

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39

# Liberty Authentication Context Specification

Version 1.0

11 July 2002

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

39 **Notice**

40

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

51

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

57

58 Implementation of this Specification may involve the use of one or more of the following United  
59 States Patents claimed by AOL Time Warner, Inc.: No.5,774,670, No.6,134,592, No.5,826,242, No.  
60 5,825,890, and No.5,671,279. The Sponsors of the Specification take no position concerning the  
61 evidence, validity or scope of the claimed subject matter of the aforementioned patents.

62 Implementation of certain elements of this Specification may also require licenses under third party  
63 intellectual property rights other than those identified above, including without limitation, patent  
64 rights. The Sponsors of the Specification are not and shall not be held responsible in any manner for  
65 identifying or failing to identify any or all such intellectual property rights that may be involved in  
66 the implementation of the Specification.

67

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

72

73 Liberty Alliance Project  
74 Licensing Administrator  
75 c/o IEEE-ISTO  
76 445 Hoes Lane, P.O. Box 1331  
77 Piscataway, NJ 08855-1331, USA

78

78 **Editor**

79 Paul Madsen, Entrust, Inc.

80 **Contributors**

81  
82 The following Liberty Alliance Project Sponsor companies contributed to the development of  
83 this specification:  
84

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

85

86

86 **Table of Contents**

87	1	Introduction .....	5
88	1.1	Notation.....	5
89	2	Overview .....	6
90	3	Authentication Context.....	6
91	3.1	Authentication Context Classes .....	7
92	3.2	Authentication Quality.....	9
93	3.2.1	Service Provider Request .....	9
94	3.2.2	Identity Provider Response .....	9
95	4	Previous work.....	10
96	4.1	PKI.....	10
97	4.2	SAML .....	10
98	5	Liberty Authentication Context Mechanisms.....	11
99	5.1	Authentication Context Classes .....	11
100	5.1.1	MobileContract.....	11
101	5.1.2	MobileDigitalID.....	14
102	5.1.3	MobileUnregistered.....	16
103	5.1.4	Password.....	18
104	5.1.5	Password- ProtectedTransport.....	19
105	5.1.6	Previous-Session .....	20
106	5.1.7	Smartcard .....	21
107	5.1.8	Smartcard-PKI.....	22
108	5.1.9	Software-PKI.....	24
109	5.1.10	Time-Sync-Token .....	25
110	5.2	Authentication Context Schema .....	27
111	5.2.1	XML Schema .....	27
112	6	References .....	35
113			
114			

## 114 1 Introduction

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

### 117 1.1 Notation

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

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

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

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

132  
133 Listings of XML schemas appear like this.

134  
135 Example code listings appear like this.

136

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

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

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

## 153 2 Overview

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

170  
171 *Authentication context* is defined as the information additional to the authentication assertion itself  
172 that the service provider may require before it makes an entitlements decision.

## 173 3 Authentication Context

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

- 177
- 178 • Initial user identification mechanisms (for example, face-to-face, online, shared secret)
- 179 • Mechanisms for minimizing compromise of a Principal's credentials (for example, credential  
180 renewal frequency, client-side key generation)
- 181 • Mechanisms for storing and protecting credentials (for example, smartcard, password rules)
- 182 • Authentication mechanism (for example, password, certificate-based SSL)

183  
184 The variations and permutations in the examples above guarantee that not all authentication  
185 assertions are the same; a particular authentication assertion will be characterized by the values for  
186 each of these variables. A somewhat helpful model is to think of an authentication assertion as  
187 defined by its coordinates in a multidimensional space. This model is demonstrated in Figure 1  
188 (where only three axes are shown).  
189

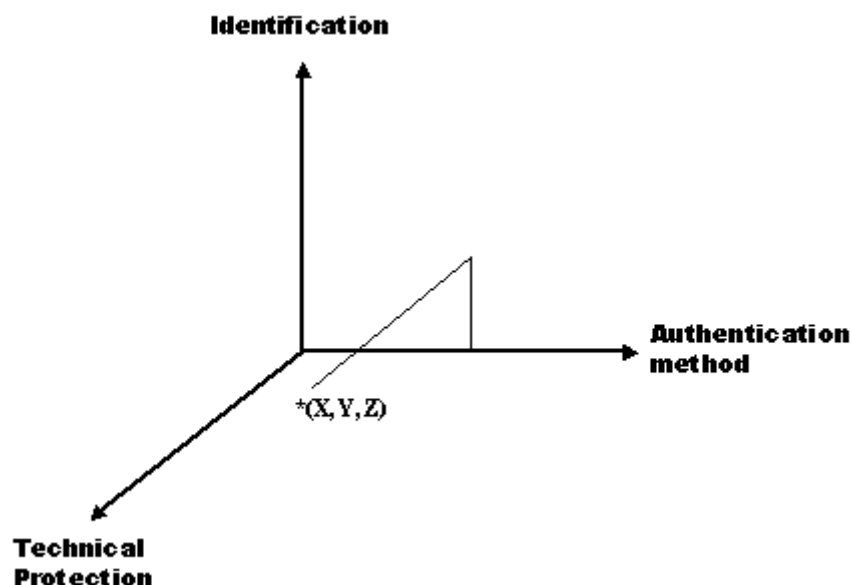


Figure 1: Authentication assertion as defined by its coordinates in multidimensional space

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

### 3.1 Authentication Context Classes

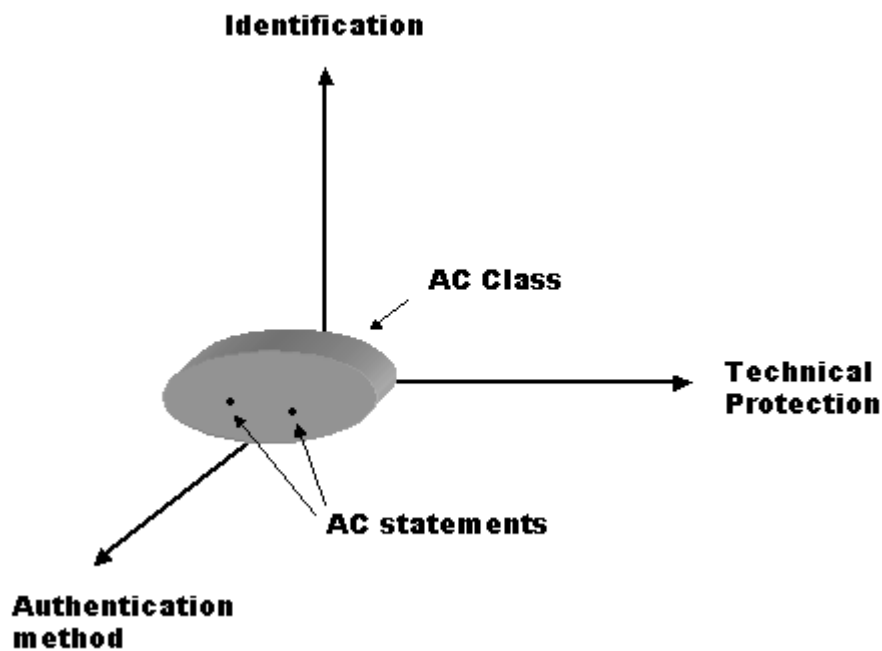
Liberty can simplify for service providers the task of assessing and comparing authentication assertions by defining particular authentication contexts that are representative of current technologies and practices among identity providers. For instance, a typical authentication context will be when a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

