

November 23-26, 2004**Shenzhen, China**

Source: Ericsson
Title: Ericsson Comments to Nokia's Tdoc S3-040980 on "Liberty and GAA relationship"
Agenda item: 6.9.1 (GAA)
Document for: Information

1 Introduction

The relationship between Liberty and GAA has been brought up during several SA3 meetings. As GAA has become mature, its relationship and possible overlap areas with Liberty can now be identified more clearly. In this contribution, we point out the details on the possible relationship between GAA and Liberty, and invite further comments on the details of the relationship and possible interaction methods.

Below is a list of previous contributions on Liberty and GAA relationship:

- Discussion document on Bootstrapping and Subscriber Certificate Use Cases that includes the Liberty use case (SA3-030407, Nokia) from SA3#29 meeting
 - Introductory Presentation on Liberty Alliance (SA3-030764, Nokia) SA3#31 meeting
 - Presentation on synergies between Liberty Alliance and 3GPP (SA3-030719, Nokia) SA#31 meeting
 - Discussion document on the Relationship between GAA and Liberty (SA3-040813, Vodafone) SA#35 meeting
-

2 Discussion

Liberty Alliance Project (LAP) defines in the [Identity Federation Framework, ID-FF](http://www.projectliberty.org/resources/specifications.php#box1), specification a family of protocols (<http://www.projectliberty.org/resources/specifications.php#box1>). Those protocols enable reuse of an authentication done at an Identity Provider (IdP) for user authentication towards a Service Provider, that trusts this authentication confirmation of the IdP. LAP does not specify the actual means of authentication, but leaves that for agreement between the business partners. The Liberty Alliance specifications just use the provided authentication means without restricting the type of authentication mechanism, to enable easy integration with the existing infrastructure and legacy authentication mechanism (e.g. username / password). GAA, which is specific to mobile subscribers, is one of the methods that can be used to provide the authentication mechanism for Liberty. For example, the Liberty Identity Provider (IdP) can act as a NAF, as depicted in figure 1 (see also S3-030407, and S3-040719). This configuration setting would not provide any overlap in provided functionality. GAA would be used to authenticate the UE towards the IdP (via HTTP digest authentication). After the HTTP digest authentication, the LAP protocols towards Liberty SPs would be used, as outlined in the message flow diagram below. An advantage of this mobile specific configuration is that the authentication will then be more secure and user-friendly as compared to passwords.

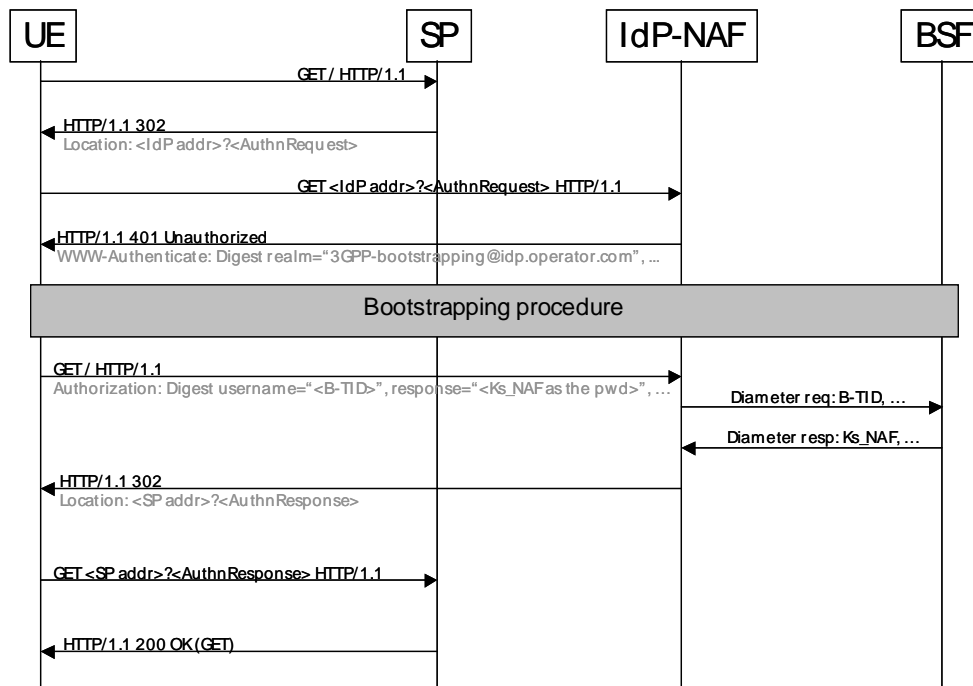


Figure 1. Example of Liberty IdP working as a NAF

[Ericsson fully agrees with this view on how a particular deployment would be able to combine 3GPP-GAA and LAP technologies in order to perform user authentication. This is a feasible set-up based on 2 standard technologies completely dependant on deployment policies and not requiring in our view any additional standardization.](#)

3 Potential topics for further studies

First, Vodafone contribution (S3-040831) stated that the BSF provides a similar functionality then the Liberty IdP. Hence the IdP could also be co-located with GAA's BSF. The details of this approach are yet unknown and require further studies.

[To this point, Ericsson would like to highlight in the first place that this particular combination of standard functionalities into one particular implementation is something perfectly possible without the need for any specification process.](#)

[Secondly and while agreeing that IdP and BSF performs similar functionality, Ericsson would like to point that this is not quite the same. Within the scope of a Liberty ID-FF SSO process, the IdP is responsible of authenticating the user \(rather the role of the GAA-NAF function\) while in the scope of GAA, BSF is responsible of authenticating the user as well but with the ultimate goal to be able to generate some key material that will be later on distributed and utilized in a subsequent user authentication process between the UE and the NAF. This difference in concept is already highlighted by the original Nokia's contribution in its second topic below.](#)

[This means that if an LAP-IdP implementation chooses to bundle 3GPP-NAF and BSF functionality the result would be that the same implementation would be conducting two consecutive user authentication processes ... one acting as BSF and another one acting as NAF.](#)

[This situation will not be very optimal from an implementation point of view and less optimal from an standards point of view.](#)

Second potential topic for further studies is the case where Liberty SP and GAA NAF are co-located, then the SP supports both Liberty protocols and GAA's NAF functionality. There is a functionality needed which authentication method it should use based on the Service Provider requirements and UE's capabilities. If the Liberty Service Provider (SP) is compared to GAA's Network Application Function (NAF), the difference is that with the Liberty protocols the SP gets authentication confirmation, but it does not get the keys of the terminal. In GAA the NAF gets the keys and the authentication confirmation. With Liberty, the SP has to verify the SAML assertion given by the IdP., In GAA, the NAF has to authenticate the UE using the GAA credentials (i.e., B-TID, and Ks_NAF).

Right, here it is clearly presented the conceptual difference between the two authentication models used by Liberty and 3GPP-GAA. SPs shall be able to work with the two of them and it is assumed that SPs will use one or the other depending on (deployment/business) policies amongst other things. From GAA TR 33.919 ...

“Depending on network configuration and policies of the operator, an AS or an AP will be able to use any of the alternatives provided by GAA or even any other user authentication mechanisms specified outside of 3GPP if such mechanisms are at their disposal. It is therefore assumed that an AS and an AP should be able to take the decision what parts of GAA shall be used if any.”

GAA bootstrapping initiation process already provides coverage for this. When an SP receives a service access request (without GAA bootstrapping info) it shall decide whether to redirect the UE to the BSF or execute any other action like for example initiate a Liberty ID-FF Authentication Request instead.

Third, Liberty Alliance project also does not provide guidance on the interfaces between IdP and the backend, “legacy” databases (or how the IdP is organized “internally”). Thus, if the “legacy” database is HSS, the GAA Zh interface might be a good complementing functionality. But further studies might be required there.

The reuse of specific components of 3GPP-GAA could be something desirable. In particular the reuse of GAA Zh interface in order for the IdP to be able to fetch authentication vectors from e.g. HSS would be beneficial. Furthermore, the use of these fetched AVs via Zh interface in order for the IdP to further execute user authentication via http-digest-AKA would be also interesting.

However, Ericsson wonders which additional standardization work would be required here. It is out of the scope of Liberty to specify the details of the different authentication mechanisms being used and 3GPP already provides the interfaces that a possible deployment/implementation might require.

An open issue with GAA and Liberty interworking is that Liberty Alliance requires in their SAML tokens timestamps for the authentication. However, GAA currently only offers an expiry timestamp of the bootstrapping key to NAFs, not the timestamp of the exact timestamp when the bootstrapping was performed.

The timestamps required by the Liberty Alliance SAML assertions corresponds to the actual time when the IdP authentication took place. This is a data perfectly known by the IdP since, this is the entity that executes or orders the execution of the authentication process. An IdP also acting as a NAF will be therefore capable of providing the correct information for the Liberty interactions as this will be a different data from the actual timestamp of the bootstrapping process.

Finally, Liberty Alliance has introduced the notion of authentication context as a quality measure for the authentication, and they specified some examples for authentication contexts. 3GPP could specify a GAA specific authentication context for Liberty.

