a PKI disclosure statement is not intended to replace a certificate policy or practice
275 statement. (See [[PDS](#)].)

276 **Key Usage**, as defined in X.509, defines the intended use for a key contained in a certificate. These uses (or *values*)
277 are digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment, keyAgreement, keyCertSign, CRLSign,
278 encipherOnly, and decipherOnly.

279 **Extended Key Usage**, as the name indicates, extends the possible uses for a key beyond the original nine, each use
280 identified by an object identifier. Extended key usage is primarily used by the relying party. As part of its validation
281 algorithm, a relying party will check for these values to determine whether a given certificate is appropriate for the
282 application.

283 4.2 SAML

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

```
289 <?xml version="1.0"?>  
290 <saml:Assertion>  
291 <saml:AuthenticationStatement AuthenticationMethod=" urn:ietf:rfc:2246">  
292   <saml:Subject>  
293     <saml:NameIdentifier  
294       Format="http://www.oasis-open.org/committees/security/docs/cs-sstc-core-  
295         28#X509SubjectName">cn=Joe User,dc=projectliberty,dc=org  
296     </saml:NameIdentifier>  
297   </saml:Subject>  
298 </saml:AuthenticationStatement>  
299 <saml:Advice>  
300 <!--additional elements in separate namespace à  
301 </saml:Advice>  
302 </saml:Assertion>
```

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

307 `<saml:Condition>` [Optional]. Conditions that **MUST** be taken into account in assessing the validity of the
308 assertion.

309 `<saml:Advice>` [Optional]. Additional information related to the assertion that assists processing in certain
310 situations, but **MAY** be ignored by applications that do not support its use.

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

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

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

323 5 Liberty Authentication Context Mechanisms

324 5.1 Authentication Context Classes

325 The Liberty authentication context classes are listed in this section.

326

327 No ranking is implied by the order of classes.

328

329 Classes are identified by URIs with the initial stem:

330

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

332

333 5.1.1 MobileContract

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

336 5.1.1.1 Associated Liberty URI

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

338 5.1.1.2 Class Schema

339

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

341

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

344

```
345 <annotation>
```

```
346 <documentation> http://www.projectliberty.org/schemas/authctx/classes/MobileContract
```

```
347 </documentation>
```

```
348 </annotation>
```

```
349 <xs:element name="AuthenticationContextStatement">
```

```
350   <xs:complexType>
```

```
351     <xs:sequence>
```

```
352       <xs:element minOccurs="1" maxOccurs="1" ref="Identification"/>
```

```
353       <xs:element minOccurs="1" maxOccurs="1" ref="TechnicalProtection"/>
```

```
354       <xs:element minOccurs="1" maxOccurs="1" ref="AuthenticationMethod"/>
```

```
355       <xs:element minOccurs="1" maxOccurs="1" ref="OperationalProtection"/>
```

```
356       <xs:element minOccurs="1" maxOccurs="1" ref="GoverningAgreements"/>
```

```
357         <xs:element ref="AC:extension" minOccurs="0"
```

```
358         maxOccurs="unbounded"/></xs:sequence>
```

```
359     </xs:complexType>
```

```
360 </xs:element>
```

```
361 <xs:element name="AuthenticationMethod">
```

```
362   <xs:complexType>
```

```
363     <xs:sequence>
```

```
364       <xs:element minOccurs="1" maxOccurs="1" ref="Authenticator"/>
```

```
365       <xs:element minOccurs="1" maxOccurs="1"
```

```
366       ref="AuthenticatorTransportProtocol"/>
```

```
367         <xs:element ref="AC:extension" minOccurs="0"
```

```
368         maxOccurs="unbounded"/></xs:sequence>
```

```
369     </xs:complexType>
```

```
370 </xs:element>
```

371

```
372 <xs:element name="Authenticator">
```

```
373   <xs:complexType>
```

```
374     <xs:sequence>
```

```
375     <xs:element minOccurs="1" maxOccurs="1" ref="SharedSecretChallengeResponse"/>
376     <xs:element ref="AC:extension" minOccurs="0"
377     maxOccurs="unbounded"/></xs:sequence>
378 </xs:complexType>
379 </xs:element>
380
381 <xs:element name="AuthenticatorTransportProtocol">
382 <xs:complexType>
383 <xs:sequence>
384 <xs:element minOccurs="1" maxOccurs="1" ref="MobileNetwork"/>
385 <xs:element ref="AC:extension" minOccurs="0" maxOccurs="unbounded"/>
386 </xs:sequence>
387 </xs:complexType>
388 </xs:element>
389
390 <xs:element name="DeactivationCallCenter">
391 <xs:complexType>
392 <xs:sequence>
393 <xs:element ref="AC:extension" minOccurs="0"
394 maxOccurs="unbounded"/>
395 </xs:sequence>
396 </xs:complexType>
397 </xs:element>
398
399 <xs:element name="GoverningAgreementRef">
400 <xs:complexType>
401 <xs:attribute name="ref"
402 fixed="http://SomeMobileforum.org/namespaces/authcontext/classes/Mobile-
403 Class2.pdf"/>
404 </xs:complexType>
405 </xs:element>
406 <xs:element name="GoverningAgreements">
407 <xs:complexType>
408 <xs:sequence>
409 <xs:element minOccurs="1" maxOccurs="1" ref="GoverningAgreementRef"/>
410 <xs:element ref="AC:extension" minOccurs="0"
411 maxOccurs="unbounded"/></xs:sequence>
412 </xs:complexType>
413 </xs:element>
414 <xs:element name="Identification">
415 <xs:complexType>
416 <xs:sequence>
417 <xs:element minOccurs="1" maxOccurs="1" ref="PhysicalVerification"/>
418 <xs:element ref="AC:extension" minOccurs="0"
419 maxOccurs="unbounded"/></xs:sequence>
420 <xs:attribute name="nym" type="xs:string" use="required"/>
421 </xs:complexType>
422 </xs:element>
423 <xs:element name="MobileAuthCard">
424 <xs:complexType>
425 <xs:sequence>
426 <xs:element ref="AC:extension" minOccurs="0"
427 maxOccurs="unbounded"/>
428 </xs:sequence>
429 </xs:complexType>
430 </xs:element>
431 <xs:element name="MobileDevice">
432 <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
433 maxOccurs="unbounded"/></xs:sequence></xs:complexType>
434 </xs:element>
435 <xs:element name="MobileNetwork">
436 <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
437 maxOccurs="unbounded"/></xs:sequence></xs:complexType>
438 </xs:element>
439 <xs:element name="OperationalProtection">
440 <xs:complexType>
441 <xs:sequence>
```

```
442     <xs:element minOccurs="1" maxOccurs="1" ref="SecurityAudit"/>
443     <xs:element minOccurs="1" maxOccurs="1" ref="DeactivationCallCenter"/>
444     <xs:element ref="AC:extension" minOccurs="0"
445     maxOccurs="unbounded"/></xs:sequence>
446   </xs:complexType>
447 </xs:element>
448
449 <xs:element name="PhysicalVerification">
450   <xs:complexType>
451     <xs:attribute name="credentialLevel" type="xs:string" use="required"/>
452   </xs:complexType>
453 </xs:element>
454
455 <xs:element name="SecurityAudit">
456   <xs:complexType>
457     <xs:sequence>
458       <xs:element minOccurs="1" maxOccurs="1" ref="SwitchAudit"/>
459       <xs:element ref="AC:extension" minOccurs="0"
460       maxOccurs="unbounded"/></xs:sequence>
461     </xs:complexType>
462 </xs:element>
463 <xs:element name="SharedKeyProtection">
464   <xs:complexType>
465     <xs:choice>
466       <xs:element minOccurs="1" maxOccurs="1" ref="MobileAuthCard"/>
467       <xs:element minOccurs="1" maxOccurs="1" ref="MobileDevice"/>
468     </xs:choice>
469   </xs:complexType>
470 </xs:element>
471 <xs:element name="SharedSecretChallengeResponse">
472   <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
473   maxOccurs="unbounded"/></xs:sequence></xs:complexType>
474 </xs:element>
475 <xs:element name="SwitchAudit">
476   <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
477   maxOccurs="unbounded"/></xs:sequence></xs:complexType>
478 </xs:element>
479 <xs:element name="TechnicalProtection">
480   <xs:complexType>
481     <xs:sequence>
482       <xs:element minOccurs="1" maxOccurs="1" ref="SharedKeyProtection"/>
483       <xs:element ref="AC:extension" minOccurs="0"
484       maxOccurs="unbounded"/></xs:sequence>
485     </xs:complexType>
486 </xs:element>
487 </xs:schema>
```