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

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

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

225 Rather than a class being a rigid collection of these characteristics, a class will define a set of  
226 conformant authentication context statements (for example, multiple and different authentication  
227 context statements will satisfy the requirements of a given class). The relationship between an  
228 authentication context class and particular authentication context statements is shown in Figure 2,  
229 where all the authentication context statements satisfy the requirements expressed by the class.  
230



231  
232 **Figure 2: Relationship between authentication context class and statements**

233  
234 By introducing the additional layer of classes and by defining an initial list of representative and  
235 flexible classes, Liberty architecture

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



## 246 **3.2 Authentication Quality**

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

### 254 **3.2.1 Service Provider Request**

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

- 259 1. A match on a particular authentication context statement
- 260 2. A match within a specific authentication context class
- 261 3. A match or better on a particular authentication context class
- 262 4. A match within an ordered list (which is designated by the service provider) of authentication  
263 context classes
- 264

265  
266 Option 1 will require that the identity provider and service provider have previously agreed on the  
267 details of a particular authentication context that either does not fall into one of the Liberty-defined  
268 authentication context classes or needs to be constrained more tightly.

269  
270 Option 2 is expected to be the typical scenario.

271  
272 For option 3, the decision as to what is better is left to the entity best qualified to make that  
273 determination, the identity provider. The service provider, trusting the identity provider's judgment,  
274 will accept the assertion it receives back because it will be confident the assertion meets (or  
275 exceeds) the provider's requirements.

276  
277 Option 4 will give the service provider greater control over the authentication context classes to  
278 which the authentication assertions it receives conform. The identity provider is given no leeway in  
279 providing an authentication assertion conforming to a class not on the list.

280  
281 If the service provider does not specify any of the above options in the <lib:AuthnAndFedRequest>,  
282 the identity provider will be free to provide an authentication context of its choosing.

### 283 **3.2.2 Identity Provider Response**

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

287  
288 The returned authentication assertion will include a URI specifying the associated authentication  
289 context statement.

## 290 4 Previous work

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

### 292 4.1 PKI

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

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

### 324 4.2 SAML

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

329

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

335  
336  
337  
338  
339  
340  
341  
342  
343

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

344  
345  
346  
347  
348  
349

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

350  
351

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

352  
353  
354  
355

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

356  
357  
358  
359  
360

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

361  
362  
363

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

## 364 5 Liberty Authentication Context Mechanisms

### 365 5.1 Authentication Context Classes

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

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

370 Classes are identified by URIs with the initial stem:

371  
372  
373

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

#### 374 5.1.1 MobileContract

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

##### 377 5.1.1.1 Associated Liberty URI

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

379 **5.1.1.2 Class Schema**

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
<annotation>
<documentation> http://www.projectliberty.org/schemas/authctx/classes/MobileContract
</documentation>
</annotation>
  <xs:element name="AuthenticationContextStatement">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" ref="Identification"/>
        <xs:element minOccurs="1" maxOccurs="1"
ref="TechnicalProtection"/>
        <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticationMethod"/>
        <xs:element minOccurs="1" maxOccurs="1"
ref="OperationalProtection"/>
        <xs:element minOccurs="1" maxOccurs="1"
ref="GoverningAgreements"/>
        <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="AuthenticationMethod">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="1" maxOccurs="1" ref="Authenticator"/>
          <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticatorTransportProtocol"/>
          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Authenticator">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1"
ref="SharedSecretChallengeResponse"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="AuthenticatorTransportProtocol">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="1" maxOccurs="1" ref="MobileNetwork"/>
              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="DeactivationCallCenter">
            <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
          </xs:element>
          <xs:element name="GoverningAgreementRef">
            <xs:complexType>
              <xs:attribute name="ref"
fixed="http://SomeMobileforum.org/namespaces/authcontext/classes/Mobile-Class2.pdf"/>
            </xs:complexType>
          </xs:element>

```

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

```
514     <xs:element name="SwitchAudit">
515         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
516 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
517     </xs:element>
518     <xs:element name="TechnicalProtection">
519         <xs:complexType>
520             <xs:sequence>
521                 <xs:element minOccurs="1" maxOccurs="1"
522 ref="SharedKeyProtection"/>
523                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
524 processContents="lax" /></xs:sequence>
525             </xs:complexType>
526         </xs:element>
527 </xs:schema>
```

## 528 5.1.2 MobileDigitalID

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

### 531 5.1.2.1 Associated Liberty URI

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

### 533 5.1.2.2 Class Schema

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

```

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

```

```

641     </xs:element>
642     <xs:element name="PrivateKeyProtection">
643         <xs:complexType>
644             <xs:sequence>
645                 <xs:element ref="KeyStorage"/>
646                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
647             </xs:complexType>
648         </xs:element>
649     <xs:element name="SSL">
650         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
651 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
652     </xs:element>
653     <xs:element name="SecurityAudit">
654         <xs:complexType>
655             <xs:sequence>
656                 <xs:element ref="SwitchAudit"/>
657                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
658             </xs:complexType>
659     </xs:element>
660     <xs:element name="SwitchAudit">
661         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
662 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
663     </xs:element>
664     <xs:element name="TechnicalProtection">
665         <xs:complexType>
666             <xs:sequence>
667                 <xs:element ref="PrivateKeyProtection"/>
668                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
669             </xs:complexType>
670     </xs:element>
671     <xs:element name="WTLS">
672         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
673 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
674     </xs:element>
675     <xs:element name="WrittenConsent">
676         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
677 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
678     </xs:element>
679     <xs:element name="ZeroKnowledge">
680         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
681 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
682     </xs:element>
683 </xs:schema>
684
685
686

```

### 687 5.1.3 MobileUnregistered

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

#### 690 5.1.3.1 Associated Liberty URI

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

#### 692 5.1.3.2 Class Schema