3GPP-GAA specifications provide a number of mechanisms for distribution of credentials between UE and NAF functions (via BSF function) and with the exception of the mechanisms defined at TS 33.222 for the set up of a secure HTTPS session, they do not further define the actual authentication mechanism that the NAF function ultimately executes in order to authenticate the UE using the credentials obtained via GAA mechanisms.

The authentication context information included within a Liberty SAML assertion corresponds to the details of the authentication mechanism performed (or ordered) by the IdP (the details of the authentication mechanism executed by the NAF in this case). It will be therefore not 100% correct to define an authentication context for GAA since GAA is not primarily addressing the NAF authentication mechanism but rather the means for the NAF to get the credentials for a the actual user authentication.

It would make sense however to investigate if there is any extra work required in order for an IdP to use http-digest-AKA as a valid authentication mechanism within the Liberty ID-FF. After an initial analysis, the already defined authentication contexts for “MobileOneFactorUnregistered” and “MobileOneFactorContract” will be already providing good coverage for a user authentication performed by the IdP using http-digest-AKA. Even, http-digest-AKA might be also already considered as a fundamental step for the “MobileTwoFactorUnregistered” and “MobileTwoFactorContract” authentication contexts.

Please refer to chapters 5.2.3 to 5.2.6 in Liberty ID-FF authentication context specification. The whole specification is attached to this contribution and can be also found at the public Liberty site ...

<http://www.projectliberty.org/specs/draft-liberty-authentication-context-1.2-errata-v1.0.pdf>

4 Conclusion

As briefly described in this paper, the Liberty Alliance protocols can use GAA as authentication method for mobile subscribers. The authentication will then be more secure and user-friendly as compared to passwords. Most straightforward way is to implement this is to add NAF functionality to Liberty IdP. To further support this functionality, SA3 could specify a GAA specific authentication context for Liberty.

Possible Liberty Alliance and GAA overlaps and other synergies are FFS.

Ericsson conclusion is that it is perfectly possible to combine 3GPP-GAA and LAP technologies with no extra standardization effort. All possible combinations (LAP IdP as NAF and/or BSF) are deployable although some of them will not make sense from an implementation point of view (LAP IdP acting both as NAF and BSF).

In this area, Ericsson would rather ensure that a Liberty IdP is able to use http-digest-AKA authentication as a valid user authentication mechanism within the Liberty ID-FF. It should be investigated what extra standardization efforts should be required (if any) in order to achieve that and where the corresponding activities should be executed (3GPP-SA3 and/or Liberty).



Liberty ID-FF Authentication Context Specification

Version: 1.2-errata-v1.0

Editors:

Paul Madsen, Entrust, Inc.

Contributors:

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

Abstract:

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

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

1

Notice

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

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

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

25 Liberty Alliance Project
26 Licensing Administrator
27 c/o IEEE-ISTO
28 445 Hoes Lane
29 Piscataway, NJ 08855-1331, USA
30 info@projectliberty.org

31 **Contents**

32 [1. About this Document](#) 4
33 [2. Overview](#) 5
34 [3. Authentication Context](#) 6
35 [4. Authentication Context Statement](#) 7
36 [5. Authentication Context Classes](#) 19
37 [References](#) 39

38 1. About this Document

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

41 1.1. Notation and Terminology

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

45 1.1.1. Notational Conventions

46 Note: Phrases and numbers in brackets [] refer to other documents; details of these references can be found in
47 [References](#) (at the end of this document).

48 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT",
49 "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

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

53 Listings of XML schemas appear like this.

```
54  
55         <?xml version="1.0" encoding="UTF-8"?>  
56 <xs:schema targetNamespace="urn:liberty:ac:2003-08"  
57     xmlns:xs="http://www.w3.org/2001/XMLSchema"  
58     xmlns="urn:liberty:ac:2003-08">  
59  
60     <!-- Add Stuff Here -->  
61  
62 </xs:schema>  
63  
64
```

65 1.1.2. Namespaces

66 The following namespaces are referred to in this document:

67 **Table 1. Namespaces**

Prefix	Namespace
ac	urn:liberty:ac:1.2
lib	urn:liberty:iff:1.2
xs	http://www.w3.org/2001/XMLSchema
xsi	http://www.w3.org/2001/XMLSchema-instance

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

70 2. Overview

71 Liberty will not prescribe a single technology, protocol, or policy for the processes by which identity providers issue
72 identities to Principals and by which those Principals subsequently authenticate themselves to the identity provider.
73 Different identity providers will choose different technologies, follow different processes, and be bound by different
74 legal obligations with respect to how they authenticate Principals.

75 The choices that an identity provider makes here will be driven in large part by the requirements of the service providers
76 with which the identity provider has affiliated into a circle of trust. These requirements themselves will be determined
77 by the nature of the service (that is, the sensitivity of any information exchanged, the associated financial value, the
78 service providers risk tolerance, etc.) that the service provider will be providing to the Principal.

79 Consequently, for anything other than trivial services, if the service provider is to place sufficient confidence in the
80 authentication assertions it receives from an identity provider, it will be necessary for the service provider to know
81 which technologies, protocols, and processes were used or followed for the original authentication mechanism on
82 which the authentication assertion is based. Armed with this information and trusting the origin of the actual assertion,
83 the service provider will be better able to make an informed entitlements decision regarding what services the subject
84 of the authentication assertion should be allowed to access.

85 *Authentication context* is defined as the information, additional to the authentication assertion itself, that the service
86 provider may require before it makes an entitlements decision with respect to an authentication assertion.

87 **3. Authentication Context**

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

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

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

99 4. Authentication Context Statement

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

103 4.1. Authentication Context Statement Data Model

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

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

118 4.2. Authentication Context Statement Schema

119 This section lists the complete Authentication Context XML Schema.

```
120
121 <?xml version="1.0" encoding="UTF-8"?>
122 <xs:schema targetNamespace="urn:liberty:ac:2003-08"
123   xmlns:xs="http://www.w3.org/2001/XMLSchema"
124   xmlns="urn:liberty:ac:2003-08">
125
126   <!-- added to get the Extension element -->
127   <xs:include schemaLocation="liberty-utility-v1.0.xsd"/>
128
129   <xs:annotation>
130     <xs:documentation> ### IMPORTANT NOTICE ###
131
132     The source code in this XSD file was excerpted verbatim from:
133
134     Liberty Authentication Context Specification
135     Version 1.2-errata-v1.0
136     12 September 2004
137
138     Copyright (c) 2004 Liberty Alliance participants, see
139     http://www.projectliberty.org/specs/idff_copyrights.html
140   </xs:documentation>
141 </xs:annotation>
142 <xs:element name="AuthenticationContextStatement" type="AuthenticationContextStatementType">
143   <xs:annotation>
144     <xs:documentation>
145       A particular assertion on an identity
146       provider's part with respect to the authentication
```

```

147         context associated with an authentication assertion.
148     </xs:documentation>
149 </xs:annotation>
150 </xs:element>
151 <xs:element name="Identification" type="IdentificationType">
152     <xs:annotation>
153         <xs:documentation>
154             Refers to those characteristics that describe the processes and mechanisms
155             the Authentication Authority uses to initially create an association between a Principal
156             and the identity (or name) by which the Principal will be known
157         </xs:documentation>
158     </xs:annotation>
159 </xs:element>
160 <xs:element name="PhysicalVerification">
161     <xs:annotation>
162         <xs:documentation>
163             This element indicates that identification has been performed in a physical
164             face-to-face meeting with the principal and not in an online manner.
165         </xs:documentation>
166     </xs:annotation>
167     <xs:complexType>
168         <xs:attribute name="credentialLevel">
169             <xs:simpleType>
170                 <xs:restriction base="xs:NMTOKEN">
171                     <xs:enumeration value="primary"/>
172                     <xs:enumeration value="secondary"/>
173                 </xs:restriction>
174             </xs:simpleType>
175         </xs:attribute>
176     </xs:complexType>
177 </xs:element>
178 <xs:element name="WrittenConsent">
179     <xs:complexType>
180         <xs:sequence>
181             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
182         </xs:sequence>
183     </xs:complexType>
184 </xs:element>
185 <xs:element name="TechnicalProtection" type="TechnicalProtectionType">
186     <xs:annotation>
187         <xs:documentation>
188             Refers to those characteristics that describe how the 'secret' (the knowledge or
189             possession of which allows the Principal to authenticate to the Authentication
190             Authority) is kept secure
191         </xs:documentation>
192     </xs:annotation>
193 </xs:element>
194 <xs:element name="SecretKeyProtection" type="SecretKeyProtectionType">
195     <xs:annotation>
196         <xs:documentation>
197             This element indicates the types and strengths of facilities
198             of a UA used to protect a shared secret key from unauthorized access and/or use.
199         </xs:documentation>
200     </xs:annotation>
201 </xs:element>
202 <xs:element name="PrivateKeyProtection" type="PrivateKeyProtectionType">
203     <xs:annotation>
204         <xs:documentation>
205             This element indicates the types and strengths of facilities
206             of a UA used to protect a private key from unauthorized access and/or use.
207         </xs:documentation>
208     </xs:annotation>
209 </xs:element>
210 <xs:element name="KeyActivation" type="KeyActivationType">
211     <xs:annotation>
212         <xs:documentation>The actions that must be performed before the private key can be used.↵
213 </xs:documentation>