488 5.1.2 MobileDigitalID

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

491 5.1.2.1 Associated Liberty URI

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

493 5.1.2.2 Class Schema

```
494 <?xml version="1.0" encoding="UTF-8"?>
495
496 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
497   elementFormDefault="qualified">
498
499 <annotation>
```

```
501 <documentation> http://www.projectliberty.org/schemas/authctx/classes/MobileDigitalID
502 </documentation>
503 </annotation>
504
505 <xs:element name="AuthenticationContextStatement">
506 <xs:complexType>
507 <xs:sequence>
508 <xs:element ref="Identification"/>
509 <xs:element ref="TechnicalProtection"/>
510 <xs:element ref="AuthenticationMethod"/>
511 <xs:element ref="OperationalProtection"/>
512 <xs:element ref="GoverningAgreements"/>
513 <xs:element ref="AC:extension" minOccurs="0"
514 maxOccurs="unbounded"/></xs:sequence>
515 </xs:complexType>
516 </xs:element>
517 <xs:element name="AuthenticationMethod">
518 <xs:complexType>
519 <xs:sequence>
520 <xs:element ref="Authenticator"/>
521 <xs:element ref="AuthenticatorTransportProtocol"/>
522 <xs:element ref="AC:extension" minOccurs="0"
523 maxOccurs="unbounded"/></xs:sequence>
524 </xs:complexType>
525 </xs:element>
526 <xs:element name="Authenticator">
527 <xs:complexType>
528 <xs:choice>
529 <xs:element ref="Dig-sig"/>
530 <xs:element ref="ZeroKnowledge"/>
531 </xs:choice>
532 </xs:complexType>
533 </xs:element>
534 <xs:element name="AuthenticatorTransportProtocol">
535 <xs:complexType>
536 <xs:choice>
537 <xs:element ref="MobileNetwork"/>
538 <xs:element ref="SSL"/>
539 <xs:element ref="WTLS"/>
540 <xs:element ref="IPSec"/>
541 </xs:choice>
542 </xs:complexType>
543 </xs:element>
544 <xs:element name="DeactivationCallCenter">
545 <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
546 maxOccurs="unbounded"/></xs:sequence></xs:complexType>
547 </xs:element>
548 <xs:element name="Dig-sig">
549 <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
550 maxOccurs="unbounded"/></xs:sequence></xs:complexType>
551 </xs:element>
552 <xs:element name="GoverningAgreementRef">
553 <xs:complexType>
554 <xs:attribute name="ref"
555 fixed="http://SomeMobileforum.org/namespaces/authcontext/classes/Mobile-
556 Class3.pdf"/>
557 </xs:complexType>
558 </xs:element>
559 <xs:element name="GoverningAgreements">
560 <xs:complexType>
561 <xs:sequence>
562 <xs:element ref="GoverningAgreementRef"/>
563 <xs:element ref="AC:extension" minOccurs="0"
564 maxOccurs="unbounded"/></xs:sequence>
565 </xs:complexType>
566 </xs:element>
567 <xs:element name="IPSec">
```

```
568     <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
569     maxOccurs="unbounded" /></xs:sequence></xs:complexType>
570 </xs:element>
571 <xs:element name="Identification">
572   <xs:complexType>
573     <xs:sequence>
574       <xs:element ref="PhysicalVerification" />
575       <xs:element ref="WrittenConsent" />
576       <xs:element ref="AC:extension" minOccurs="0"
577     maxOccurs="unbounded" /></xs:sequence>
578     <xs:attribute name="nym" type="xs:string" use="required" />
579   </xs:complexType>
580 </xs:element>
581 <xs:element name="KeyStorage">
582   <xs:complexType>
583     <xs:attribute name="medium" type="xs:string" use="required" />
584   </xs:complexType>
585 </xs:element>
586 <xs:element name="MobileNetwork">
587   <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
588     maxOccurs="unbounded" /></xs:sequence></xs:complexType>
589 </xs:element>
590 <xs:element name="OperationalProtection">
591   <xs:complexType>
592     <xs:sequence>
593       <xs:element ref="SecurityAudit" />
594       <xs:element ref="DeactivationCallCenter" />
595       <xs:element ref="AC:extension" minOccurs="0"
596     maxOccurs="unbounded" /></xs:sequence>
597   </xs:complexType>
598 </xs:element>
599 <xs:element name="PhysicalVerification">
600   <xs:complexType>
601     <xs:attribute name="credentialLevel" type="xs:string" use="required" />
602   </xs:complexType>
603 </xs:element>
604 <xs:element name="PrivateKeyProtection">
605   <xs:complexType>
606     <xs:sequence>
607       <xs:element ref="KeyStorage" />
608       <xs:element ref="AC:extension" minOccurs="0"
609     maxOccurs="unbounded" /></xs:sequence>
610   </xs:complexType>
611 </xs:element>
612 <xs:element name="SSL">
613   <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
614     maxOccurs="unbounded" /></xs:sequence></xs:complexType>
615 </xs:element>
616 <xs:element name="SecurityAudit">
617   <xs:complexType>
618     <xs:sequence>
619       <xs:element ref="SwitchAudit" />
620       <xs:element ref="AC:extension" minOccurs="0"
621     maxOccurs="unbounded" /></xs:sequence>
622   </xs:complexType>
623 </xs:element>
624 <xs:element name="SwitchAudit">
625   <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
626     maxOccurs="unbounded" /></xs:sequence></xs:complexType>
627 </xs:element>
628 <xs:element name="TechnicalProtection">
629   <xs:complexType>
630     <xs:sequence>
631       <xs:element ref="PrivateKeyProtection" />
632       <xs:element ref="AC:extension" minOccurs="0"
633     maxOccurs="unbounded" /></xs:sequence>
634   </xs:complexType>
```

```
635 </xs:element>
636 <xs:element name="WTLS">
637   <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
638     maxOccurs="unbounded" /></xs:sequence></xs:complexType>
639 </xs:element>
640 <xs:element name="WrittenConsent">
641   <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
642     maxOccurs="unbounded" /></xs:sequence></xs:complexType>
643 </xs:element>
644 <xs:element name="ZeroKnowledge">
645   <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
646     maxOccurs="unbounded" /></xs:sequence></xs:complexType>
647 </xs:element>
648 </xs:schema>
```

649 5.1.3 MobileUnregistered

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

651 5.1.3.1 Associated Liberty URI

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

653 5.1.3.2 Class Schema

```
654
655 <?xml version="1.0" encoding="UTF-8"?>
656
657 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
658   elementFormDefault="qualified">
659
660 <annotation>
661 <documentation>
662   http://www.projectliberty.org/schemas/authctx/classes/MobileUnregistered
663 </documentation>
664 </annotation>
665
666 <xs:element name="AuthenticationContextStatement">
667   <xs:complexType>
668     <xs:sequence>
669       <xs:element ref="TechnicalProtection"/>
670       <xs:element ref="AuthenticationMethod"/>
671       <xs:element ref="OperationalProtection"/>
672       <xs:element ref="GoverningAgreements"/>
673       <xs:element ref="AC:extension" minOccurs="0"
674         maxOccurs="unbounded" /></xs:sequence>
675     </xs:complexType>
676   </xs:element>
677 <xs:element name="AuthenticationMethod">
678   <xs:complexType>
679     <xs:sequence>
680       <xs:element ref="Authenticator"/>
681       <xs:element ref="AuthenticatorTransportProtocol"/>
682       <xs:element ref="AC:extension" minOccurs="0"
683         maxOccurs="unbounded" /></xs:sequence>
684     </xs:complexType>
685   </xs:element>
686 <xs:element name="Authenticator">
687   <xs:complexType>
688     <xs:sequence>
689       <xs:element ref="SharedSecretChallengeResponse"/>
690       <xs:element ref="AC:extension" minOccurs="0"
691         maxOccurs="unbounded" /></xs:sequence>
692     </xs:complexType>
693   </xs:element>
```



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

```
761 </xs:element>
762 <xs:element name="SwitchAudit">
763   <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
764     maxOccurs="unbounded" /></xs:sequence></xs:complexType>
765 </xs:element>
766 <xs:element name="TechnicalProtection">
767   <xs:complexType>
768     <xs:sequence>
769       <xs:element ref="SharedKeyProtection" />
770       <xs:element ref="AC:extension" minOccurs="0"
771         maxOccurs="unbounded" /></xs:sequence>
772     </xs:complexType>
773 </xs:element>
774 </xs:schema>
```