```

693 <?xml version="1.0" encoding="UTF-8"?>
694 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
695 <annotation>

```



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

```

766     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
767 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
768   </xs:element>
769   <xs:element name="OperationalProtection">
770     <xs:complexType>
771       <xs:sequence>
772         <xs:element ref="SecurityAudit"/>
773         <xs:element ref="DeactivationCallCenter"/>
774         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
775 processContents="lax" /></xs:sequence>
776       </xs:complexType>
777     </xs:element>
778     <xs:element name="SecurityAudit">
779       <xs:complexType>
780         <xs:sequence>
781           <xs:element ref="SwitchAudit"/>
782           <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
783 processContents="lax" /></xs:sequence>
784         </xs:complexType>
785       </xs:element>
786       <xs:element name="SharedKeyProtection">
787         <xs:complexType>
788           <xs:choice>
789             <xs:element ref="MobileAuthCard"/>
790             <xs:element ref="MobileDevice"/>
791           </xs:choice>
792         </xs:complexType>
793       </xs:element>
794       <xs:element name="SharedSecretChallengeResponse">
795         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
796 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
797       </xs:element>
798       <xs:element name="SwitchAudit">
799         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
800 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
801       </xs:element>
802       <xs:element name="TechnicalProtection">
803         <xs:complexType>
804           <xs:sequence>
805             <xs:element ref="SharedKeyProtection"/>
806             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
807 processContents="lax" /></xs:sequence>
808           </xs:complexType>
809         </xs:element>
810       </xs:schema>
811

```

## 812 5.1.4 Password

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

### 815 5.1.4.1 Associated Liberty URI

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

### 817 5.1.4.2 Class Schema

```

818 <?xml version="1.0" encoding="UTF-8"?>
819
820 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
821
822 <annotation>
823 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Password
824 </documentation>

```

```

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

```

## 877 5.1.5 Password- ProtectedTransport

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

### 880 5.1.5.1 Associated Liberty URI

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

### 882 5.1.5.2 Class Schema

```

883 <?xml version="1.0" encoding="UTF-8"?>
884

```

```

885
886 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
887
888 <annotation>
889 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Password-
890 ProtectedTransport </documentation>
891 </annotation>
892
893     <xs:element name="AuthenticationContextStatement">
894         <xs:complexType>
895             <xs:sequence>
896                 <xs:element ref="AuthenticationMethod"/>
897                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
899             </xs:complexType>
900         </xs:element>
901     <xs:element name="AuthenticationMethod">
902         <xs:complexType>
903             <xs:all>
904                 <xs:element ref="PrincipalAuthenticationMechanism"/>
905                 <xs:element ref="AuthenticatorTransportProtocol"/>
906             </xs:all>
907         </xs:complexType>
908     </xs:element>
909     <xs:element name="AuthenticatorTransportProtocol">
910         <xs:complexType>
911             <xs:sequence>
912                 <xs:element ref="SSL"/>
913                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
915             </xs:complexType>
916         </xs:element>
917         <xs:element name="Length">
918             <xs:complexType>
919                 <xs:attribute name="min" fixed="3"/>
920             </xs:complexType>
921         </xs:element>
922         <xs:element name="Password">
923             <xs:complexType>
924                 <xs:sequence>
925                     <xs:element ref="Length"/>
926                     <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
928                 </xs:complexType>
929             </xs:element>
930             <xs:element name="PrincipalAuthenticationMechanism">
931                 <xs:complexType>
932                     <xs:sequence>
933                         <xs:element ref="Password"/>
934                         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
936                     </xs:complexType>
937                 </xs:element>
938                 <xs:element name="SSL">
939                     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
941                 </xs:element>
942 </xs:schema>

```

### 943 5.1.6 Previous-Session

944 The Previous-Session class is identified when a Principal had authenticated to an identity provider at  
945 some point in the past using any authentication context supported by that identity provider.

946 Consequently, a subsequent authentication event that the identity provider will assert to the service  
947 provider may be significantly separated in time from the Principal's current resource access request.

948

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

### 953 5.1.6.1 Associated Liberty URI

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

### 955 5.1.6.2 Class Schema

```
956 <?xml version="1.0" encoding="UTF-8"?>
957 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
958
959 <annotation>
960 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Previous-Session
961 </documentation>
962 </annotation>
963
964 <xs:element name="AuthenticationContextStatement">
965 <xs:complexType>
966 <xs:sequence>
967 <xs:element minOccurs="1" maxOccurs="1"
968 ref="AuthenticationMethod"/>
969 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
970 processContents="lax" /></xs:sequence>
971 </xs:complexType>
972 </xs:element>
973
974 <xs:element name="AuthenticationMethod">
975 <xs:complexType>
976 <xs:sequence>
977 <xs:element ref="Authenticator" minOccurs="0" maxOccurs="1"/>
978 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
979 processContents="lax" /></xs:sequence>
980 </xs:complexType>
981 </xs:element>
982
983 <xs:element name="Authenticator">
984 <xs:complexType>
985 <xs:sequence>
986 <xs:element minOccurs="1" maxOccurs="1" ref="PreviousSession"/>
987 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
988 processContents="lax" /></xs:sequence>
989 </xs:complexType>
990 </xs:element>
991
992 <xs:element name="PreviousSession">
993 <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
994 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
995 </xs:element>
996 </xs:schema>
```

1000

### 1001 5.1.7 Smartcard

1002 The Smartcard class is identified when a Principal authenticates to an identity provider using a  
1003 smartcard.

1004 **5.1.7.1 Associated Liberty URI**

1005 <http://www.projectliberty.org/schemas/authctx/classes/Smartcard>

1006 **5.1.7.2 Class Schema**