```

```

214     </xs:annotation>
215 </xs:element>
216 <xs:element name="KeySharing" type="KeySharingType">
217   <xs:annotation>
218     <xs:documentation>
219       Whether or not the private key is shared with the certificate authority.
220     </xs:documentation>
221   </xs:annotation>
222 </xs:element>
223 <xs:element name="KeyStorage" type="KeyStorageType">
224   <xs:annotation>
225     <xs:documentation>
226       In which medium is the key stored.
227       memory - the key is stored in memory.
228       smartcard - the key is stored in a smartcard.
229       token - the key is stored in a hardware token.
230       MobileDevice - the key is stored in a mobile device.
231       MobileAuthCard - the key is stored in a mobile authentication card.
232     </xs:documentation>
233   </xs:annotation>
234 </xs:element>
235 <xs:element name="Password" type="PasswordType">
236   <xs:annotation>
237     <xs:documentation>
238       This element indicates that a password (or passphrase) has been used to
239       authenticate the Principal to a remote system.
240     </xs:documentation>
241   </xs:annotation>
242 </xs:element>
243 <xs:element name="ActivationPin" type="ActivationPinType">
244   <xs:annotation>
245     <xs:documentation>
246       This element indicates that a Pin (Personal Identification Number) has been used
247       to authenticate the Principal to some local system in order to activate a key.
248     </xs:documentation>
249   </xs:annotation>
250 </xs:element>
251 <xs:element name="Token" type="TokenType">
252   <xs:annotation>
253     <xs:documentation>
254       This element indicates that a hardware or software token is used
255       as a method of identifying the Principal.
256     </xs:documentation>
257   </xs:annotation>
258 </xs:element>
259 <xs:element name="TimeSyncToken" type="TimeSyncTokenType">
260   <xs:annotation>
261     <xs:documentation>
262       This element indicates that a time synchronization
263       token is used to identify the Principal.
264       hardware - the time synchronization token has been implemented in hardware.
265       software - the time synchronization token has been implemented in software.
266       SeedLength - the length, in bits, of the random seed used in the time synchronization
267       token.
268     </xs:documentation>
269   </xs:annotation>
270 </xs:element>
271 <xs:element name="Smartcard">
272   <xs:annotation>
273     <xs:documentation>
274       This element indicates that a smartcard is used to identity the Principal.
275     </xs:documentation>
276   </xs:annotation>
277   <xs:complexType>
278     <xs:sequence>
279       <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
280     </xs:sequence>

```

```

281     </xs:complexType>
282 </xs:element>
283 <xs:element name="Length" type="LengthType">
284   <xs:annotation>
285     <xs:documentation>
286       This element indicates the minimum and/or maximum ASCII length of the password
287       which is enforced (by the UA or the IdP). In other words, this is the minimum
288       and/or maximum number of ASCII characters required to represent a valid password.
289       min - the minimum number of ASCII characters required in a valid password,
290           as enforced by the UA or the IdP.
291       max - the maximum number of ASCII characters required in a valid password,
292           as enforced by the UA or the IdP.
293     </xs:documentation>
294   </xs:annotation>
295 </xs:element>
296 <xs:element name="ActivationLimit" type="ActivationLimitType">
297   <xs:annotation>
298     <xs:documentation>
299       This element indicates the length of time for which an PIN-based authentication is valid.
300     </xs:documentation>
301   </xs:annotation>
302 </xs:element>
303 <xs:element name="Generation">
304   <xs:annotation>
305     <xs:documentation>
306       Indicates whether the password was chosen by the Principal or auto-supplied
307       by the Authentication Authority.
308       principalchosen - the Principal is allowed to choose the value of the password.
309       This is true even if the initial password is chosen at random by the UA or the
310       IdP and the Principal is then free to change the password.
311       automatic - the password is chosen by the UA or the IdP to be cryptographically strong
312       in some sense, or to satisfy certain password rules, and that the Principal
313       is not free to change it or to choose a new password.
314     </xs:documentation>
315   </xs:annotation>
316   <xs:complexType>
317     <xs:attribute name="mechanism" use="required">
318       <xs:simpleType>
319         <xs:restriction base="xs:NMTOKEN">
320           <xs:enumeration value="principalchosen"/>
321           <xs:enumeration value="automatic"/>
322         </xs:restriction>
323       </xs:simpleType>
324     </xs:attribute>
325   </xs:complexType>
326 </xs:element>
327 <xs:element name="AuthenticationMethod" type="AuthenticationMethodType">
328   <xs:annotation>
329     <xs:documentation>
330       Refers to those characteristics that define the mechanisms by which the
331       Principal authenticates to the Authentication Authority.
332     </xs:documentation>
333   </xs:annotation>
334 </xs:element>
335 <xs:element name="PrincipalAuthenticationMechanis
336 m" type="PrincipalAuthenticationMechanismType">
337   <xs:annotation>
338     <xs:documentation>
339       The method that a Principal employs to perform authentication to local system components.
340     </xs:documentation>
341   </xs:annotation>
342 </xs:element>
343 <xs:element name="Authenticator" type="AuthenticatorType">
344   <xs:annotation>
345     <xs:documentation>
346       The method applied to validate a principal's authentication across a network
347     </xs:documentation>

```

```

348     </xs:annotation>
349   </xs:element>
350   <xs:element name="PreviousSession">
351     <xs:annotation>
352       <xs:documentation>
353         Indicates that the Principal has been strongly authenticated in a previous session,
354         during which the IdP has set a cookie in the UA. During the present session the Principal has only
355         been authenticated by the UA returning the cookie to the IdP.
356       </xs:documentation>
357     </xs:annotation>
358     <xs:complexType>
359       <xs:sequence>
360         <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
361       </xs:sequence>
362     </xs:complexType>
363   </xs:element>
364
365   <xs:element name="ResumeSession">
366     <xs:annotation>
367       <xs:documentation>
368         Rather like PreviousSession but using stronger security. A secret that was established
369         in a previous session with the Authentication Authority has been cached by the local system and is
370         now re-used (e.g. a Master Secret is used to derive new session keys in TLS, SSL, WTLS).
371       </xs:documentation>
372     </xs:annotation>
373     <xs:complexType>
374       <xs:sequence>
375         <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
376       </xs:sequence>
377     </xs:complexType>
378   </xs:element>
379
380   <xs:element name="ZeroKnowledge">
381     <xs:annotation>
382       <xs:documentation>
383         This element indicates that the Principal has been authenticated by a zero knowledge
384         technique as specified in ISO/IEC 9798-5.
385       </xs:documentation>
386     </xs:annotation>
387     <xs:complexType>
388       <xs:sequence>
389         <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
390       </xs:sequence>
391     </xs:complexType>
392   </xs:element>
393   <xs:element name="SharedSecretChallengeResponse">
394     <xs:annotation>
395       <xs:documentation>
396         This element indicates that the Principal has been authenticated by a challenge-response
397         protocol utilizing shared secret keys and symmetric cryptography.
398       </xs:documentation>
399     </xs:annotation>
400     <xs:complexType>
401       <xs:sequence>
402         <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
403       </xs:sequence>
404     </xs:complexType>
405   </xs:element>
406   <xs:element name="DigSig">
407     <xs:annotation>
408       <xs:documentation>
409         This element indicates that the Principal has been authenticated by a mechanism which
410         involves the Principal computing a digital signature over at least challenge data provided by the IdP.
411       </xs:documentation>
412     </xs:annotation>
413     <xs:complexType>
414       <xs:sequence>

```

```

415         <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
416     </xs:sequence>
417 </xs:complexType>
418 </xs:element>
419
420 <xs:element name="IPAddress">
421     <xs:annotation>
422         <xs:documentation>
423             This element indicates that the Principal has been authenticated through connection from
424 a particular IP address.
425         </xs:documentation>
426     </xs:annotation>
427     <xs:complexType>
428         <xs:sequence>
429             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
430         </xs:sequence>
431     </xs:complexType>
432 </xs:element>
433
434 <xs:element name="AsymmetricDecryption">
435     <xs:annotation>
436         <xs:documentation>
437             The local system has a private key but it is used in decryption mode, rather than
438 signature mode. For example, the Authentication Authority generates a secret and encrypts it using
439 the local system's public key: the local system then proves it has decrypted the secret.
440         </xs:documentation>
441     </xs:annotation>
442     <xs:complexType>
443         <xs:sequence>
444             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
445         </xs:sequence>
446     </xs:complexType>
447 </xs:element>
448
449 <xs:element name="AsymmetricKeyAgreement">
450     <xs:annotation>
451         <xs:documentation>
452             The local system has a private key and uses it for shared secret key agreement with
453 the Authentication Authority (e.g. via Diffie Helman).
454         </xs:documentation>
455     </xs:annotation>
456     <xs:complexType>
457         <xs:sequence>
458             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
459         </xs:sequence>
460     </xs:complexType>
461 </xs:element>
462
463 <xs:element name="SharedSecretDynamicPlaintext">
464     <xs:annotation>
465         <xs:documentation>
466             The local system and Authentication Authority share a secret key. The local system
467 uses this to encrypt a randomized string to pass to the Authentication Authority.
468         </xs:documentation>
469     </xs:annotation>
470     <xs:complexType>
471         <xs:sequence>
472             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
473         </xs:sequence>
474     </xs:complexType>
475 </xs:element>
476
477 <xs:element name="AuthenticatorTransportProtocol" type="AuthenticatorTransportProtocolType">
478     <xs:annotation>
479         <xs:documentation>
480             The protocol across which Authenticator information is transferred to an Authentication
481 Authority verifier.
  