775 5.1.4 Password

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

778 5.1.4.1 Associated Liberty URI

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

780 5.1.4.2 Class Schema

```
781 <?xml version="1.0" encoding="UTF-8"?>
782
783 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
784   elementFormDefault="qualified">
785
786 <annotation>
787 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Password
788   </documentation>
789 </annotation>
790
791   <xs:element name="AuthenticationContextStatement">
792     <xs:complexType>
793       <xs:sequence>
794         <xs:element ref="AuthenticationMethod" />
795         <xs:element ref="AC:extension" minOccurs="0"
796           maxOccurs="unbounded" /></xs:sequence>
797       </xs:complexType>
798     </xs:element>
799   <xs:element name="AuthenticationMethod">
800     <xs:complexType>
801       <xs:all>
802         <xs:element ref="PrincipalAuthenticationMechanism" />
803         <xs:element ref="AuthenticatorTransportProtocol" />
804       </xs:all>
805     </xs:complexType>
806   </xs:element>
807   <xs:element name="AuthenticatorTransportProtocol">
808     <xs:complexType>
809       <xs:sequence>
810         <xs:element ref="HTTP" />
811         <xs:element ref="AC:extension" minOccurs="0"
812           maxOccurs="unbounded" /></xs:sequence>
813       </xs:complexType>
814     </xs:element>
815   <xs:element name="HTTP">
816     <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
817       maxOccurs="unbounded" /></xs:sequence></xs:complexType>
818   </xs:element>
819   <xs:element name="Length">
```

```
820 <xs:complexType>
821 <xs:attribute name="min" fixed="3" />
822 </xs:complexType>
823 </xs:element>
824 <xs:element name="Password">
825 <xs:complexType>
826 <xs:sequence>
827 <xs:element ref="Length" />
828 <xs:element ref="AC:extension" minOccurs="0"
829 maxOccurs="unbounded" /></xs:sequence>
830 </xs:complexType>
831 </xs:element>
832 <xs:element name="PrincipalAuthenticationMechanism">
833 <xs:complexType>
834 <xs:sequence>
835 <xs:element ref="Password" />
836 <xs:element ref="AC:extension" minOccurs="0"
837 maxOccurs="unbounded" /></xs:sequence>
838 </xs:complexType>
839 </xs:element>
840 </xs:schema>
```

841 5.1.5 Password- ProtectedTransport

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

844 5.1.5.1 Associated Liberty URI

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

846 5.1.5.2 Class Schema

```
847
848 <?xml version="1.0" encoding="UTF-8"?>
849
850 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
851 elementFormDefault="qualified">
852
853 <annotation>
854 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Password-
855 ProtectedTransport </documentation>
856 </annotation>
857
858 <xs:element name="AuthenticationContextStatement">
859 <xs:complexType>
860 <xs:sequence>
861 <xs:element ref="AuthenticationMethod" />
862 <xs:element ref="AC:extension" minOccurs="0"
863 maxOccurs="unbounded" /></xs:sequence>
864 </xs:complexType>
865 </xs:element>
866 <xs:element name="AuthenticationMethod">
867 <xs:complexType>
868 <xs:all>
869 <xs:element ref="PrincipalAuthenticationMechanism" />
870 <xs:element ref="AuthenticatorTransportProtocol" />
871 </xs:all>
872 </xs:complexType>
873 </xs:element>
874 <xs:element name="AuthenticatorTransportProtocol">
875 <xs:complexType>
876 <xs:sequence>
877 <xs:element ref="SSL" />
878 <xs:element ref="AC:extension" minOccurs="0"
879 maxOccurs="unbounded" /></xs:sequence>
```

```
880     </xs:complexType>
881 </xs:element>
882 <xs:element name="Length">
883   <xs:complexType>
884     <xs:attribute name="min" fixed="3" />
885   </xs:complexType>
886 </xs:element>
887 <xs:element name="Password">
888   <xs:complexType>
889     <xs:sequence>
890       <xs:element ref="Length" />
891       <xs:element ref="AC:extension" minOccurs="0"
892         maxOccurs="unbounded" /></xs:sequence>
893     </xs:complexType>
894 </xs:element>
895 <xs:element name="PrincipalAuthenticationMechanism">
896   <xs:complexType>
897     <xs:sequence>
898       <xs:element ref="Password" />
899       <xs:element ref="AC:extension" minOccurs="0"
900         maxOccurs="unbounded" /></xs:sequence>
901     </xs:complexType>
902 </xs:element>
903 <xs:element name="SSL">
904   <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
905     maxOccurs="unbounded" /></xs:sequence></xs:complexType>
906 </xs:element>
907 </xs:schema>
```

908 5.1.6 Previous-Session

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

913
914 The context for the previously authenticated session is explicitly not included in this context class because the user has
915 not authenticated during this session, and so the mechanism that the user employed to authenticate in a previous
916 session should not be used as part of a decision on whether to *now* allow access to a resource.

917 5.1.6.1 Associated Liberty URI

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

919 5.1.6.2 Class Schema

```
920 <?xml version="1.0" encoding="UTF-8"?>
921 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
922   elementFormDefault="qualified">
923 <annotation>
924 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Previous-
925   Session </documentation>
926 </annotation>
927 <xs:element name="AuthenticationContextStatement">
928   <xs:complexType>
929     <xs:sequence>
930       <xs:element minOccurs="1" maxOccurs="1" ref="AuthenticationMethod" />
931       <xs:element ref="AC:extension" minOccurs="0"
932         maxOccurs="unbounded" /></xs:sequence>
933     </xs:complexType>
934 </xs:element>
```

```
939
940 <xs:element name="AuthenticationMethod">
941   <xs:complexType>
942     <xs:sequence>
943       <xs:element ref="Authenticator" minOccurs="0" maxOccurs="1"/>
944       <xs:element ref="AC:extension" minOccurs="0"
945         maxOccurs="unbounded"/></xs:sequence>
946     </xs:complexType>
947   </xs:element>
948
949 <xs:element name="Authenticator">
950   <xs:complexType>
951     <xs:sequence>
952       <xs:element minOccurs="1" maxOccurs="1" ref="PreviousSession"/>
953       <xs:element ref="AC:extension" minOccurs="0"
954         maxOccurs="unbounded"/></xs:sequence>
955     </xs:complexType>
956   </xs:element>
957
958 <xs:element name="PreviousSession">
959   <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
960     maxOccurs="unbounded"/></xs:sequence></xs:complexType>
961 </xs:element>
962
963 </xs:schema>
964
```

965 5.1.7 Smartcard

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

967 5.1.7.1 Associated Liberty URI

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

969 5.1.7.2 Class Schema

```
970 <?xml version="1.0" encoding="UTF-8"?>
971
972 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
973   elementFormDefault="qualified">
974
975   <annotation>
976     <documentation> http://www.projectliberty.org/schemas/authctx/classes/Smartcard
977     </documentation>
978   </annotation>
979
980   <xs:element name="AuthenticationContextStatement">
981     <xs:complexType>
982       <xs:sequence>
983         <xs:element minOccurs="1" maxOccurs="1" ref="AuthenticationMethod"/>
984         <xs:element ref="AC:extension" minOccurs="0"
985           maxOccurs="unbounded"/></xs:sequence>
986       </xs:complexType>
987     </xs:element>
988
989     <xs:element name="AuthenticationMethod">
990       <xs:complexType>
991         <xs:sequence>
992           <xs:element minOccurs="1" maxOccurs="1"
993             ref="PrincipalAuthenticationMechanism"/>
994           <xs:element ref="AC:extension" minOccurs="0"
995             maxOccurs="unbounded"/></xs:sequence>
996         </xs:complexType>
997       </xs:element>
998
999     <xs:element name="PrincipalAuthenticationMechanism">
1000       <xs:complexType>
```

```
999     <xs:sequence>
1000         <xs:element minOccurs="1" maxOccurs="1" ref="Smartcard"/>
1001         <xs:element ref="AC:extension" minOccurs="0"
1002             maxOccurs="unbounded" /></xs:sequence>
1003     </xs:complexType>
1004 </xs:element>
1005 <xs:element name="Smartcard">
1006     <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
1007         maxOccurs="unbounded" /></xs:sequence></xs:complexType>
1008 </xs:element>
1009 </xs:schema>
1010
```

1011 5.1.8 Smartcard-PKI

1012 The Smartcard-PKI class is identified when a Principal authenticates to an identity provider through a two-factor
1013 authentication mechanism using a smartcard with enclosed private key and a PIN.