```
1007 <?xml version="1.0" encoding="UTF-8"?>
1008
1009 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
1010
1011 <annotation>
1012 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Smartcard
1013 </documentation>
1014 </annotation>
1015
1016 <xs:element name="AuthenticationContextStatement">
1017 <xs:complexType>
1018 <xs:sequence>
1019 <xs:element minOccurs="1" maxOccurs="1"
1020 ref="AuthenticationMethod"/>
1021 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1022 processContents="lax" /></xs:sequence>
1023 </xs:complexType>
1024 </xs:element>
1025 <xs:element name="AuthenticationMethod">
1026 <xs:complexType>
1027 <xs:sequence>
1028 <xs:element minOccurs="1" maxOccurs="1"
1029 ref="PrincipalAuthenticationMechanism"/>
1030 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1031 processContents="lax" /></xs:sequence>
1032 </xs:complexType>
1033 </xs:element>
1034 <xs:element name="PrincipalAuthenticationMechanism">
1035 <xs:complexType>
1036 <xs:sequence>
1037 <xs:element minOccurs="1" maxOccurs="1" ref="Smartcard"/>
1038 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1039 processContents="lax" /></xs:sequence>
1040 </xs:complexType>
1041 </xs:element>
1042 <xs:element name="Smartcard">
1043 <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1044 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1045 </xs:element>
1046 </xs:schema>
1047
```

1048 **5.1.8 Smartcard-PKI**

1049 The Smartcard-PKI class is identified when a Principal authenticates to an identity provider through  
1050 a two-factor authentication mechanism using a smartcard with enclosed private key and a PIN.

1051 **5.1.8.1 Associated Liberty URI**

1052 <http://www.projectliberty.org/schemas/authctx/classes/Smartcard-PKI>

1053 **5.1.8.2 Class Schema**

```
1054 <?xml version="1.0" encoding="UTF-8"?>
1055
1056 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
1057 <annotation>
```

```
1059 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Smartcard-
1060 PKI</documentation>
1061 </annotation>
1062
1063     <xs:element name="AuthenticationContextStatement">
1064         <xs:complexType>
1065             <xs:sequence>
1066                 <xs:element minOccurs="1" maxOccurs="1"
1067 ref="TechnicalProtection"/>
1068                 <xs:element minOccurs="1" maxOccurs="1"
1069 ref="AuthenticationMethod"/>
1070                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1071 processContents="lax" /></xs:sequence>
1072             </xs:complexType>
1073         </xs:element>
1074     <xs:element name="AuthenticationMethod">
1075         <xs:complexType>
1076             <xs:sequence>
1077                 <xs:element minOccurs="1" maxOccurs="1"
1078 ref="PrincipalAuthenticationMechanism"/>
1079                 <xs:element minOccurs="1" maxOccurs="1" ref="Authenticator"/>
1080                 <xs:element minOccurs="1" maxOccurs="1"
1081 ref="AuthenticatorTransportProtocol"/>
1082                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1083 processContents="lax" /></xs:sequence>
1084             </xs:complexType>
1085         </xs:element>
1086     <xs:element name="Authenticator">
1087         <xs:complexType>
1088             <xs:sequence>
1089                 <xs:element minOccurs="1" maxOccurs="1" ref="Dig-sig"/>
1090                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1091 processContents="lax" /></xs:sequence>
1092             </xs:complexType>
1093         </xs:element>
1094     <xs:element name="AuthenticatorTransportProtocol">
1095         <xs:complexType>
1096             <xs:sequence>
1097                 <xs:element minOccurs="1" maxOccurs="1" ref="SSL"/>
1098                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1099 processContents="lax" /></xs:sequence>
1100             </xs:complexType>
1101         </xs:element>
1102     <xs:element name="Dig-sig">
1103         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1104 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1105     </xs:element>
1106     <xs:element name="KeyActivation">
1107         <xs:complexType>
1108             <xs:sequence>
1109                 <xs:element minOccurs="1" maxOccurs="1" ref="Password"/>
1110                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1111 processContents="lax" /></xs:sequence>
1112             </xs:complexType>
1113         </xs:element>
1114     <xs:element name="Length">
1115         <xs:complexType>
1116             <xs:attribute name="min" type="xs:byte" use="required"/>
1117         </xs:complexType>
1118     </xs:element>
1119     <xs:element name="Password">
1120         <xs:complexType>
1121             <xs:sequence>
1122                 <xs:element minOccurs="1" maxOccurs="1" ref="Length"/>
1123                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1124 processContents="lax" /></xs:sequence>
1125             </xs:complexType>
1126         </xs:element>
```

```
1127     <xs:element name="PrincipalAuthenticationMechanism">
1128         <xs:complexType>
1129             <xs:sequence>
1130                 <xs:element minOccurs="1" maxOccurs="1" ref="Smartcard"/>
1131                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
1132             </xs:complexType>
1133         </xs:element>
1134     <xs:element name="PrivateKeyProtection">
1135         <xs:complexType>
1136             <xs:sequence>
1137                 <xs:element minOccurs="1" maxOccurs="1" ref="KeyActivation"/>
1138                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
1139             </xs:complexType>
1140         </xs:element>
1141     <xs:element name="SSL">
1142         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1143     </xs:element>
1144     <xs:element name="Smartcard">
1145         <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1146     </xs:element>
1147     <xs:element name="TechnicalProtection">
1148         <xs:complexType>
1149             <xs:sequence>
1150                 <xs:element minOccurs="1" maxOccurs="1"
ref="PrivateKeyProtection"/>
1151                 <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
1152             </xs:complexType>
1153         </xs:element>
1154     </xs:schema>
```

## 1162 5.1.9 Software-PKI

1163 The Software-PKI class is identified when a Principal uses an X.509 certificate stored in software to  
1164 authenticate to the identity provider over an SSL protected session.

### 1166 5.1.9.1 Associated Liberty URI

1167  
1168 <http://www.projectliberty.org/schemas/authctx/classes/Software-PKI>

### 1169 5.1.9.2 Class Schema

```
1170 <?xml version="1.0" encoding="UTF-8"?>
1171 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
1172 <annotation>
1173 <documentation> http://www.projectliberty.org/schemas/authctx/classes/Software-PKI
1174 </documentation>
1175 </annotation>
1176 <xs:element name="AuthenticationContextStatement">
1177     <xs:complexType>
1178         <xs:sequence>
1179             <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticationMethod"/>
```