```



```

482         </xs:documentation>
483     </xs:annotation>
484 </xs:element>
485 <xs:element name="HTTP">
486     <xs:annotation>
487         <xs:documentation>
488             This element indicates that the Authenticator has been transmitted using bare HTTP,
489 utilizing no additional security protocols.
490         </xs:documentation>
491     </xs:annotation>
492     <xs:complexType>
493         <xs:sequence>
494             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
495         </xs:sequence>
496     </xs:complexType>
497 </xs:element>
498 <xs:element name="IPSec">
499     <xs:annotation>
500         <xs:documentation>
501             This element indicates that the Authenticator has been transmitted using a transport
502 mechanism protected by an IPSEC session.
503         </xs:documentation>
504     </xs:annotation>
505     <xs:complexType>
506         <xs:sequence>
507             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
508         </xs:sequence>
509     </xs:complexType>
510 </xs:element>
511 <xs:element name="WTLS">
512     <xs:annotation>
513         <xs:documentation>
514             This element indicates that the Authenticator has been transmitted using a transport
515 mechanism protected by a WTLS session.
516         </xs:documentation>
517     </xs:annotation>
518     <xs:complexType>
519         <xs:sequence>
520             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
521         </xs:sequence>
522     </xs:complexType>
523 </xs:element>
524 <xs:element name="MobileNetworkNoEncryption">
525     <xs:annotation>
526         <xs:documentation>
527             This element indicates that the Authenticator has been transmitted solely across a
528 mobile network using no additional security mechanism.
529         </xs:documentation>
530     </xs:annotation>
531     <xs:complexType>
532         <xs:sequence>
533             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
534         </xs:sequence>
535     </xs:complexType>
536 </xs:element>
537 <xs:element name="MobileNetworkRadioEncryption">
538     <xs:complexType>
539         <xs:sequence>
540             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
541         </xs:sequence>
542     </xs:complexType>
543 </xs:element>
544 <xs:element name="MobileNetworkEndToEndEncryption">
545     <xs:complexType>
546         <xs:sequence>
547             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
548         </xs:sequence>
    
```

```

549     </xs:complexType>
550 </xs:element>
551
552 <xs:element name="SSL">
553   <xs:annotation>
554     <xs:documentation>
555       This element indicates that the Authenticator has been transmitted using a transport_
556 mechanism protected by an SSL or TLS session.
557     </xs:documentation>
558   </xs:annotation>
559   <xs:complexType>
560     <xs:sequence>
561       <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
562     </xs:sequence>
563   </xs:complexType>
564 </xs:element>
565 <xs:element name="OperationalProtection" type="OperationalProtectionType">
566   <xs:annotation>
567     <xs:documentation>
568       Refers to those characteristics that describe procedural security controls employed by_
569 the Authentication Authority.
570     </xs:documentation>
571   </xs:annotation>
572 </xs:element>
573 <xs:element name="SecurityAudit" type="SecurityAuditType" />
574 <xs:element name="SwitchAudit">
575   <xs:complexType>
576     <xs:sequence>
577       <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
578     </xs:sequence>
579   </xs:complexType>
580 </xs:element>
581 <xs:element name="DeactivationCallCenter">
582   <xs:complexType>
583     <xs:sequence>
584       <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
585     </xs:sequence>
586   </xs:complexType>
587 </xs:element>
588 <xs:element name="GoverningAgreements" type="GoverningAgreementsType">
589   <xs:annotation>
590     <xs:documentation>
591       Provides a mechanism for linking to external (likely human readable) documents in which_
592 additional business agreements,(e.g. liability constraints, obligations, etc) can be placed.
593     </xs:documentation>
594   </xs:annotation>
595 </xs:element>
596 <xs:element name="GoverningAgreementRef" type="GoverningAgreementRefType" />
597 <xs:element name="AuthenticatingAuthority" type="AuthenticatingAuthorityType">
598   <xs:annotation>
599     <xs:documentation>
600       The Authority that originally authenticated the Principal.
601     </xs:documentation>
602   </xs:annotation>
603 </xs:element>
604 <xs:complexType name="IdentificationType">
605   <xs:sequence>
606     <xs:element ref="PhysicalVerification" minOccurs="0" />
607     <xs:element ref="WrittenConsent" minOccurs="0" />
608     <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
609   </xs:sequence>
610   <xs:attribute name="nym">
611     <xs:annotation>
612       <xs:documentation>
613         This attribute indicates whether or not the Identification mechanisms allow the_
614 actions of the Principal to be linked to an actual end user.
615       </xs:documentation>

```

```

616         </xs:annotation>
617         <xs:simpleType>
618             <xs:restriction base="xs:NMTOKEN">
619                 <xs:enumeration value="anonymity" />
620                 <xs:enumeration value="verinymity" />
621                 <xs:enumeration value="pseudonymity" />
622             </xs:restriction>
623         </xs:simpleType>
624     </xs:attribute>
625 </xs:complexType>
626 <xs:complexType name="GoverningAgreementsType">
627     <xs:sequence>
628         <xs:element ref="GoverningAgreementRef" maxOccurs="unbounded" />
629     </xs:sequence>
630 </xs:complexType>
631 <xs:complexType name="GoverningAgreementRefType">
632     <xs:attribute name="governingAgreementRef" type="xs:anyURI" use="required" />
633 </xs:complexType>
634 <xs:complexType name="AuthenticatingAuthorityType">
635     <xs:sequence>
636         <xs:element ref="GoverningAgreements" />
637     </xs:sequence>
638     <xs:attribute name="ID" type="xs:anyURI" use="required" />
639 </xs:complexType>
640 <xs:complexType name="AuthenticatorTransportProtocolType">
641     <xs:choice>
642         <xs:element ref="HTTP" />
643         <xs:element ref="SSL" />
644         <xs:element ref="MobileNetworkNoEncryption" />
645         <xs:element ref="MobileNetworkRadioEncryption" />
646         <xs:element ref="MobileNetworkEndToEndEncryption" />
647         <xs:element ref="WTLS" />
648         <xs:element ref="IPSec" />
649         <xs:element ref="Extension" maxOccurs="unbounded" />
650     </xs:choice>
651 </xs:complexType>
652 <xs:complexType name="PrincipalAuthenticationMechanismType">
653     <xs:choice>
654         <xs:element ref="Password" />
655         <xs:element ref="Token" />
656         <xs:element ref="Smartcard" />
657         <xs:element ref="ActivationPin" />
658         <xs:element ref="Extension" maxOccurs="unbounded" />
659     </xs:choice>
660 </xs:complexType>
661 <xs:complexType name="AuthenticationMethodType">
662     <xs:sequence>
663         <xs:element ref="PrincipalAuthenticationMechanism" minOccurs="0" />
664         <xs:element ref="Authenticator" minOccurs="0" />
665         <xs:element ref="AuthenticatorTransportProtocol" minOccurs="0" />
666         <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
667     </xs:sequence>
668 </xs:complexType>
669 <xs:complexType name="AuthenticationContextStatement Type">
670     <xs:sequence>
671         <xs:element ref="Identification" minOccurs="0" />
672         <xs:element ref="TechnicalProtection" minOccurs="0" />
673         <xs:element ref="OperationalProtection" minOccurs="0" />
674         <xs:element ref="AuthenticationMethod" minOccurs="0" />
675         <xs:element ref="GoverningAgreements" minOccurs="0" />
676         <xs:element ref="AuthenticatingAuthority" minOccurs="0" maxOccurs="unbounded" />
677         <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
678     </xs:sequence>
679     <xs:attribute name="ID" type="xs:ID" />
680 </xs:complexType>
681 <xs:complexType name="TechnicalProtectionType">
682     <xs:choice>