1014 5.1.8.1 Associated Liberty URI

1015 <http://www.projectliberty.org/schemas/authctx/classes/Smartcard-PKI>

1016 5.1.8.2 Class Schema

```
1017 <?xml version="1.0" encoding="UTF-8"?>
1018
1019 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
1020     elementFormDefault="qualified">
1021
1022     <annotation>
1023     <documentation> http://www.projectliberty.org/schemas/authctx/classes/Smartcard-
1024     PKI</documentation>
1025     </annotation>
1026
1027     <xs:element name="AuthenticationContextStatement">
1028     <xs:complexType>
1029     <xs:sequence>
1030         <xs:element minOccurs="1" maxOccurs="1" ref="TechnicalProtection"/>
1031         <xs:element minOccurs="1" maxOccurs="1" ref="AuthenticationMethod"/>
1032         <xs:element ref="AC:extension" minOccurs="0"
1033             maxOccurs="unbounded" /></xs:sequence>
1034     </xs:complexType>
1035 </xs:element>
1036 <xs:element name="AuthenticationMethod">
1037     <xs:complexType>
1038     <xs:sequence>
1039         <xs:element minOccurs="1" maxOccurs="1"
1040             ref="PrincipalAuthenticationMechanism"/>
1041         <xs:element minOccurs="1" maxOccurs="1" ref="Authenticator"/>
1042         <xs:element minOccurs="1" maxOccurs="1"
1043             ref="AuthenticatorTransportProtocol"/>
1044         <xs:element ref="AC:extension" minOccurs="0"
1045             maxOccurs="unbounded" /></xs:sequence>
1046     </xs:complexType>
1047 </xs:element>
1048 <xs:element name="Authenticator">
1049     <xs:complexType>
1050     <xs:sequence>
1051         <xs:element minOccurs="1" maxOccurs="1" ref="Dig-sig"/>
1052         <xs:element ref="AC:extension" minOccurs="0"
1053             maxOccurs="unbounded" /></xs:sequence>
1054     </xs:complexType>
1055 </xs:element>
1056 <xs:element name="AuthenticatorTransportProtocol">
1057     <xs:complexType>
```

```
1058     <xs:sequence>
1059         <xs:element minOccurs="1" maxOccurs="1" ref="SSL" />
1060         <xs:element ref="AC:extension" minOccurs="0"
1061 maxOccurs="unbounded" /></xs:sequence>
1062     </xs:complexType>
1063 </xs:element>
1064 <xs:element name="Dig-sig">
1065     <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
1066 maxOccurs="unbounded" /></xs:sequence></xs:complexType>
1067 </xs:element>
1068 <xs:element name="KeyActivation">
1069     <xs:complexType>
1070     <xs:sequence>
1071         <xs:element minOccurs="1" maxOccurs="1" ref="Password" />
1072         <xs:element ref="AC:extension" minOccurs="0"
1073 maxOccurs="unbounded" /></xs:sequence>
1074     </xs:complexType>
1075 </xs:element>
1076 <xs:element name="Length">
1077     <xs:complexType>
1078         <xs:attribute name="min" type="xs:byte" use="required" />
1079     </xs:complexType>
1080 </xs:element>
1081 <xs:element name="Password">
1082     <xs:complexType>
1083     <xs:sequence>
1084         <xs:element minOccurs="1" maxOccurs="1" ref="Length" />
1085         <xs:element ref="AC:extension" minOccurs="0"
1086 maxOccurs="unbounded" /></xs:sequence>
1087     </xs:complexType>
1088 </xs:element>
1089 <xs:element name="PrincipalAuthenticationMechanism">
1090     <xs:complexType>
1091     <xs:sequence>
1092         <xs:element minOccurs="1" maxOccurs="1" ref="Smartcard" />
1093         <xs:element ref="AC:extension" minOccurs="0"
1094 maxOccurs="unbounded" /></xs:sequence>
1095     </xs:complexType>
1096 </xs:element>
1097 <xs:element name="PrivateKeyProtection">
1098     <xs:complexType>
1099     <xs:sequence>
1100         <xs:element minOccurs="1" maxOccurs="1" ref="KeyActivation" />
1101         <xs:element ref="AC:extension" minOccurs="0"
1102 maxOccurs="unbounded" /></xs:sequence>
1103     </xs:complexType>
1104 </xs:element>
1105 <xs:element name="SSL">
1106     <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
1107 maxOccurs="unbounded" /></xs:sequence></xs:complexType>
1108 </xs:element>
1109 <xs:element name="Smartcard">
1110     <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
1111 maxOccurs="unbounded" /></xs:sequence></xs:complexType>
1112 </xs:element>
1113 <xs:element name="TechnicalProtection">
1114     <xs:complexType>
1115     <xs:sequence>
1116         <xs:element minOccurs="1" maxOccurs="1" ref="PrivateKeyProtection" />
1117         <xs:element ref="AC:extension" minOccurs="0"
1118 maxOccurs="unbounded" /></xs:sequence>
1119     </xs:complexType>
1120 </xs:element>
1121 </xs:schema>
1122
```

1123 5.1.9 Software-PKI

1124 The Software-PKI class is identified when a Principal uses an X.509 certificate stored in software to authenticate to
1125 the identity provider over an SSL protected session.
1126

1127 5.1.9.1 Associated Liberty URI

1128

1129 <http://www.projectliberty.org/schemas/authctx/classes/Software-PKI>

1130 5.1.9.2 Class Schema

```
1131 <?xml version="1.0" encoding="UTF-8"?>
1132 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
1133           elementFormDefault="qualified">
1134 <annotation>
1135 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Software-PKI
1136 </documentation>
1137 </annotation>
1138 <xs:element name="AuthenticationContextStatement">
1139 <xs:complexType>
1140 <xs:sequence>
1141 <xs:element minOccurs="1" maxOccurs="1" ref="AuthenticationMethod"/>
1142 <xs:element ref="AC:extension" minOccurs="0"
1143 maxOccurs="unbounded"/></xs:sequence>
1144 </xs:complexType>
1145 </xs:element>
1146 <xs:element name="AuthenticationMethod">
1147 <xs:complexType>
1148 <xs:sequence>
1149 <xs:element minOccurs="1" maxOccurs="1"
1150 ref="PrincipalAuthenticationMechanism"/>
1151 <xs:element minOccurs="1" maxOccurs="1" ref="Authenticator"/>
1152 <xs:element minOccurs="1" maxOccurs="1"
1153 ref="AuthenticatorTransportProtocol"/>
1154 <xs:element ref="AC:extension" minOccurs="0"
1155 maxOccurs="unbounded"/></xs:sequence>
1156 </xs:complexType>
1157 </xs:element>
1158 <xs:element name="Authenticator">
1159 <xs:complexType>
1160 <xs:sequence>
1161 <xs:element minOccurs="1" maxOccurs="1" ref="Dig-sig"/>
1162 <xs:element ref="AC:extension" minOccurs="0"
1163 maxOccurs="unbounded"/></xs:sequence>
1164 </xs:complexType>
1165 </xs:element>
1166 <xs:element name="AuthenticatorTransportProtocol">
1167 <xs:complexType>
1168 <xs:sequence>
1169 <xs:element minOccurs="1" maxOccurs="1" ref="SSL"/>
1170 <xs:element ref="AC:extension" minOccurs="0"
1171 maxOccurs="unbounded"/></xs:sequence>
1172 </xs:complexType>
1173 </xs:element>
1174 <xs:element name="Dig-sig">
1175 <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
1176 maxOccurs="unbounded"/></xs:sequence></xs:complexType>
1177 </xs:element>
```



```
1182 <xs:element name="Password">
1183   <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
1184     maxOccurs="unbounded" /></xs:sequence></xs:complexType>
1185 </xs:element>
1186 <xs:element name="PrincipalAuthenticationMechanism">
1187   <xs:complexType>
1188     <xs:sequence>
1189       <xs:element minOccurs="1" maxOccurs="1" ref="Password"/>
1190       <xs:element ref="AC:extension" minOccurs="0"
1191         maxOccurs="unbounded" /></xs:sequence>
1192     </xs:complexType>
1193 </xs:element>
1194 <xs:element name="SSL">
1195   <xs:complexType><xs:sequence><xs:element ref="AC:extension" minOccurs="0"
1196     maxOccurs="unbounded" /></xs:sequence></xs:complexType>
1197 </xs:element>
1198 </xs:schema>
1199
```