```

1185         <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1186 processContents="lax" /></xs:sequence>
1187     </xs:complexType>
1188 </xs:element>
1189 <xs:element name="AuthenticationMethod">
1190     <xs:complexType>
1191         <xs:sequence>
1192             <xs:element minOccurs="1" maxOccurs="1"
1193 ref="PrincipalAuthenticationMechanism" />
1194             <xs:element minOccurs="1" maxOccurs="1" ref="Authenticator" />
1195             <xs:element minOccurs="1" maxOccurs="1"
1196 ref="AuthenticatorTransportProtocol" />
1197             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1198 processContents="lax" /></xs:sequence>
1199         </xs:complexType>
1200     </xs:element>
1201 <xs:element name="Authenticator">
1202     <xs:complexType>
1203         <xs:sequence>
1204             <xs:element minOccurs="1" maxOccurs="1" ref="Dig-sig" />
1205             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1206 processContents="lax" /></xs:sequence>
1207         </xs:complexType>
1208     </xs:element>
1209 <xs:element name="AuthenticatorTransportProtocol">
1210     <xs:complexType>
1211         <xs:sequence>
1212             <xs:element minOccurs="1" maxOccurs="1" ref="SSL" />
1213             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1214 processContents="lax" /></xs:sequence>
1215         </xs:complexType>
1216     </xs:element>
1217 <xs:element name="Dig-sig">
1218     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1219 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1220 </xs:element>
1221 <xs:element name="Password">
1222     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1223 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1224 </xs:element>
1225 <xs:element name="PrincipalAuthenticationMechanism">
1226     <xs:complexType>
1227         <xs:sequence>
1228             <xs:element minOccurs="1" maxOccurs="1" ref="Password" />
1229             <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1230 processContents="lax" /></xs:sequence>
1231         </xs:complexType>
1232     </xs:element>
1233 <xs:element name="SSL">
1234     <xs:complexType><xs:sequence><xs:any namespace="##any" minOccurs="0"
1235 maxOccurs="unbounded" processContents="lax" /></xs:sequence></xs:complexType>
1236 </xs:element>
1237 </xs:schema>
1238

```

### 1239 5.1.10 Time-Sync-Token

1240 The Time-Sync-Token class is identified when a Principal authenticates through a time  
1241 synchronization token.

#### 1242 5.1.10.1 Associated Liberty URI

1243 <http://www.projectliberty.org/schemas/authctx/classes/Time-Sync-Token>

1244 **5.1.10.2 Class Schema**

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295

1296

1297

1298

1299

1300

1301

1302

1303

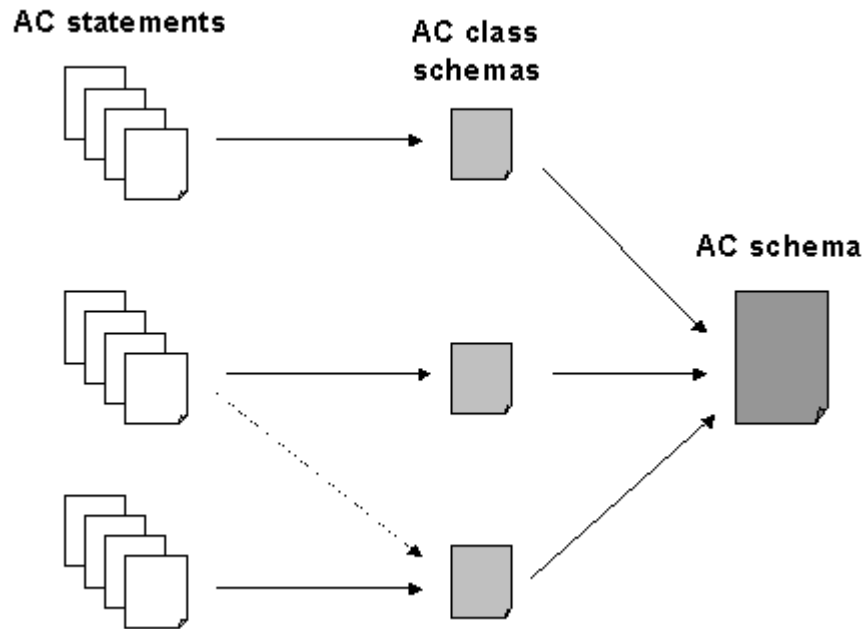
1304

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <annotation>
    <documentation> http://www.projectliberty.org/schemas/authctx/classes/Time-Sync-Token
  </documentation>
</annotation>
  <xs:element name="AuthenticationContextStatement">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1"
ref="AuthenticationMethod"/>
        <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="AuthenticationMethod">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="1" maxOccurs="1"
ref="PrincipalAuthenticationMechanism"/>
          <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Generation">
        <xs:complexType>
          <xs:attribute name="mechanism" fixed="principalchosen" />
        </xs:complexType>
      </xs:element>
      <xs:element name="PrincipalAuthenticationMechanism">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" ref="Token"/>
            <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="TimeSyncToken">
          <xs:complexType>
            <xs:attribute name="deviceType" fixed="hardware" />
            <xs:attribute name="seedLength" fixed="64" />
            <xs:attribute name="deviceInHand" fixed="true" />
          </xs:complexType>
        </xs:element>
        <xs:element name="Token">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="1" maxOccurs="1" ref="TimeSyncToken"/>
              <xs:any namespace="##any" minOccurs="0" maxOccurs="unbounded"
processContents="lax" /></xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:schema>
```

1305 **5.2 Authentication Context Schema**

1306 The relationship between authentication context statements, authentication context classes, and the  
1307 authentication context XML schema is shown in Figure 3.

1308



1309

1310 **Figure 3: Relationship between authentication context statements, classes, and XML schema**

1311

1312 Authentication context statements may conform to authentication context classes, which are  
1313 themselves logical subsets of the authentication context XML schema.

1314

1315

1316 **5.2.1 XML Schema**

1317

```
1318 <?xml version="1.0" encoding="UTF-8"?>  
1319 <schema targetNamespace="http://www.projectliberty.org/schemas/authctx/2002/05"  
1320 xmlns:xsd="http://www.w3.org/2001/XMLSchema" "  
1321 xmlns:AC="http://www.projectliberty.org/schemas/authctx/2002/05"  
1322 xmlns="http://www.w3.org/2001/XMLSchema" version="1.0">  
1323 <annotation>  
1324 <documentation> http://www.projectliberty.org/schemas/authctx/2002/05/  
1325 </documentation>  
1326 </annotation>  
1327 <element name="AuthenticationContextStatement">  
1328 <annotation>  
1329 <documentation>A claim made by an identity provider with respect to  
1330 the authentication context associated with an authentication assertion. </documentation>  
1331 </annotation>  
1332 <complexType>  
1333 <sequence>  
1334 <element ref="AC:Identification" minOccurs="0"/>  
1335 </sequence>  
1336 </complexType>  
1337 </element>
```