```

```

683     <xs:element ref="PrivateKeyProtection" minOccurs="0"/>
684     <xs:element ref="SecretKeyProtection" minOccurs="0"/>
685     <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
686   </xs:choice>
687 </xs:complexType>
688 <xs:complexType name="OperationalProtectionType">
689   <xs:sequence>
690     <xs:element ref="SecurityAudit" minOccurs="0"/>
691     <xs:element ref="DeactivationCallCenter" minOccurs="0"/>
692     <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
693   </xs:sequence>
694 </xs:complexType>
695 <xs:complexType name="AuthenticatorType">
696   <xs:choice>
697     <xs:element ref="PreviousSession"/>
698     <xs:element ref="ResumeSession"/>
699     <xs:element ref="DigSig"/>
700     <xs:element ref="Password"/>
701     <xs:element ref="ZeroKnowledge"/>
702     <xs:element ref="SharedSecretChallengeResponse"/>
703     <xs:element ref="SharedSecretDynamicPlaintext"/>
704     <xs:element ref="IPAddress"/>
705     <xs:element ref="AsymmetricDecryption"/>
706     <xs:element ref="AsymmetricKeyAgreement"/>
707     <xs:element ref="Extension" maxOccurs="unbounded"/>
708   </xs:choice>
709 </xs:complexType>
710 <xs:complexType name="KeyActivationType">
711   <xs:choice>
712     <xs:element ref="ActivationPin"/>
713     <xs:element ref="Extension" maxOccurs="unbounded"/>
714   </xs:choice>
715 </xs:complexType>
716 <xs:complexType name="KeySharingType">
717   <xs:attribute name="sharing" type="xs:boolean" use="required"/>
718 </xs:complexType>
719 <xs:complexType name="PrivateKeyProtectionType">
720   <xs:sequence>
721     <xs:element ref="KeyActivation" minOccurs="0"/>
722     <xs:element ref="KeyStorage" minOccurs="0"/>
723     <xs:element ref="KeySharing" minOccurs="0"/>
724     <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
725   </xs:sequence>
726 </xs:complexType>
727
728 <xs:complexType name="PasswordType">
729   <xs:sequence>
730     <xs:element ref="Length" minOccurs="0"/>
731     <xs:element ref="Alphabet" minOccurs="0"/>
732     <xs:element ref="Generation" minOccurs="0"/>
733     <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
734   </xs:sequence>
735 </xs:complexType>
736
737 <xs:complexType name="ActivationPinType">
738   <xs:sequence>
739     <xs:element ref="Length" minOccurs="0"/>
740     <xs:element ref="Alphabet" minOccurs="0"/>
741     <xs:element ref="Generation" minOccurs="0"/>
742     <xs:element ref="ActivationLimit" minOccurs="0"/>
743     <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
744   </xs:sequence>
745 </xs:complexType>
746
747 <xs:element name="Alphabet" type="AlphabetType"/>
748
749 <xs:complexType name="AlphabetType">

```

```

750     <xs:attribute name="requiredChars" type="xs:string" use="required"/>
751     <xs:attribute name="excludedChars" type="xs:string" use="optional"/>
752     <xs:attribute name="case" type="xs:string" use="optional"/>
753 </xs:complexType>
754
755 <xs:complexType name="TokenType">
756   <xs:sequence>
757     <xs:element ref="TimeSyncToken"/>
758     <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
759   </xs:sequence>
760 </xs:complexType>
761 <xs:complexType name="TimeSyncTokenType">
762   <xs:attribute name="DeviceType" use="required">
763     <xs:simpleType>
764       <xs:restriction base="xs:NMTOKEN">
765         <xs:enumeration value="hardware"/>
766         <xs:enumeration value="software"/>
767       </xs:restriction>
768     </xs:simpleType>
769   </xs:attribute>
770   <xs:attribute name="SeedLength" type="xs:integer" use="required"/>
771   <xs:attribute name="DeviceInHand" use="required">
772     <xs:simpleType>
773       <xs:restriction base="xs:NMTOKEN">
774         <xs:enumeration value="true"/>
775         <xs:enumeration value="false"/>
776       </xs:restriction>
777     </xs:simpleType>
778   </xs:attribute>
779 </xs:complexType>
780 <xs:complexType name="ActivationLimitType">
781   <xs:choice>
782     <xs:element ref="ActivationLimitDuration"/>
783     <xs:element ref="ActivationLimitUsages"/>
784     <xs:element ref="ActivationLimitSession"/>
785   </xs:choice>
786 </xs:complexType>
787
788 <xs:element name="ActivationLimitDuration" type="ActivationLimitDurationType">
789   <xs:annotation>
790     <xs:documentation>
791       This element indicates that the Key Activation Limit is defined
792       as a specific duration of time.
793     </xs:documentation>
794   </xs:annotation>
795 </xs:element>
796
797 <xs:element name="ActivationLimitUsages" type="ActivationLimitUsagesType">
798   <xs:annotation>
799     <xs:documentation>
800       This element indicates that the Key Activation Limit is defined as a number of usages.
801     </xs:documentation>
802   </xs:annotation>
803 </xs:element>
804
805 <xs:element name="ActivationLimitSession" type="ActivationLimitSessionType">
806   <xs:annotation>
807     <xs:documentation>
808       This element indicates that the Key Activation Limit is the session.
809     </xs:documentation>
810   </xs:annotation>
811 </xs:element>
812
813 <xs:complexType name="ActivationLimitDurationType">
814   <xs:attribute name="duration" type="xs:duration" use="required"/>
815 </xs:complexType>
816

```

```
817     <xs:complexType name="ActivationLimitUsagesType">
818       <xs:attribute name="number" type="xs:integer" use="required"/>
819     </xs:complexType>
820
821     <xs:complexType name="ActivationLimitSessionType"/>
822
823     <xs:complexType name="LengthType">
824       <xs:attribute name="min" type="xs:integer" use="required"/>
825       <xs:attribute name="max" type="xs:integer" use="optional"/>
826     </xs:complexType>
827
828     <xs:complexType name="KeyStorageType">
829       <xs:attribute name="medium" use="required">
830         <xs:simpleType>
831           <xs:restriction base="xs:NMTOKEN">
832             <xs:enumeration value="memory"/>
833             <xs:enumeration value="smartcard"/>
834             <xs:enumeration value="token"/>
835             <xs:enumeration value="MobileDevice"/>
836             <xs:enumeration value="MobileAuthCard"/>
837           </xs:restriction>
838         </xs:simpleType>
839       </xs:attribute>
840     </xs:complexType>
841     <xs:complexType name="SecretKeyProtectionType">
842       <xs:sequence>
843         <xs:element ref="KeyActivation" minOccurs="0"/>
844         <xs:element ref="KeyStorage" minOccurs="0"/>
845         <xs:element ref="Extension" maxOccurs="unbounded"/>
846       </xs:sequence>
847     </xs:complexType>
848     <xs:complexType name="SecurityAuditType">
849       <xs:sequence>
850         <xs:element ref="SwitchAudit" minOccurs="0"/>
851         <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
852       </xs:sequence>
853     </xs:complexType>
854 </xs:schema>
855
856
```

857 4.3. Authentication Context Statement Extensibility

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

863 4.4. Authentication Context Statement Processing Rules

864 The processing rules for Authentication Context Statements are listed in [\[LibertyProtSchema\]](#).

865 **5. Authentication Context Classes**

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

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

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

889 **5.1. Advantages of Authentication Context Classes**

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

- 892 • Make it easier for the identity provider and service provider to come to an agreement on what are acceptable
893 authentication contexts by giving them a framework for discussion.
- 894 • Make it easier for service providers to indicate their preferences when requesting a step-up authentication assertion
895 from an identity provider.
- 896 • Simplify for service providers the burden of processing authentication context statements by giving them the option
897 of being satisfied by the associated class.
- 898 • Protect service providers from impact of new authentication technologies.
- 899 • Make it easier for identity providers to publish their authentication capabilities, for example, through WSDL.

900 **5.2. Authentication Context Class Schemas**

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

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

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

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

907 **5.2.1. Internet Protocol**

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

909 **5.2.1.1. Associated Liberty URI**

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

911 **5.2.1.2. Class Schema**

```
912
913 <?xml version="1.0" encoding="UTF-8"?>
914 <xs:schema targetNamespace="urn:liberty:ac:2003-08"
915   xmlns:xs="http://www.w3.org/2001/XMLSchema"
916   xmlns="urn:liberty:ac:2003-08"
917   version="1.2-08" finalDefault="extension">
918   <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
919   <xs:annotation>
920     <xs:documentation>
921       http://www.projectliberty.org/schemas/authctx/classes/InternetProtocol</x
922 s:documentation>
923   </xs:annotation>
924   <xs:complexType name="InternetProtocolAuthenticatorType">
925     <xs:complexContent>
926       <xs:restriction base="AuthenticatorType">
927         <xs:choice>
928           <xs:element ref="IPAddress"/>
929         </xs:choice>
930       </xs:restriction>
931     </xs:complexContent>
932   </xs:complexType>
933 </xs:schema>
934
935
```