1200 5.1.10 Time-Sync-Token

1201 The Time-Sync-Token class is identified when a Principal authenticates through a time synchronization token.

1202 5.1.10.1 Associated Liberty URI

1203 <http://www.projectliberty.org/schemas/authctx/classes/Time-Sync-Token>

1204 5.1.10.2 Class Schema

```
1205
1206 <?xml version="1.0" encoding="UTF-8"?>
1207
1208 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
1209   elementFormDefault="qualified">
1210
1211 <annotation>
1212 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Time-Sync-Token
1213 </documentation>
1214 </annotation>
1215
1216 <xs:element name="AuthenticationContextStatement">
1217   <xs:complexType>
1218     <xs:sequence>
1219       <xs:element minOccurs="1" maxOccurs="1" ref="AuthenticationMethod"/>
1220       <xs:element ref="AC:extension" minOccurs="0"
1221         maxOccurs="unbounded" /></xs:sequence>
1222     </xs:complexType>
1223 </xs:element>
1224 <xs:element name="AuthenticationMethod">
1225   <xs:complexType>
1226     <xs:sequence>
1227       <xs:element minOccurs="1" maxOccurs="1"
1228         ref="PrincipalAuthenticationMechanism"/>
1229       <xs:element ref="AC:extension" minOccurs="0"
1230         maxOccurs="unbounded" /></xs:sequence>
1231     </xs:complexType>
1232 </xs:element>
1233 <xs:element name="Generation">
1234   <xs:complexType>
1235     <xs:attribute name="mechanism" fixed="principalchosen" />
1236   </xs:complexType>
1237 </xs:element>
1238
1239 <xs:element name="PrincipalAuthenticationMechanism">
1240   <xs:complexType>
1241     <xs:sequence>
```

```
1242     <xs:element minOccurs="1" maxOccurs="1" ref="Token" />
1243     <xs:element ref="AC:extension" minOccurs="0"
1244     maxOccurs="unbounded" /></xs:sequence>
1245   </xs:complexType>
1246 </xs:element>
1247 <xs:element name="TimeSyncToken">
1248   <xs:complexType>
1249     <xs:attribute name="deviceType" fixed="hardware" />
1250     <xs:attribute name="seedLength" fixed="64" />
1251     <xs:attribute name="deviceInHand" fixed="true" />
1252   </xs:complexType>
1253 </xs:element>
1254 <xs:element name="Token">
1255   <xs:complexType>
1256     <xs:sequence>
1257       <xs:element minOccurs="1" maxOccurs="1" ref="TimeSyncToken" />
1258       <xs:element ref="AC:extension" minOccurs="0"
1259       maxOccurs="unbounded" /></xs:sequence>
1260     </xs:complexType>
1261   </xs:element>
1262 </xs:schema>
1263
```

1264 5.1.11 Internet Protocol

1265 The Internet Protocol class is identified when a Principal is authenticated through the use of a provided IP address.

1266 5.1.11.1 Associated Liberty URI

1267 <http://www.projectliberty.org/schemas/authctx/classes/IAP-IPAddress>

1268 5.1.11.2 Class Schema

```
1269 <?xml version="1.0" encoding="UTF-8"?>
1270 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
1271   elementFormDefault="qualified">
1272   <xs:annotation>
1273     <xs:documentation>http://www.projectliberty.org/schemas/authctx/classes/IAP-
1274     IPAddress</xs:documentation>
1275   </xs:annotation>
1276   <xs:element name="AuthenticationContextStatement">
1277     <xs:complexType>
1278       <xs:sequence>
1279         <xs:element ref="AuthenticationMethod" />
1280         <xs:element ref="AC:extension" minOccurs="0" maxOccurs="unbounded" />
1281       </xs:sequence>
1282     </xs:complexType>
1283   </xs:element>
1284   <xs:element name="AuthenticationMethod">
1285     <xs:complexType>
1286       <xs:sequence>
1287         <xs:element ref="Authenticator" />
1288         <xs:element ref="AC:extension" minOccurs="0" maxOccurs="unbounded" />
1289       </xs:sequence>
1290     </xs:complexType>
1291   </xs:element>
1292   <xs:element name="Authenticator">
1293     <xs:complexType>
1294       <xs:sequence>
1295         <xs:element ref="IPAddress" />
1296         <xs:element ref="AC:extension" minOccurs="0" maxOccurs="unbounded" />
1297       </xs:sequence>
1298     </xs:complexType>
1299   </xs:element>
1300   <xs:element name="IPAddress">
1301     <xs:complexType>
```

```
1302     <xs:sequence>
1303         <xs:element ref="AC:extension" minOccurs="0" maxOccurs="unbounded"/>
1304     </xs:sequence>
1305 </xs:complexType>
1306 </xs:element>
1307 </xs:schema>
1308
```

1309 5.1.12 Internet Protocol + Password

1310 The Internet Protocol Password class is identified when a Principal is authenticated through the use of a provided IP
1311 address, in addition to username/password.

1312 5.1.12.1 Associated Liberty URI

1313 <http://www.projectliberty.org/schemas/authctx/classes/IAP-Password>

1314 5.1.12.2 Class Schema

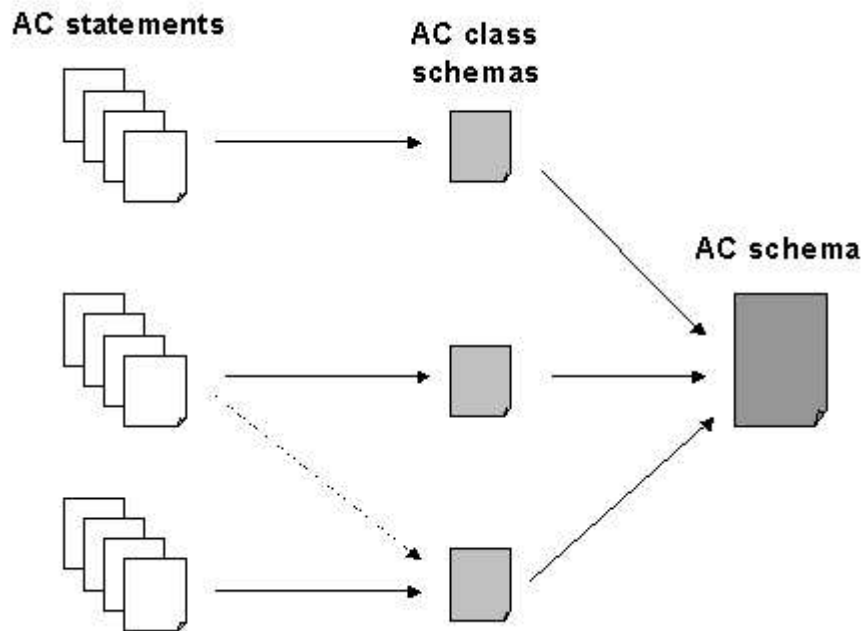
```
1315 <?xml version="1.0" encoding="UTF-8"?>
1316 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
1317     elementFormDefault="qualified">
1318     <xs:annotation>
1319         <xs:documentation>http://www.projectliberty.org/schemas/authctx/classes/IAP-
1320 Password</xs:documentation>
1321     </xs:annotation>
1322     <xs:element name="AuthenticationContextStatement">
1323         <xs:complexType>
1324             <xs:sequence>
1325                 <xs:element ref="AuthenticationMethod"/>
1326                 <xs:element ref="AC:extension" minOccurs="0" maxOccurs="unbounded"/>
1327             </xs:sequence>
1328         </xs:complexType>
1329     </xs:element>
1330     <xs:element name="AuthenticationMethod">
1331         <xs:complexType>
1332             <xs:sequence>
1333                 <xs:element ref="Authenticator"/>
1334                 <xs:element ref="PrincipalAuthenticationMechanism"/>
1335                 <xs:element ref="AC:extension" minOccurs="0" maxOccurs="unbounded"/>
1336             </xs:sequence>
1337         </xs:complexType>
1338     </xs:element>
1339     <xs:element name="Authenticator">
1340         <xs:complexType>
1341             <xs:sequence>
1342                 <xs:element ref="IPAddress"/>
1343                 <xs:element ref="AC:extension" minOccurs="0" maxOccurs="unbounded"/>
1344             </xs:sequence>
1345         </xs:complexType>
1346     </xs:element>
1347     <xs:element name="IPAddress">
1348         <xs:complexType>
1349             <xs:sequence>
1350                 <xs:element ref="AC:extension" minOccurs="0" maxOccurs="unbounded"/>
1351             </xs:sequence>
1352         </xs:complexType>
1353     </xs:element>
1354     <xs:element name="PrincipalAuthenticationMechanism">
1355         <xs:complexType>
1356             <xs:sequence>
1357                 <xs:element ref="Password"/>
1358                 <xs:element ref="AC:extension" minOccurs="0" maxOccurs="unbounded"/>
1359             </xs:sequence>
1360         </xs:complexType>
1361     </xs:element>
```

```
1362 <xs:element name="Length">
1363   <xs:complexType>
1364     <xs:attribute name="min" fixed="3"/>
1365   </xs:complexType>
1366 </xs:element>
1367 <xs:element name="Password">
1368   <xs:complexType>
1369     <xs:sequence>
1370       <xs:element ref="Length"/>
1371       <xs:element ref="AC:extension" minOccurs="0" maxOccurs="unbounded"/>
1372     </xs:sequence>
1373   </xs:complexType>
1374 </xs:element>
1375 </xs:schema>
```

1376 5.2 Authentication Context Schema

1377 The relationship between authentication context statements, authentication context classes, and the authentication
1378 context XML schema is shown in Figure 3.

1379



1380

1381 **Figure 3: Relationship between authentication context statements, classes, and XML schema**

1382

1383 Authentication context statements may conform to authentication context classes, which are themselves logical subsets
1384 of the authentication context XML schema.