1337

```

1338         <element ref="AC:TechnicalProtection" minOccurs="0"/>
1339         <element ref="AC:OperationalProtection" minOccurs="0"/>
1340         <element ref="AC:AuthenticationMethod" minOccurs="0"/>
1341         <element ref="AC:GoverningAgreements" minOccurs="0"/>
1342         <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1343 processContents="lax" />
1344         </sequence>
1345         <attribute name="ID" type="ID"/>
1346     </complexType>
1347 </element>
1348
1349     <element name="Identification">
1350         <annotation>
1351             <documentation>Refers to those characteristics that describe the
1352 processes and mechanisms the
1353 identity provider uses to initially create an association between a
1354 Principal and the identity
1355 (or name) by which the Principal will be known</documentation>
1356         </annotation>
1357         <complexType>
1358             <sequence>
1359                 <element ref="AC:PhysicalVerification" minOccurs="0"/>
1360                 <element ref="AC:WrittenConsent" minOccurs="0"/>
1361                 <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1362 processContents="lax" />
1363             </sequence>
1364             <attribute name="nym">
1365                 <annotation>
1366                     <documentation>This attribute indicates whether or not
1367 the Identification mechanisms allow the
1368 actions of the Principal to be linked to an actual end
1369 user.</documentation>
1370                 </annotation>
1371                 <simpleType>
1372                     <restriction base="NMTOKEN">
1373                         <enumeration value="anonymity"/>
1374                         <enumeration value="verinymity"/>
1375                         <enumeration value="pseudonymity"/>
1376                     </restriction>
1377                 </simpleType>
1378             </attribute>
1379         </complexType>
1380     </element>
1381     <element name="PhysicalVerification">
1382         <annotation>
1383             <documentation>This element indicates that identification has been
1384 performed in a physical
1385 face-to-face meeting with the principal and not in an online manner.
1386 </documentation>
1387         </annotation>
1388         <complexType>
1389             <attribute name="credentialLevel">
1390                 <simpleType>
1391                     <restriction base="NMTOKEN">
1392                         <enumeration value="primary"/>
1393                         <enumeration value="secondary"/>
1394                     </restriction>
1395                 </simpleType>
1396             </attribute>
1397         </complexType>
1398     </element>
1399     <element name="WrittenConsent">
1400         <complexType><sequence><any namespace="##any" minOccurs="0"
1401 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1402     </element>
1403     <element name="TechnicalProtection">
1404         <annotation>

```

```

1405         <documentation>Refers to those characteristics that describe how the
1406 'secret' (the knowledge or possession of which allows the Principal to authenticate to the
1407 identity provider) is kept secure</documentation>
1408     </annotation>
1409     <complexType>
1410         <sequence>
1411             <element ref="AC:PrivateKeyProtection" minOccurs="0"/>
1412             <element ref="AC:SharedKeyProtection" minOccurs="0"/>
1413             <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1414 processContents="lax" />
1415         </sequence>
1416     </complexType>
1417 </element>
1418 <element name="SharedKeyProtection">
1419     <annotation>
1420         <documentation>This element indicates the types and strengths of
1421 facilities
1422         of a UA used to protect a shared secret key from unauthorized access
1423 and/or use.</documentation>
1424     </annotation>
1425     <complexType>
1426         <choice minOccurs="0">
1427             <element ref="AC:MobileDevice"/>
1428             <element ref="AC:MobileAuthCard"/>
1429         </choice>
1430     </complexType>
1431 </element>
1432 <element name="MobileDevice">
1433     <annotation>
1434         <documentation>This element indicates that the shared secret key is
1435 securely maintained in a mobile device
1436 (as opposed to being stored in a mobile authentication
1437 card).</documentation>
1438     </annotation>
1439     <complexType><sequence><any namespace="##any" minOccurs="0"
1440 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1441 </element>
1442 <element name="MobileAuthCard">
1443     <annotation>
1444         <documentation>This element indicates that the shared secret key is
1445 securely maintained in a mobile authentication card (e.g., a SIM card).</documentation>
1446     </annotation>
1447     <complexType><sequence><any namespace="##any" minOccurs="0"
1448 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1449 </element>
1450 <element name="PrivateKeyProtection">
1451     <annotation>
1452         <documentation>This element indicates the types and strengths of
1453 facilities
1454 of a UA used to protect a private key from unauthorized access and/or
1455 use.</documentation>
1456     </annotation>
1457     <complexType>
1458         <sequence>
1459             <element ref="AC:KeyActivation" minOccurs="0"/>
1460             <element ref="AC:KeyStorage" minOccurs="0"/>
1461             <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1462 processContents="lax" />
1463         </sequence>
1464     </complexType>
1465 </element>
1466 <element name="KeyActivation">
1467     <annotation>
1468         <documentation>The actions that must be performed before the private
1469 key can be used. </documentation>
1470     </annotation>
1471     <complexType>
1472         <choice>

```

```

1473         <element ref="AC:Password"/>
1474     </choice>
1475     </complexType>
1476 </element>
1477 <element name="KeyStorage">
1478     <annotation>
1479         <documentation>In which medium is the private key stored.
1480
1481         memory - the private key is stored in memory.
1482
1483         smartcard - the private key is stored in a smartcard and may not be
1484 read from that smartcard unless authorized.
1485
1486         token - the private key is stored in a hardware token and may not be
1487 read from that token unless authorized.
1488
1489         MobileAuthCard - the private key is stored in a mobile authentication
1490 card (e.g., SIM card) and may not be read from that token unless authorized.
1491     </documentation>
1492 </annotation>
1493 <complexType>
1494     <attribute name="medium" use="required">
1495         <simpleType>
1496             <restriction base="NMTOKEN">
1497                 <enumeration value="memory"/>
1498                 <enumeration value="smartcard"/>
1499                 <enumeration value="token"/>
1500                 <enumeration value="MobileAuthCard"/>
1501             </restriction>
1502         </simpleType>
1503     </attribute>
1504 </complexType>
1505 </element>
1506 <element name="Password">
1507     <annotation>
1508         <documentation>This element indicates that a password (or PIN or
1509 passphrase) has been used to authenticate the Principal or
1510 to gain access to some resource (for example, to gain access to the
1511 private key).</documentation>
1512 </annotation>
1513 <complexType>
1514     <sequence>
1515         <element ref="AC:Length" minOccurs="0"/>
1516         <element ref="AC:Generation" minOccurs="0"/>
1517         <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1518 processContents="lax" />
1519     </sequence>
1520 </complexType>
1521 </element>
1522 <element name="Token">
1523     <annotation>
1524         <documentation>This element indicates that a hardware or software
1525 token is
1526 used as a method of identifying the Principal.</documentation>
1527 </annotation>
1528 <complexType>
1529     <sequence>
1530         <element ref="AC:TimeSyncToken"/>
1531         <any namespace="##any" minOccurs="0"
1532 maxOccurs="unbounded" processContents="lax" />
1533     </sequence>
1534 </complexType>
1535 </element>
1536 <element name="TimeSyncToken">
1537     <annotation>
1538         <documentation>This element indicates that a time synchronization
1539 token is used to identify the Principal.
1540