936 **5.2.2. InternetProtocolPassword**

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

939 **5.2.2.1. Associated Liberty URI**

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

941 **5.2.2.2. Class Schema**

```
942
943 <?xml version="1.0" encoding="UTF-8"?>
944 <xs:schema targetNamespace="urn:liberty:ac:2003-08"
945   xmlns="urn:liberty:ac:2003-08">
```



```

946   xmlns:xs="http://www.w3.org/2001/XMLSchema"
947   version="1.2-08" finalDefault="extension">
948   <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
949   <xs:annotation>
950     <xs:documentation>
951       http://www.projectliberty.org/schemas/authctx/classes/InternetProtocolPas
952 sword</xs:documentation>
953   </xs:annotation>
954
955   <xs:complexType name="InternetProtocolPasswordType">
956     <xs:complexContent>
957       <xs:restriction base="PasswordType">
958         <xs:sequence>
959           <xs:element ref="Length"/>
960           <xs:element ref="Generation" minOccurs="0"/>
961           <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
962         </xs:sequence>
963       </xs:restriction>
964     </xs:complexContent>
965   </xs:complexType>
966   <xs:complexType name="InternetProtocolPasswordLengthType">
967     <xs:complexContent>
968       <xs:restriction base="LengthType">
969         <xs:attribute name="min" use="required">
970           <xs:simpleType>
971             <xs:restriction base="xs:integer">
972               <xs:minInclusive value="3"/>
973             </xs:restriction>
974           </xs:simpleType>
975         </xs:attribute>
976         <xs:attribute name="max" type="xs:integer" use="optional"/>
977       </xs:restriction>
978     </xs:complexContent>
979   </xs:complexType>
980   <xs:complexType name="InternetProtocolPasswordAuthenticatorType">
981     <xs:complexContent>
982       <xs:restriction base="AuthenticatorType">
983         <xs:sequence>
984           <xs:element ref="IPAddress"/>
985           <xs:element ref="Password"/>
986           <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
987         </xs:sequence>
988       </xs:restriction>
989     </xs:complexContent>
990   </xs:complexType>
991 </xs:schema>
992
993

```

994 5.2.3. MobileOneFactorUnregistered

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

998 5.2.3.1. Associated Liberty URI

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

1000 5.2.3.2. Class Schema

```

1001
1002 <?xml version="1.0" encoding="UTF-8"?>
1003 <xs:schema targetNamespace="urn:liberty:ac:2003-08"
1004   xmlns="urn:liberty:ac:2003-08"

```

```

1005     xmlns:xs="http://www.w3.org/2001/XMLSchema"
1006     finalDefault="extension" version="1.2-08">
1007     <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
1008     <xs:annotation>
1009       <xs:documentation>http://www.projectliberty.org/schemas/authctx/classes/Mo
1010 bileOneFactorUnregistered</xs:documentation>
1011     </xs:annotation>
1012     <xs:complexType name="MobileOneFactorUnregisteredAuthenticatorType">
1013       <xs:complexContent>
1014         <xs:restriction base="AuthenticatorType">
1015           <xs:choice>
1016             <xs:element ref="DigSig"/>
1017             <xs:element ref="ZeroKnowledge"/>
1018             <xs:element ref="SharedSecretChallengeResponse"/>
1019             <xs:element ref="AsymmetricDecryption"/>
1020             <xs:element ref="AsymmetricKeyAgreement"/>
1021             <xs:element ref="SharedSecretDynamicPlaintext"/>
1022           </xs:choice>
1023         </xs:restriction>
1024       </xs:complexContent>
1025     </xs:complexType>
1026     <xs:complexType name="MobileOneFactorUnregisteredAuthenticatorTransportProtocolType">
1027       <xs:complexContent>
1028         <xs:restriction base="AuthenticatorTransportProtocolType">
1029           <xs:choice>
1030             <xs:element ref="MobileNetworkNoEncryption"/>
1031             <xs:element ref="MobileNetworkRadioEncryption"/>
1032             <xs:element ref="MobileNetworkEndToEndEncryption"/>
1033             <xs:element ref="WTLS"/>
1034           </xs:choice>
1035         </xs:restriction>
1036       </xs:complexContent>
1037     </xs:complexType>
1038     <xs:complexType name="MobileOneFactorUnregisteredOperationalProtectionType">
1039       <xs:complexContent>
1040         <xs:restriction base="OperationalProtectionType">
1041           <xs:sequence>
1042             <xs:element ref="SecurityAudit"/>
1043             <xs:element ref="DeactivationCallCenter"/>
1044             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1045           </xs:sequence>
1046         </xs:restriction>
1047       </xs:complexContent>
1048     </xs:complexType>
1049     <xs:complexType name="MobileOneFactorUnregisteredTechnicalProtectionType">
1050       <xs:complexContent>
1051         <xs:restriction base="TechnicalProtectionType">
1052           <xs:choice>
1053             <xs:element ref="PrivateKeyProtection"/>
1054             <xs:element ref="SecretKeyProtection"/>
1055           </xs:choice>
1056         </xs:restriction>
1057       </xs:complexContent>
1058     </xs:complexType>
1059     <xs:complexType name="MobileOneFactorUnregisteredPrivateKeyProtectionType">
1060       <xs:complexContent>
1061         <xs:restriction base="PrivateKeyProtectionType">
1062           <xs:sequence>
1063             <xs:element ref="KeyStorage"/>
1064             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1065           </xs:sequence>
1066         </xs:restriction>
1067       </xs:complexContent>
1068     </xs:complexType>
1069     <xs:complexType name="MobileOneFactorUnregisteredSecretKeyProtectionType">
1070       <xs:complexContent>
1071         <xs:restriction base="SecretKeyProtectionType">

```

```

1072         <xs:sequence>
1073             <xs:element ref="KeyStorage" />
1074             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
1075         </xs:sequence>
1076     </xs:restriction>
1077 </xs:complexContent>
1078 </xs:complexType>
1079 <xs:complexType name="MobileOneFactorUnregisteredKeyStorageType">
1080     <xs:complexContent>
1081         <xs:restriction base="KeyStorageType">
1082             <xs:attribute name="medium" use="required">
1083                 <xs:simpleType>
1084                     <xs:restriction base="xs:NMTOKEN">
1085                         <xs:enumeration value="MobileDevice" />
1086                         <xs:enumeration value="MobileAuthCard" />
1087                         <xs:enumeration value="smartcard" />
1088                     </xs:restriction>
1089                 </xs:simpleType>
1090             </xs:attribute>
1091         </xs:restriction>
1092     </xs:complexContent>
1093 </xs:complexType>
1094 <xs:complexType name="MobileOneFactorUnregisteredSecurityAuditType">
1095     <xs:complexContent>
1096         <xs:restriction base="SecurityAuditType">
1097             <xs:sequence>
1098                 <xs:element ref="SwitchAudit" />
1099                 <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
1100             </xs:sequence>
1101         </xs:restriction>
1102     </xs:complexContent>
1103 </xs:complexType>
1104 <xs:complexType name="MobileOneFactorUnregisteredIdentificationType">
1105     <xs:complexContent>
1106         <xs:restriction base="IdentificationType">
1107             <xs:attribute name="nym">
1108                 <xs:simpleType>
1109                     <xs:restriction base="xs:NMTOKEN">
1110                         <xs:enumeration value="anonymity" />
1111                         <xs:enumeration value="pseudonymity" />
1112                     </xs:restriction>
1113                 </xs:simpleType>
1114             </xs:attribute>
1115         </xs:restriction>
1116     </xs:complexContent>
1117 </xs:complexType>
1118 </xs:schema>
1119
1120
    
```

1121 5.2.4. MobileTwoFactorUnregistered

1122 Reflects no mobile customer registration procedures and a two-factor based authentication, such as secure device and
 1123 user PIN. This context class is useful when a service other than the mobile operator wants to link their customer ID
 1124 to a mobile supplied two-factor authentication service by capturing mobile phone data at enrollment.

1125 5.2.4.1. Associated Liberty URI

1126 <http://www.projectliberty.org/schemas/authctx/classes/MobileTwoFactorUnregistered>