1385

1386 5.2.1 XML Schema

```
1387 <?xml version="1.0" encoding="UTF-8"?>
1388 <xs:schema targetNamespace="urn:liberty:ac:1.2"
1389   xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="urn:liberty:ac:1.2"
1390   xmlns:xs="http://www.w3.org/2001/XMLSchema" version="1.2-04">
1391   <!-- added to get the Extension element -->
1392   <xs:include schemaLocation="lib-arch-utility.xsd"/>
1393   <xs:annotation>
```

```
1394     <xs:documentation> ### IMPORTANT NOTICE ### The source code in this XSD
1395 file was excerpted verbatim from: Liberty Authentication Context Specification
1396 (liberty-architecture-authentication-context-v1.2) Version 1.2 04 April 2003
1397 The following notices pertain to this source file: Copyright (c) 2002,2003
1398 ActivCard; American Express Travel Related Services; America Online, Inc.; Bank
1399 of America; Bell Canada; Catavault; Cingular Wireless; Cisco Systems, Inc.;
1400 Citigroup; Communicator, Inc.; Consignia; Cyberun Corporation; Deloitte &
1401 Touche LLP; Earthlink, Inc.; Electronic Data Systems, Inc.; Entrust, Inc.;
1402 Ericsson; Fidelity Investments; France Telecom; Gemplus; General Motors; Hewlett-
1403 Packard Company; i2 Technologies, Inc.; Internet2; Intuit Inc.; MasterCard
1404 International; NEC Corporation; Netegrity; NeuStar; Nextel Communications; Nippon
1405 Telegraph and Telephone Company; Nokia Corporation; Novell, Inc.; NTT DoCoMo,
1406 Inc.; OneName Corporation; Openwave Systems Inc.; Phaos Technology;
1407 PricewaterhouseCoopers LLP; Register.com; RSA Security Inc; Sabre Holdings
1408 Corporation; SAP AG; SchlumbergerSema; SK Telecom; Sony Corporation; Sun
1409 Microsystems, Inc.; Trustgenix; United Airlines; VeriSign, Inc.; Visa
1410 International; Vodafone Group Plc; Wave Systems;. All rights reserved. This
1411 specification document has been prepared by Sponsors of the Liberty Alliance.
1412 Permission is hereby granted to use the document solely for the purpose of
1413 implementing the Specification. No rights are granted to prepare derivative works
1414 of this Specification. Entities seeking permission to reproduce portions of this
1415 document for other uses must contact the Liberty Alliance to determine whether an
1416 appropriate license for such use is available. Implementation of certain
1417 elements of this Specification may require licenses under third party
1418 intellectual property rights, including without limitation, patent rights. The
1419 Sponsors of and any other contributors to the Specification are not, and shall
1420 not be held responsible in any manner, for identifying or failing to identify any
1421 or all such third party intellectual property rights. This Specification is
1422 provided "AS IS", and no participant in the Liberty Alliance makes any warranty
1423 of any kind, express or implied, including any implied warranties of
1424 merchantability, non-infringement of third party intellectual property rights,
1425 and fitness for a particular purpose. Implementors of this Specification are
1426 advised to review the Liberty Alliance Project's website
1427 (http://www.projectliberty.org) for information concerning any Necessary Claims
1428 Disclosure Notices that have been received by the Liberty Alliance Management
1429 Board. Liberty Alliance Project Licensing Administrator c/o IEEE-ISTO 445
1430 Hoes Lane Piscataway, NJ 08855-1331, USA
1431     </xs:documentation>
1432     <xs:documentation>
1433 http://www.projectliberty.org/schemas/authctx/2002/05/</xs:documentation>
1434     </xs:annotation>
1435     <xs:element name="AuthenticationContextStatement">
1436         <xs:annotation>
1437             <xs:documentation>A particular assertion on an identity provider's part
1438 with respect to the authentication context associated with an authentication
1439 assertion. </xs:documentation>
1440         </xs:annotation>
1441         <xs:complexType>
1442             <xs:sequence>
1443                 <xs:element ref="Identification" minOccurs="0"/>
1444                 <xs:element ref="TechnicalProtection" minOccurs="0"/>
1445                 <xs:element ref="OperationalProtection" minOccurs="0"/>
1446                 <xs:element ref="AuthenticationMethod" minOccurs="0"/>
1447                 <xs:element ref="GoverningAgreements" minOccurs="0"/>
1448                 <xs:element ref="AuthenticatingIdP" minOccurs="0"/>
1449                 <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1450             </xs:sequence>
1451             <xs:attribute name="ID" type="xs:ID"/>
1452         </xs:complexType>
1453     </xs:element>
1454     <xs:element name="Identification">
1455         <xs:annotation>
1456             <xs:documentation>Refers to those characteristics that describe the
1457 processes and mechanisms the identity provider uses to initially create an
1458 association between a Principal and the identity
1459 (or name) by which the Principal will be known</xs:documentation>
1460         </xs:annotation>
```

```

1461     <xs:complexType>
1462         <xs:sequence>
1463             <xs:element ref="PhysicalVerification" minOccurs="0"/>
1464             <xs:element ref="WrittenConsent" minOccurs="0"/>
1465             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1466         </xs:sequence>
1467         <xs:attribute name="nym">
1468             <xs:annotation>
1469                 <xs:documentation>This attribute indicates whether or not the
1470 Identification mechanisms allow the
1471 actions of the Principal to be linked to an actual end
1472 user.</xs:documentation>
1473             </xs:annotation>
1474             <xs:simpleType>
1475                 <xs:restriction base="xs:NMTOKEN">
1476                     <xs:enumeration value="anonymity"/>
1477                     <xs:enumeration value="verinymity"/>
1478                     <xs:enumeration value="pseudonymity"/>
1479                 </xs:restriction>
1480             </xs:simpleType>
1481         </xs:attribute>
1482     </xs:complexType>
1483 </xs:element>
1484 <xs:element name="PhysicalVerification">
1485     <xs:annotation>
1486         <xs:documentation>This element indicates that identification has been
1487 performed in a physical face-to-face meeting with the principal and not in an
1488 online manner. </xs:documentation>
1489     </xs:annotation>
1490     <xs:complexType>
1491         <xs:attribute name="credentialLevel">
1492             <xs:simpleType>
1493                 <xs:restriction base="xs:NMTOKEN">
1494                     <xs:enumeration value="primary"/>
1495                     <xs:enumeration value="secondary"/>
1496                 </xs:restriction>
1497             </xs:simpleType>
1498         </xs:attribute>
1499     </xs:complexType>
1500 </xs:element>
1501 <xs:element name="WrittenConsent">
1502     <xs:complexType>
1503         <xs:sequence>
1504             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1505         </xs:sequence>
1506     </xs:complexType>
1507 </xs:element>
1508 <xs:element name="TechnicalProtection">
1509     <xs:annotation>
1510         <xs:documentation>Refers to those characteristics that describe how the
1511 'secret' (the knowledge or possession of which allows the Principal to
1512 authenticate to the identity provider) is kept secure</xs:documentation>
1513     </xs:annotation>
1514     <xs:complexType>
1515         <xs:sequence>
1516             <xs:element ref="PrivateKeyProtection" minOccurs="0"/>
1517             <xs:element ref="SharedKeyProtection" minOccurs="0"/>
1518             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1519         </xs:sequence>
1520     </xs:complexType>
1521 </xs:element>
1522 <xs:element name="SharedKeyProtection">
1523     <xs:annotation>
1524         <xs:documentation>This element indicates the types and strengths of
1525 facilities of a UA used to protect a shared secret key from unauthorized
1526 access and/or use.</xs:documentation>
1527     </xs:annotation>

```

```

1528     <xs:complexType>
1529         <xs:choice minOccurs="0">
1530             <xs:element ref="MobileDevice"/>
1531             <xs:element ref="MobileAuthCard"/>
1532         </xs:choice>
1533     </xs:complexType>
1534 </xs:element>
1535 <xs:element name="MobileDevice">
1536     <xs:annotation>
1537         <xs:documentation>This element indicates that the shared secret key is
1538 securely maintained in a mobile device
1539 (as opposed to being stored in a mobile authentication card).</xs:documentation>
1540     </xs:annotation>
1541     <xs:complexType>
1542         <xs:sequence>
1543             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1544         </xs:sequence>
1545     </xs:complexType>
1546 </xs:element>
1547 <xs:element name="MobileAuthCard">
1548     <xs:annotation>
1549         <xs:documentation>This element indicates that the shared secret key is
1550 securely maintained in a mobile authentication card (e.g., a SIM
1551 card).</xs:documentation>
1552     </xs:annotation>
1553     <xs:complexType>
1554         <xs:sequence>
1555             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1556         </xs:sequence>
1557     </xs:complexType>
1558 </xs:element>
1559 <xs:element name="PrivateKeyProtection">
1560     <xs:annotation>
1561         <xs:documentation>This element indicates the types and strengths of
1562 facilities of a UA used to protect a private key from unauthorized access
1563 and/or use.</xs:documentation>
1564     </xs:annotation>
1565     <xs:complexType>
1566         <xs:sequence>
1567             <xs:element ref="KeyActivation" minOccurs="0"/>
1568             <xs:element ref="KeyStorage" minOccurs="0"/>
1569             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1570         </xs:sequence>
1571     </xs:complexType>
1572 </xs:element>
1573 <xs:element name="KeyActivation">
1574     <xs:annotation>
1575         <xs:documentation>The actions that must be performed before the private
1576 key can be used. </xs:documentation>
1577     </xs:annotation>
1578     <xs:complexType>
1579         <xs:choice>
1580             <xs:element ref="Password"/>
1581         </xs:choice>
1582     </xs:complexType>
1583 </xs:element>
1584 <xs:element name="KeyStorage">
1585     <xs:annotation>
1586         <xs:documentation>In which medium is the private key stored.
1587 memory - the private key is stored in memory.
1588 ecard - the private key is stored in a smartcard.
1589 token - the private key is stored in a hardware token.
1590 MobileAuthCard - the private key is stored in a mobile authentication card
1591 (e.g., SIM card). </xs:documentation>
1592     </xs:annotation>
1593     <xs:complexType>
1594         <xs:attribute name="medium" use="required">