```

```

1541 hardware - the time synchronization token has been implemented in
1542 hardware.
1543
1544 software - the time synchronization token has been implemented in
1545 software.
1546
1547 SeedLength - the length, in bits, of the random seed used in the time
1548 synchronization token.
1549 </documentation>
1550 </annotation>
1551 <complexType>
1552 <attribute name="DeviceType" use="required">
1553 <simpleType>
1554 <restriction base="NMTOKEN">
1555 <enumeration value="hardware"/>
1556 <enumeration value="software"/>
1557 </restriction>
1558 </simpleType>
1559 </attribute>
1560 <attribute name="SeedLength" type="integer" use="required"/>
1561 <attribute name="DeviceInHand" use="required">
1562 <simpleType>
1563 <restriction base="NMTOKEN">
1564 <enumeration value="true"/>
1565 <enumeration value="false"/>
1566 </restriction>
1567 </simpleType>
1568 </attribute>
1569 </complexType>
1570 </element>
1571 <element name="Smartcard">
1572 <annotation>
1573 <documentation>This element indicates that a smartcard is used to
1574 identity the Principal.</documentation>
1575 </annotation>
1576 <complexType><sequence><any namespace="##any" minOccurs="0"
1577 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1578 </element>
1579 <element name="Length">
1580 <annotation>
1581 <documentation>This element indicates the minimum and/or maximum ASCII
1582 length of the password which is enforced (by the UA or the IdP). In
1583 other words,
1584 this is the minimum and/or maximum number of ASCII characters required
1585 to represent a valid password.
1586
1587 min - the minimum number of ASCII characters required in a valid
1588 password, as enforced by the UA or the IdP.
1589
1590 max - the maximum number of ASCII characters required in a valid
1591 password, as enforced by the UA or the IdP.
1592 </documentation>
1593 </annotation>
1594 <complexType>
1595 <attribute name="min" type="integer" use="required"/>
1596 <attribute name="max" type="integer" use="optional"/>
1597 </complexType>
1598 </element>
1599 <element name="Generation">
1600 <annotation>
1601 <documentation>Indicates whether the password was chosen by the
1602 Principal or auto-supplied by the identity provider.
1603
1604 principalchosen - the Principal is allowed to choose the value of the
1605 password. This is true even if the initial password is chosen at
1606 random by the UA or the IdP and the Principal is then free to change
1607 the password.
1608

```

```

1609         automatic - the password is chosen by the UA or the IdP to be
1610 cryptographically strong in some sense, or to satisfy certain
1611         password rules, and that the Principal is not free to change it or to
1612 choose a new password.
1613
1614         </documentation>
1615     </annotation>
1616     <complexType>
1617         <attribute name="mechanism" use="required">
1618             <simpleType>
1619                 <restriction base="NMTOKEN">
1620                     <enumeration value="principalchosen"/>
1621                     <enumeration value="automatic"/>
1622                 </restriction>
1623             </simpleType>
1624         </attribute>
1625     </complexType>
1626 </element>
1627 <element name="AuthenticationMethod">
1628     <annotation>
1629         <documentation>Refers to those characteristics that define the
1630 mechanisms by which the Principal authenticates to the identity provider.</documentation>
1631     </annotation>
1632     <complexType>
1633         <sequence>
1634             <element ref="AC:PrincipalAuthenticationMechanism"/>
1635             <element ref="AC:Authenticator" minOccurs="0"/>
1636             <element ref="AC:AuthenticatorTransportProtocol"/>
1637             <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1638 processContents="lax" />
1639         </sequence>
1640     </complexType>
1641 </element>
1642 <element name="PrincipalAuthenticationMechanism">
1643     <annotation>
1644         <documentation>The method that a Principal employs to perform
1645 authentication to local system components.</documentation>
1646     </annotation>
1647     <complexType>
1648         <choice minOccurs="0" maxOccurs="unbounded">
1649             <element ref="AC:Password"/>
1650             <element ref="AC:Token"/>
1651             <element ref="AC:Smartcard"/>
1652         </choice>
1653     </complexType>
1654 </element>
1655 <element name="Authenticator">
1656     <annotation>
1657         <documentation>The method applied to validate a principal's
1658 authentication across a network </documentation>
1659     </annotation>
1660     <complexType>
1661         <choice minOccurs="0" maxOccurs="unbounded">
1662             <element ref="AC:PreviousSession"/>
1663             <element ref="AC:Dig-sig"/>
1664             <element ref="AC:ZeroKnowledge"/>
1665             <element ref="AC:SharedSecretChallengeResponse"/>
1666         </choice>
1667     </complexType>
1668 </element>
1669 <element name="PreviousSession">
1670     <annotation>
1671         <documentation>Indicates that the Principal has been strongly
1672 authenticated in a previous session during which
1673         the IdP has set a cookie in the UA. During the present session the
1674 Principal has only been authenticated by
1675         the UA returning the cookie to the IdP.</documentation>
1676     </annotation>

```



```

1677     <complexType><sequence><any namespace="##any" minOccurs="0"
1678 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1679     </element>
1680     <element name="ZeroKnowledge">
1681       <annotation>
1682         <documentation>This element indicates that the Principal has been
1683 authenticated by a zero knowledge
1684 technique as specified in ISO/IEC 9798-5.</documentation>
1685       </annotation>
1686       <complexType><sequence><any namespace="##any" minOccurs="0"
1687 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1688     </element>
1689     <element name="SharedSecretChallengeResponse">
1690       <annotation>
1691         <documentation>This element indicates that the Principal has been
1692 authenticated by a challenge-response
1693 protocol utilizing shared secret keys and symmetric
1694 cryptography.</documentation>
1695       </annotation>
1696       <complexType><sequence><any namespace="##any" minOccurs="0"
1697 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1698     </element>
1699     <element name="Dig-sig">
1700       <annotation>
1701         <documentation>This element indicates that the Principal has been
1702 authenticated by a mechanism which involves the Principal
1703 computing a digital signature over at least challenge data provided by
1704 the IdP.</documentation>
1705       </annotation>
1706       <complexType><sequence><any namespace="##any" minOccurs="0"
1707 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1708     </element>
1709     <element name="AuthenticatorTransportProtocol">
1710       <annotation>
1711         <documentation>The protocol across which Authenticator information is
1712 transferred to an identity provider verifier.</documentation>
1713       </annotation>
1714       <complexType>
1715         <choice minOccurs="0" maxOccurs="unbounded">
1716           <element ref="AC:HTTP"/>
1717           <element ref="AC:SSL"/>
1718           <element ref="AC:MobileNetwork"/>
1719           <element ref="AC:WTLS"/>
1720           <element ref="AC:IPSec"/>
1721         </choice>
1722       </complexType>
1723     </element>
1724     <element name="HTTP">
1725       <annotation>
1726         <documentation>This element indicates that the Authenticator has been
1727 transmitted
1728 using bare HTTP utilizing no additional security
1729 protocols.</documentation>
1730       </annotation>
1731       <complexType><sequence><any namespace="##any" minOccurs="0"
1732 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1733     </element>
1734     <element name="IPSec">
1735       <annotation>
1736         <documentation>This element indicates that the Authenticator has been
1737 transmitted
1738 using a transport mechanism protected by an IPSEC
1739 session.</documentation>
1740       </annotation>
1741       <complexType><sequence><any namespace="##any" minOccurs="0"
1742 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1743     </element>
1744     <element name="WTLS">