1127 5.2.4.2. Class Schema

```

1128
1129 <?xml version="1.0" encoding="UTF-8"?>
1130 <xs:schema targetNamespace="urn:liberty:ac:2003-08"
    
```

```
1131 xmlns:xs="http://www.w3.org/2001/XMLSchema"
1132 xmlns="urn:liberty:ac:2003-08"
1133 version="1.2-08"
1134 finalDefault="extension">
1135 <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
1136 <xs:annotation>
1137   <xs:documentation>
1138     http://www.projectliberty.org/schemas/authctx/classes/MobileTwoFactorUnregistered
1139   </xs:documentation>
1140 </xs:annotation>
1141
1142 <xs:complexType name="MobileTwoFactorUnregisteredAuthenticatorType">
1143   <xs:complexContent>
1144     <xs:restriction base="AuthenticatorType">
1145       <xs:choice>
1146         <xs:element ref="DigSig"/>
1147         <xs:element ref="ZeroKnowledge"/>
1148         <xs:element ref="SharedSecretChallengeResponse"/>
1149         <xs:element ref="AsymmetricDecryption"/>
1150         <xs:element ref="AsymmetricKeyAgreement"/>
1151         <xs:element ref="SharedSecretDynamicPlaintext"/>
1152         <xs:sequence>
1153           <xs:element ref="Password" minOccurs="1"/>
1154           <xs:choice>
1155             <xs:element ref="SharedSecretDynamicPlaintext"/>
1156             <xs:element ref="SharedSecretChallengeResponse"/>
1157           </xs:choice>
1158           <xs:element ref="Extension" maxOccurs="unbounded"/>
1159         </xs:sequence>
1160       </xs:choice>
1161     </xs:restriction>
1162   </xs:complexContent>
1163 </xs:complexType>
1164 <xs:complexType name="MobileTwoFactorUnregisteredAuthenticatorTransportProtocolType">
1165   <xs:complexContent>
1166     <xs:restriction base="AuthenticatorTransportProtocolType">
1167       <xs:choice>
1168         <xs:element ref="MobileNetworkNoEncryption"/>
1169         <xs:element ref="MobileNetworkRadioEncryption"/>
1170         <xs:element ref="MobileNetworkEndToEndEncryption"/>
1171         <xs:element ref="WTLS"/>
1172       </xs:choice>
1173     </xs:restriction>
1174   </xs:complexContent>
1175 </xs:complexType>
1176 <xs:complexType name="MobileTwoFactorUnregisteredOperationalProtectionType">
1177   <xs:complexContent>
1178     <xs:restriction base="OperationalProtectionType">
1179       <xs:sequence>
1180         <xs:element ref="SecurityAudit"/>
1181         <xs:element ref="DeactivationCallCenter"/>
1182         <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1183       </xs:sequence>
1184     </xs:restriction>
1185   </xs:complexContent>
1186 </xs:complexType>
1187 <xs:complexType name="MobileTwoFactorUnregisteredTechnicalProtectionType">
1188   <xs:complexContent>
1189     <xs:restriction base="TechnicalProtectionType">
1190       <xs:choice>
1191         <xs:element ref="PrivateKeyProtection"/>
1192         <xs:element ref="SecretKeyProtection"/>
1193       </xs:choice>
1194     </xs:restriction>
1195   </xs:complexContent>
1196 </xs:complexType>
1197
```

```

1198 <xs:complexType name="MobileTwoFactorUnregisteredPrivateKeyProtectionType">
1199   <xs:complexContent>
1200     <xs:restriction base="PrivateKeyProtectionType">
1201       <xs:sequence>
1202         <xs:element ref="KeyActivation" minOccurs="1" maxOccurs="1"/>
1203         <xs:element ref="KeyStorage" minOccurs="0"/>
1204         <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1205       </xs:sequence>
1206     </xs:restriction>
1207   </xs:complexContent>
1208 </xs:complexType>
1209
1210 <xs:complexType name="MobileTwoFactorUnregisteredSecretKeyProtectionType">
1211   <xs:complexContent>
1212     <xs:restriction base="SecretKeyProtectionType">
1213       <xs:sequence>
1214         <xs:element ref="KeyActivation"/>
1215         <xs:element ref="KeyStorage"/>
1216         <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1217       </xs:sequence>
1218     </xs:restriction>
1219   </xs:complexContent>
1220 </xs:complexType>
1221
1222 <xs:complexType name="MobileTwoFactorUnregisteredKeyActivationType">
1223   <xs:complexContent>
1224     <xs:restriction base="KeyActivationType">
1225       <xs:sequence>
1226         <xs:element ref="ActivationPin"/>
1227         <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1228       </xs:sequence>
1229     </xs:restriction>
1230   </xs:complexContent>
1231 </xs:complexType>
1232
1233 <xs:complexType name="MobileTwoFactorUnregisteredKeyStorageType">
1234   <xs:complexContent>
1235     <xs:restriction base="KeyStorageType">
1236       <xs:attribute name="medium" use="required">
1237         <xs:simpleType>
1238           <xs:restriction base="xs:NMTOKEN">
1239             <xs:enumeration value="MobileDevice"/>
1240             <xs:enumeration value="MobileAuthCard"/>
1241             <xs:enumeration value="smartcard"/>
1242           </xs:restriction>
1243         </xs:simpleType>
1244       </xs:attribute>
1245     </xs:restriction>
1246   </xs:complexContent>
1247 </xs:complexType>
1248
1249 <xs:complexType name="MobileTwoFactorUnregisteredSecurityAuditType">
1250   <xs:complexContent>
1251     <xs:restriction base="SecurityAuditType">
1252       <xs:sequence>
1253         <xs:element ref="SwitchAudit"/>
1254         <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1255       </xs:sequence>
1256     </xs:restriction>
1257   </xs:complexContent>
1258 </xs:complexType>
1259
1260 <xs:complexType name="MobileTwoFactorUnregisteredIdentificationType">
1261   <xs:complexContent>
1262     <xs:restriction base="IdentificationType">
1263       <xs:attribute name="nym">
1264         <xs:simpleType>

```

```

1265         <xs:restriction base="xs:NMTOKEN">
1266             <xs:enumeration value="anonymity" />
1267             <xs:enumeration value="pseudonymity" />
1268         </xs:restriction>
1269     </xs:simpleType>
1270 </xs:attribute>
1271 </xs:restriction>
1272 </xs:complexContent>
1273 </xs:complexType>
1274 </xs:schema>
1275
1276
1277
  
```

1278 5.2.5. MobileOneFactorContract

1279 Reflects mobile contract customer registration procedures and a single factor authentication. For example, a digital
 1280 signing device with tamper resistant memory for key storage, such as the mobile MSISDN, but no required PIN or
 1281 biometric for real-time user authentication.

1282 5.2.5.1. Associated Liberty URI

1283 <http://www.projectliberty.org/schemas/authctx/classes/MobileOneFactorContract>

1284 5.2.5.2. Class Schema

```

1285
1286 <?xml version="1.0" encoding="UTF-8"?>
1287 <xs:schema targetNamespace="urn:liberty:ac:2003-08"
1288     xmlns:xs="http://www.w3.org/2001/XMLSchema"
1289     xmlns="urn:liberty:ac:2003-08"
1290     version="1.2-08" finalDefault="extension">
1291     <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd" />
1292     <xs:annotation>
1293 <xs:documentation>http://www.projectliberty.org/schemas/authctx/classes/MobileOneFa
1294 ctorContract</xs:documentation></xs:annotation>
1295
1296     <xs:complexType name="MobileOneFactorContractAuthenticatorType">
1297         <xs:complexContent>
1298             <xs:restriction base="AuthenticatorType">
1299                 <xs:choice maxOccurs="1">
1300                     <xs:element ref="DigSig" />
1301                     <xs:element ref="ZeroKnowledge" />
1302                     <xs:element ref="SharedSecretChallengeResponse" />
1303                     <xs:element ref="AsymmetricDecryption" />
1304                     <xs:element ref="AsymmetricKeyAgreement" />
1305                     <xs:element ref="SharedSecretDynamicPlaintext" />
1306                 </xs:choice>
1307             </xs:restriction>
1308         </xs:complexContent>
1309     </xs:complexType>
1310     <xs:complexType name="MobileOneFactorContractAuthenticatorTransportProtocolType">
1311         <xs:complexContent>
1312             <xs:restriction base="AuthenticatorTransportProtocolType">
1313                 <xs:choice>
1314                     <xs:element ref="MobileNetworkNoEncryption" />
1315                     <xs:element ref="MobileNetworkRadioEncryption" />
1316                     <xs:element ref="MobileNetworkEndToEndEncryption" />
1317                     <xs:element ref="WTLS" />
1318                 </xs:choice>
1319             </xs:restriction>
1320         </xs:complexContent>
1321     </xs:complexType>
1322     <xs:complexType name="MobileOneFactorContractOperationalProtectionType">
1323         <xs:complexContent>
  