```

```

1595         <xs:simpleType>
1596             <xs:restriction base="xs:NMTOKEN">
1597                 <xs:enumeration value="memory"/>
1598                 <xs:enumeration value="smartcard"/>
1599                 <xs:enumeration value="token"/>
1600                 <xs:enumeration value="MobileAuthCard"/>
1601             </xs:restriction>
1602         </xs:simpleType>
1603     </xs:attribute>
1604 </xs:complexType>
1605 </xs:element>
1606 <xs:element name="Password">
1607     <xs:annotation>
1608         <xs:documentation>This element indicates that a password (or PIN or
1609 passphrase) has been used to authenticate the Principal or to gain access to some
1610 resource (for example, to gain access to the private key).</xs:documentation>
1611     </xs:annotation>
1612     <xs:complexType>
1613         <xs:sequence>
1614             <xs:element ref="Length" minOccurs="0"/>
1615             <xs:element ref="Generation" minOccurs="0"/>
1616             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1617         </xs:sequence>
1618     </xs:complexType>
1619 </xs:element>
1620 <xs:element name="Token">
1621     <xs:annotation>
1622         <xs:documentation>This element indicates that a hardware or software
1623 token is used as a method of identifying the Principal.</xs:documentation>
1624     </xs:annotation>
1625     <xs:complexType>
1626         <xs:sequence>
1627             <xs:element ref="TimeSyncToken"/>
1628             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1629         </xs:sequence>
1630     </xs:complexType>
1631 </xs:element>
1632 <xs:element name="TimeSyncToken">
1633     <xs:annotation>
1634         <xs:documentation>This element indicates that a time synchronization
1635 token is used to identify the Principal. hardware - the time synchronization token
1636 has been implemented in hardware. software - the time synchronization token has
1637 been implemented in software. SeedLength - the length, in bits, of the random
1638 seed used in the time synchronization token. </xs:documentation>
1639     </xs:annotation>
1640     <xs:complexType>
1641         <xs:attribute name="DeviceType" use="required">
1642             <xs:simpleType>
1643                 <xs:restriction base="xs:NMTOKEN">
1644                     <xs:enumeration value="hardware"/>
1645                     <xs:enumeration value="software"/>
1646                 </xs:restriction>
1647             </xs:simpleType>
1648         </xs:attribute>
1649         <xs:attribute name="SeedLength" type="xs:integer" use="required"/>
1650         <xs:attribute name="DeviceInHand" use="required">
1651             <xs:simpleType>
1652                 <xs:restriction base="xs:NMTOKEN">
1653                     <xs:enumeration value="true"/>
1654                     <xs:enumeration value="false"/>
1655                 </xs:restriction>
1656             </xs:simpleType>
1657         </xs:attribute>
1658     </xs:complexType>
1659 </xs:element>
1660 <xs:element name="Smartcard">
1661     <xs:annotation>