```

```
1745         <annotation>
1746             <documentation>This element indicates that the Authenticator has been
1747 transmitted
1748             using a transport mechanism protected by a WTLS
1749 session.</documentation>
1750         </annotation>
1751         <complexType><sequence><any namespace="##any" minOccurs="0"
1752 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1753     </element>
1754     <element name="MobileNetwork">
1755         <annotation>
1756             <documentation>This element indicates that the Authenticator has been
1757 transmitted
1758             solely across a mobile network using no additional security
1759 mechanism.</documentation>
1760         </annotation>
1761         <complexType><sequence><any namespace="##any" minOccurs="0"
1762 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1763     </element>
1764     <element name="SSL">
1765         <annotation>
1766             <documentation>This element indicates that the Authenticator has been
1767 transmitted
1768             using a transport mechanism protected by an SSL or TLS
1769 session.</documentation>
1770         </annotation>
1771         <complexType><sequence><any namespace="##any" minOccurs="0"
1772 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1773     </element>
1774     <element name="OperationalProtection">
1775         <annotation>
1776             <documentation>Refers to those characteristics that describe
1777 procedural security controls employed by the identity provider.</documentation>
1778         </annotation>
1779         <complexType>
1780             <sequence>
1781                 <element ref="AC:SecurityAudit" minOccurs="0"/>
1782                 <element ref="AC:DeactivationCallCenter" minOccurs="0"/>
1783                 <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1784 processContents="lax" />
1785             </sequence>
1786         </complexType>
1787     </element>
1788     <element name="SecurityAudit">
1789         <complexType>
1790             <sequence>
1791                 <element ref="AC:SwitchAudit" minOccurs="0"/>
1792                 <any namespace="##any" minOccurs="0" maxOccurs="unbounded"
1793 processContents="lax" />
1794             </sequence>
1795         </complexType>
1796     </element>
1797     <element name="SwitchAudit">
1798         <complexType><sequence><any namespace="##any" minOccurs="0"
1799 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1800     </element>
1801     <element name="DeactivationCallCenter">
1802         <complexType><sequence><any namespace="##any" minOccurs="0"
1803 maxOccurs="unbounded" processContents="lax" /></sequence></complexType>
1804     </element>
1805     <element name="GoverningAgreements">
1806         <annotation>
1807             <documentation>Provides a mechanism for linking to external (likely
1808 human readable) documents in which the identity provider can define business level
1809 authentication context, e.g. liability constraints, contractual
1810 obligations.</documentation>
1811         </annotation>
1812     </complexType>
```

```
1813         <sequence>
1814             <element ref="AC:GoverningAgreementRef"/>
1815         </sequence>
1816     </complexType>
1817 </element>
1818 <element name="GoverningAgreementRef">
1819     <complexType>
1820         <attribute name="governingAgreementRef" type="anyURI" use="required"/>
1821     </complexType>
1822 </element>
1823 </schema>
```

## 1824 6 References

- 1825 [LibertyGloss] Mauldin, H., "Liberty Glossary," [https://66.34.4.93/members/technology](https://66.34.4.93/members/technology_expert_group/draft-liberty-tech-glossary-01.doc)  
1826 [expert\\_group/draft-liberty-tech-glossary-01.doc](https://66.34.4.93/members/technology_expert_group/draft-liberty-tech-glossary-01.doc), April 2002.
- 1827 [LibertyProtSchema] Beatty, J., "Liberty Protocols and Schemas Specification,"  
1828 [https://66.34.4.93/members/technology\\_20expert\\_group/architecture/draft-](https://66.34.4.93/members/technology_20expert_group/architecture/draft-liberty-architecture-protocols-schemas-04.doc)  
1829 [liberty-architecture-protocols-schemas-04.doc](https://66.34.4.93/members/technology_20expert_group/architecture/draft-liberty-architecture-protocols-schemas-04.doc), April 2002.
- 1830 [PDS] Santesson S. et al, "Internet X.509 Public Key Infrastructure PKI  
1831 Disclosure Statement," <http://www.verisign.com/repository/pds.txt>.
- 1832 [RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels,"  
1833 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- 1834 [RFC2527] Chokhani S. et al, "Internet X.509 Public Key Infrastructure Certificate  
1835 Policy and Certification Practices Framework,"  
1836 <http://www.ietf.org/rfc/rfc2527.txt?number=2527>.
- 1837 [SAMLBind] P. Mishra et al., "Bindings and Profiles for the OASIS Security Assertion  
1838 Markup Language (SAML)," [http://www.oasis-](http://www.oasis-open.org/committees/security/docs/draft-sstc-bindings-model-11.pdf)  
1839 [open.org/committees/security/docs/draft-sstc-bindings-model-11.pdf](http://www.oasis-open.org/committees/security/docs/draft-sstc-bindings-model-11.pdf),  
1840 OASIS, January 2002.
- 1841 [SAMLCore] Hallam-Baker, P., et al., "Assertions and Protocol for the OASIS Security  
1842 Assertion Markup Language (SAML)," [http://www.oasis-](http://www.oasis-open.org/committees/security/docs/draft-sstc-core-31.pdf)  
1843 [open.org/committees/security/docs/draft-sstc-core-31.pdf](http://www.oasis-open.org/committees/security/docs/draft-sstc-core-31.pdf), OASIS, April  
1844 2002.
- 1845 [Schema1] H. S. Thompson et al., "XML Schema Part 1: Structures,"  
1846 <http://www.w3.org/TR/xmlschema-1/>, World Wide Web Consortium  
1847 Recommendation, May 2001.
- 1848