```



```
1662         <xs:documentation>This element indicates that a smartcard is used to
1663 identity the Principal.</xs:documentation>
1664     </xs:annotation>
1665     <xs:complexType>
1666         <xs:sequence>
1667             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
1668         </xs:sequence>
1669     </xs:complexType>
1670 </xs:element>
1671 <xs:element name="Length">
1672     <xs:annotation>
1673         <xs:documentation>This element indicates the minimum and/or maximum ASCII
1674 length of the password which is enforced (by the UA or the IdP). In other words,
1675 this is the minimum and/or maximum number of ASCII characters required to
1676 represent a valid password. min - the minimum number of ASCII characters
1677 required in a valid password, as enforced by the UA or the IdP.
1678 max - the maximum number of ASCII characters required in a valid password,
1679 as enforced by the UA or the IdP.</xs:documentation>
1680     </xs:annotation>
1681     <xs:complexType>
1682         <xs:attribute name="min" type="xs:integer" use="required" />
1683         <xs:attribute name="max" type="xs:integer" use="optional" />
1684     </xs:complexType>
1685 </xs:element>
1686 <xs:element name="Generation">
1687     <xs:annotation>
1688         <xs:documentation>Indicates whether the password was chosen by the
1689 Principal or auto-supplied by the identity provider.
1690 principalchosen - the Principal is allowed to choose the value of the
1691 password. This is true even if the initial password is chosen at
1692 random by the UA or the IdP and the Principal is then free to change the
1693 password. automatic - the password is chosen by the UA or the IdP to be
1694 cryptographically strong in some sense, or to satisfy certain password rules,
1695 and that the Principal is not free to change it or to choose a new password.
1696     </xs:documentation>
1697     </xs:annotation>
1698     <xs:complexType>
1699         <xs:attribute name="mechanism" use="required">
1700             <xs:simpleType>
1701                 <xs:restriction base="xs:NMTOKEN">
1702                     <xs:enumeration value="principalchosen" />
1703                     <xs:enumeration value="automatic" />
1704                 </xs:restriction>
1705             </xs:simpleType>
1706         </xs:attribute>
1707     </xs:complexType>
1708 </xs:element>
1709 <xs:element name="AuthenticationMethod">
1710     <xs:annotation>
1711         <xs:documentation>Refers to those characteristics that define the
1712 mechanisms by which the Principal authenticates to the identity
1713 provider.</xs:documentation>
1714     </xs:annotation>
1715     <xs:complexType>
1716         <xs:sequence>
1717             <xs:element ref="PrincipalAuthenticationMechanism" />
1718             <xs:element ref="Authenticator" minOccurs="0" />
1719             <xs:element ref="AuthenticatorTransportProtocol" />
1720             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
1721         </xs:sequence>
1722     </xs:complexType>
1723 </xs:element>
1724 <xs:element name="PrincipalAuthenticationMechanism">
1725     <xs:annotation>
1726         <xs:documentation>The method that a Principal employs to perform
1727 authentication to local system components.</xs:documentation>
1728     </xs:annotation>
```

```
1729     <xs:complexType>
1730         <xs:choice minOccurs="0" maxOccurs="unbounded">
1731             <xs:element ref="Password"/>
1732             <xs:element ref="Token"/>
1733             <xs:element ref="Smartcard"/>
1734         </xs:choice>
1735     </xs:complexType>
1736 </xs:element>
1737 <xs:element name="Authenticator">
1738     <xs:annotation>
1739         <xs:documentation>The method applied to validate a principal's
1740 authentication across a network </xs:documentation>
1741     </xs:annotation>
1742     <xs:complexType>
1743         <xs:choice minOccurs="0" maxOccurs="unbounded">
1744             <xs:element ref="PreviousSession"/>
1745             <xs:element ref="Dig-sig"/>
1746             <xs:element ref="ZeroKnowledge"/>
1747             <xs:element ref="SharedSecretChallengeResponse"/>
1748         </xs:choice>
1749     </xs:complexType>
1750 </xs:element>
1751 <xs:element name="PreviousSession">
1752     <xs:annotation>
1753         <xs:documentation>Indicates that the Principal has been strongly
1754 authenticated in a previous session during which the IdP has set a cookie in the
1755 UA. During the present session the Principal has only been authenticated by the
1756 UA returning the cookie to the IdP.</xs:documentation>
1757     </xs:annotation>
1758     <xs:complexType>
1759         <xs:sequence>
1760             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1761         </xs:sequence>
1762     </xs:complexType>
1763 </xs:element>
1764 <xs:element name="ZeroKnowledge">
1765     <xs:annotation>
1766         <xs:documentation>This element indicates that the Principal has been
1767 authenticated by a zero knowledge technique as specified in ISO/IEC 9798-
1768 5.</xs:documentation>
1769     </xs:annotation>
1770     <xs:complexType>
1771         <xs:sequence>
1772             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1773         </xs:sequence>
1774     </xs:complexType>
1775 </xs:element>
1776 <xs:element name="SharedSecretChallengeResponse">
1777     <xs:annotation>
1778         <xs:documentation>This element indicates that the Principal has been
1779 authenticated by a challenge-response protocol utilizing shared secret keys and
1780 symmetric cryptography.</xs:documentation>
1781     </xs:annotation>
1782     <xs:complexType>
1783         <xs:sequence>
1784             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1785         </xs:sequence>
1786     </xs:complexType>
1787 </xs:element>
1788 <xs:element name="Dig-sig">
1789     <xs:annotation>
1790         <xs:documentation>This element indicates that the Principal has been
1791 authenticated by a mechanism which involves the Principal
1792 computing a digital signature over at least challenge data provided by the
1793 IdP.</xs:documentation>
1794     </xs:annotation>
1795     <xs:complexType>
```

```
1796         <xs:sequence>
1797             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
1798         </xs:sequence>
1799     </xs:complexType>
1800 </xs:element>
1801 <xs:element name="AuthenticatorTransportProtocol">
1802     <xs:annotation>
1803         <xs:documentation>The protocol across which Authenticator information is
1804 transferred to an identity provider verifier.</xs:documentation>
1805     </xs:annotation>
1806     <xs:complexType>
1807         <xs:choice minOccurs="0" maxOccurs="unbounded">
1808             <xs:element ref="HTTP" />
1809             <xs:element ref="SSL" />
1810             <xs:element ref="MobileNetwork" />
1811             <xs:element ref="WTLS" />
1812             <xs:element ref="IPSec" />
1813         </xs:choice>
1814     </xs:complexType>
1815 </xs:element>
1816 <xs:element name="HTTP">
1817     <xs:annotation>
1818         <xs:documentation>This element indicates that the Authenticator has been
1819 transmitted using bare HTTP utilizing no additional security
1820 protocols.</xs:documentation>
1821     </xs:annotation>
1822     <xs:complexType>
1823         <xs:sequence>
1824             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
1825         </xs:sequence>
1826     </xs:complexType>
1827 </xs:element>
1828 <xs:element name="IPSec">
1829     <xs:annotation>
1830         <xs:documentation>This element indicates that the Authenticator has been
1831 transmitted using a transport mechanism protected by an IPSEC
1832 session.</xs:documentation>
1833     </xs:annotation>
1834     <xs:complexType>
1835         <xs:sequence>
1836             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
1837         </xs:sequence>
1838     </xs:complexType>
1839 </xs:element>
1840 <xs:element name="WTLS">
1841     <xs:annotation>
1842         <xs:documentation>This element indicates that the Authenticator has been
1843 transmitted using a transport mechanism protected by a WTLS
1844 session.</xs:documentation>
1845     </xs:annotation>
1846     <xs:complexType>
1847         <xs:sequence>
1848             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
1849         </xs:sequence>
1850     </xs:complexType>
1851 </xs:element>
1852 <xs:element name="MobileNetwork">
1853     <xs:annotation>
1854         <xs:documentation>This element indicates that the Authenticator has been
1855 transmitted solely across a mobile network using no additional security
1856 mechanism.</xs:documentation>
1857     </xs:annotation>
1858     <xs:complexType>
1859         <xs:sequence>
1860             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
1861         </xs:sequence>
1862     </xs:complexType>
```

```
1863 </xs:element>
1864 <xs:element name="SSL">
1865   <xs:annotation>
1866     <xs:documentation>This element indicates that the Authenticator has been
1867     transmitted using a transport mechanism protected by an SSL or TLS
1868     session.</xs:documentation>
1869   </xs:annotation>
1870   <xs:complexType>
1871     <xs:sequence>
1872       <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1873     </xs:sequence>
1874   </xs:complexType>
1875 </xs:element>
1876 <xs:element name="OperationalProtection">
1877   <xs:annotation>
1878     <xs:documentation>Refers to those characteristics that describe
1879     procedural security controls employed by the identity
1880     provider.</xs:documentation>
1881   </xs:annotation>
1882   <xs:complexType>
1883     <xs:sequence>
1884       <xs:element ref="SecurityAudit" minOccurs="0"/>
1885       <xs:element ref="DeactivationCallCenter" minOccurs="0"/>
1886       <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1887     </xs:sequence>
1888   </xs:complexType>
1889 </xs:element>
1890 <xs:element name="SecurityAudit">
1891   <xs:complexType>
1892     <xs:sequence>
1893       <xs:element ref="SwitchAudit" minOccurs="0"/>
1894       <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1895     </xs:sequence>
1896   </xs:complexType>
1897 </xs:element>
1898 <xs:element name="SwitchAudit">
1899   <xs:complexType>
1900     <xs:sequence>
1901       <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1902     </xs:sequence>
1903   </xs:complexType>
1904 </xs:element>
1905 <xs:element name="DeactivationCallCenter">
1906   <xs:complexType>
1907     <xs:sequence>
1908       <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1909     </xs:sequence>
1910   </xs:complexType>
1911 </xs:element>
1912 <xs:element name="GoverningAgreements">
1913   <xs:annotation>
1914     <xs:documentation>Provides a mechanism for linking to external (likely
1915     human readable) documents in which the identity provider can define business
1916     level authentication context, e.g. liability constraints, contractual
1917     obligations.</xs:documentation>
1918   </xs:annotation>
1919   <xs:complexType>
1920     <xs:sequence>
1921       <xs:element ref="GoverningAgreementRef"/>
1922     </xs:sequence>
1923   </xs:complexType>
1924 </xs:element>
1925 <xs:element name="GoverningAgreementRef">
1926   <xs:complexType>
1927     <xs:attribute name="governingAgreementRef" type="xs:anyURI"
1928     use="required"/>
1929   </xs:complexType>
```

```
1930 </xs:element>
1931 <xs:element name="AuthenticatingIdP">
1932   <xs:annotation>
1933     <xs:documentation>The IdP that originally authenticated the Principal.
1934   </xs:documentation>
1935   </xs:annotation>
1936   <xs:complexType>
1937     <xs:sequence>
1938       <xs:element ref="GoverningAgreements" minOccurs="0" />
1939     </xs:sequence>
1940     <xs:attribute name="ID" type="xs:anyURI" use="required" />
1941   </xs:complexType>
1942 </xs:element>
1943 </xs:schema>
```

1944 6 References

- 1945 [LibertyGloss] Mauldin, H., & Wason, T., eds. (January 2003). "Liberty Architecture Glossary,"
1946 Version 1.1. Liberty Alliance Project, <<http://www.projectliberty.org/specs/>>.
- 1947 [LibertyProtSchema] Beatty, J., & Kemp, J., eds. (January 2003). "Liberty Protocols and Schema
1948 Specification," Version 1.1. Liberty Alliance Project,
1949 <<http://www.projectliberty.org/specs/>>.
- 1950 [PDS] Santesson, S. & Baum, M., (May 2000). "Internet X.509 Public Key Infrastructure PKI
1951 Disclosure Statement," Internet Draft . The Internet Engineering Task Force,
1952 <<http://www.verisign.com/repository/pds.txt>> [20 December 2002].
- 1953 [RFC2119] Bradner, S. (March 1997). "Key words for use in RFCs to Indicate Requirement
1954 Levels," RFC 2119. The Internet Engineering Task Force, <[http://www.rfc-
1955 editor.org/rfc/rfc2119.txt](http://www.rfc-editor.org/rfc/rfc2119.txt)> [18 December 2002].
- 1956 [RFC2527] Chokhani, S., Ford, W. (March 1999). "Internet X.509 Public Key Infrastructure
1957 Certificate Policy and Certification Practices Framework," RFC 2527. The Internet
1958 Engineering Task Force, <<http://www.ietf.org/rfc/rfc2527.txt?number=2527>> [20
1959 December 2002].
- 1960 [SAMLBind] Mishra, P., ed. (05 Nov. 2002). "Bindings and Profiles for the OASIS Security
1961 Assertion Markup Language (SAML)," Version 1.0, OASIS Standard. Organization for
1962 the Advancement of Structured Information Standards, <[http://www.oasis-
1963 open.org/committees/security/#documents](http://www.oasis-open.org/committees/security/#documents)> [18 December 2002].
- 1964 [SAMLCore] Hallam-Baker, P., Maler, E., eds. (05 Nov. 2002). "Assertions and Protocol for the
1965 OASIS Security Assertion Markup Language (SAML)," Version 1.0, OASIS Standard.
1966 Organization for the Advancement of Structured Information Standards,
1967 <<http://www.oasis-open.org/committees/security/#documents>> [18 December 2002].
- 1968 [Schema1] Thompson, H. S., Beech, D., Maloney, M., & Mendleson, N., eds. (May 2002). "XML
1969 Schema Part 1: Structures," Recommendation. World Wide Web Consortium,
1970 <<http://www.w3.org/TR/xmlschema-1/>> [18 December 2002].
- 1971