```

```
1324     <xs:restriction base="OperationalProtectionType">
1325         <xs:sequence>
1326             <xs:element ref="SecurityAudit"/>
1327             <xs:element ref="DeactivationCallCenter"/>
1328             <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1329         </xs:sequence>
1330     </xs:restriction>
1331 </xs:complexContent>
1332 </xs:complexType>
1333 <xs:complexType name="MobileOneFactorContractTechnicalProtectionType">
1334     <xs:complexContent>
1335         <xs:restriction base="TechnicalProtectionType">
1336             <xs:choice>
1337                 <xs:element ref="PrivateKeyProtection"/>
1338                 <xs:element ref="SecretKeyProtection"/>
1339             </xs:choice>
1340         </xs:restriction>
1341     </xs:complexContent>
1342 </xs:complexType>
1343
1344 <xs:complexType name="MobileOneFactorContractPrivateKeyProtectionType">
1345     <xs:complexContent>
1346         <xs:restriction base="PrivateKeyProtectionType">
1347             <xs:sequence maxOccurs="1">
1348                 <xs:element ref="KeyStorage"/>
1349                 <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1350             </xs:sequence>
1351         </xs:restriction>
1352     </xs:complexContent>
1353 </xs:complexType>
1354
1355 <xs:complexType name="MobileOneFactorContractSecretKeyProtectionType">
1356     <xs:complexContent>
1357         <xs:restriction base="SecretKeyProtectionType">
1358             <xs:sequence>
1359                 <xs:element ref="KeyStorage"/>
1360                 <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1361             </xs:sequence>
1362         </xs:restriction>
1363     </xs:complexContent>
1364 </xs:complexType>
1365
1366 <xs:complexType name="MobileOneFactorContractKeyStorageType">
1367     <xs:complexContent>
1368         <xs:restriction base="KeyStorageType">
1369             <xs:attribute name="medium" use="required">
1370                 <xs:simpleType>
1371                     <xs:restriction base="xs:NMTOKEN">
1372                         <xs:enumeration value="MobileDevice"/>
1373                         <xs:enumeration value="MobileAuthCard"/>
1374                         <xs:enumeration value="smartcard"/>
1375                     </xs:restriction>
1376                 </xs:simpleType>
1377             </xs:attribute>
1378         </xs:restriction>
1379     </xs:complexContent>
1380 </xs:complexType>
1381
1382 <xs:complexType name="MobileOneFactorContractSecurityAuditType">
1383     <xs:complexContent>
1384         <xs:restriction base="SecurityAuditType">
1385             <xs:sequence>
1386                 <xs:element ref="SwitchAudit"/>
1387                 <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1388             </xs:sequence>
1389         </xs:restriction>
1390     </xs:complexContent>
```

```

1391     </xs:complexType>
1392
1393     <xs:complexType name="MobileOneFactorContractIdentificationType">
1394         <xs:complexContent>
1395             <xs:restriction base="IdentificationType">
1396                 <xs:sequence>
1397                     <xs:element ref="PhysicalVerification"/>
1398                     <xs:element ref="WrittenConsent"/>
1399                     <xs:element ref="GoverningAgreements"/>
1400                     <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1401                 </xs:sequence>
1402                 <xs:attribute name="nym">
1403                     <xs:simpleType>
1404                         <xs:restriction base="xs:NMTOKEN">
1405                             <xs:enumeration value="anonymity"/>
1406                             <xs:enumeration value="verinymity"/>
1407                             <xs:enumeration value="pseudonymity"/>
1408                         </xs:restriction>
1409                     </xs:simpleType>
1410                 </xs:attribute>
1411             </xs:restriction>
1412         </xs:complexContent>
1413     </xs:complexType>
1414
1415 </xs:schema>
1416
1417
    
```

1418 5.2.6. MobileTwoFactorContract

1419 Reflects mobile contract customer registration procedures and a two-factor based authentication. For example, a
 1420 digital signing device with tamper resistant memory for key storage, such as a GSM SIM, that requires explicit proof
 1421 of user identity and intent, such as a PIN or biometric.

1422 5.2.6.1. Associated Liberty URI

1423 <http://www.projectliberty.org/schemas/authctx/classes/MobileTwoFactorContract>

1424 5.2.6.2. Class Schema

```

1425
1426 <?xml version="1.0" encoding="UTF-8"?>
1427 <xs:schema targetNamespace="urn:liberty:ac:2003-08"
1428     xmlns:xs="http://www.w3.org/2001/XMLSchema"
1429     xmlns="urn:liberty:ac:2003-08"
1430     version="1.2-08"
1431     finalDefault="extension">
1432     <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
1433     <xs:annotation><xs:documentation>http://www.projectliberty.org/schemas/authctx/classes/MobileT
1434 woFactorContract</xs:documentation>
1435 </xs:annotation>
1436
1437     <xs:complexType name="MobileTwoFactorContractAuthenticatorType">
1438 <xs:complexContent>
1439 <xs:restriction base="AuthenticatorType">
1440     <xs:choice>
1441         <xs:element ref="DigSig"/>
1442         <xs:element ref="ZeroKnowledge"/>
1443         <xs:element ref="SharedSecretChallengeResponse"/>
1444         <xs:element ref="AsymmetricDecryption"/>
1445         <xs:element ref="AsymmetricKeyAgreement"/>
1446         <xs:element ref="SharedSecretDynamicPlaintext"/>
1447     </xs:sequence>
1448     <xs:element ref="Password" minOccurs="1"/>
1449 </xs:choice>
    
```



```
1450         <xs:element ref="SharedSecretDynamicPlaintext" />
1451         <xs:element ref="SharedSecretChallengeResponse" />
1452     </xs:choice>
1453     <xs:element ref="Extension" maxOccurs="unbounded" />
1454 </xs:sequence>
1455 </xs:choice>
1456 </xs:restriction>
1457 </xs:complexContent>
1458 </xs:complexType>
1459 <xs:complexType name="MobileTwoFactorContractAuthenticatorTransportProtocolType">
1460     <xs:complexContent>
1461         <xs:restriction base="AuthenticatorTransportProtocolType">
1462             <xs:choice>
1463                 <xs:element ref="MobileNetworkNoEncryption" />
1464                 <xs:element ref="MobileNetworkRadioEncryption" />
1465                 <xs:element ref="MobileNetworkEndToEndEncryption" />
1466                 <xs:element ref="WTLS" />
1467             </xs:choice>
1468         </xs:restriction>
1469     </xs:complexContent>
1470 </xs:complexType>
1471 <xs:complexType name="MobileTwoFactorContractOperationalProtectionType">
1472     <xs:complexContent>
1473         <xs:restriction base="OperationalProtectionType">
1474             <xs:sequence>
1475                 <xs:element ref="SecurityAudit" />
1476                 <xs:element ref="DeactivationCallCenter" />
1477                 <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
1478             </xs:sequence>
1479         </xs:restriction>
1480     </xs:complexContent>
1481 </xs:complexType>
1482 <xs:complexType name="MobileTwoFactorContractTechnicalProtectionType">
1483     <xs:complexContent>
1484         <xs:restriction base="TechnicalProtectionType">
1485             <xs:choice>
1486                 <xs:element ref="PrivateKeyProtection" />
1487                 <xs:element ref="SecretKeyProtection" />
1488             </xs:choice>
1489         </xs:restriction>
1490     </xs:complexContent>
1491 </xs:complexType>
1492
1493 <xs:complexType name="MobileTwoFactorContractPrivateKeyProtectionType">
1494     <xs:complexContent>
1495         <xs:restriction base="PrivateKeyProtectionType">
1496             <xs:sequence>
1497                 <xs:element ref="KeyActivation" minOccurs="1" maxOccurs="1" />
1498                 <xs:element ref="KeyStorage" minOccurs="0" />
1499                 <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
1500             </xs:sequence>
1501         </xs:restriction>
1502     </xs:complexContent>
1503 </xs:complexType>
1504
1505 <xs:complexType name="MobileTwoFactorContractSecretKeyProtectionType">
1506     <xs:complexContent>
1507         <xs:restriction base="SecretKeyProtectionType">
1508             <xs:sequence>
1509                 <xs:element ref="KeyActivation" />
1510                 <xs:element ref="KeyStorage" />
1511                 <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
1512             </xs:sequence>
1513         </xs:restriction>
1514     </xs:complexContent>
1515 </xs:complexType>
1516
```

```

1517 <xs:complexType name="MobileTwoFactorContractKeyActivationType">
1518 <xs:complexContent>
1519 <xs:restriction base="KeyActivationType">
1520 <xs:sequence>
1521 <xs:element ref="ActivationPin"/>
1522 <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1523 </xs:sequence>
1524 </xs:restriction>
1525 </xs:complexContent>
1526 </xs:complexType>
1527
1528 <xs:complexType name="MobileTwoFactorContractKeyStorageType">
1529 <xs:complexContent>
1530 <xs:restriction base="KeyStorageType">
1531 <xs:attribute name="medium" use="required">
1532 <xs:simpleType>
1533 <xs:restriction base="xs:NMTOKEN">
1534 <xs:enumeration value="MobileDevice"/>
1535 <xs:enumeration value="MobileAuthCard"/>
1536 <xs:enumeration value="smartcard"/>
1537 </xs:restriction>
1538 </xs:simpleType>
1539 </xs:attribute>
1540 </xs:restriction>
1541 </xs:complexContent>
1542 </xs:complexType>
1543
1544 <xs:complexType name="MobileTwoFactorContractSecurityAuditType">
1545 <xs:complexContent>
1546 <xs:restriction base="SecurityAuditType">
1547 <xs:sequence>
1548 <xs:element ref="SwitchAudit"/>
1549 <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1550 </xs:sequence>
1551 </xs:restriction>
1552 </xs:complexContent>
1553 </xs:complexType>
1554
1555 <xs:complexType name="MobileTwoFactorContractIdentificationType">
1556 <xs:complexContent>
1557 <xs:restriction base="IdentificationType">
1558 <xs:sequence>
1559 <xs:element ref="PhysicalVerification"/>
1560 <xs:element ref="WrittenConsent"/>
1561 <xs:element ref="GoverningAgreements"/>
1562 <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1563 </xs:sequence>
1564 <xs:attribute name="nym">
1565 <xs:simpleType>
1566 <xs:restriction base="xs:NMTOKEN">
1567 <xs:enumeration value="anonymity"/>
1568 <xs:enumeration value="veronymity"/>
1569 <xs:enumeration value="pseudonymity"/>
1570 </xs:restriction>
1571 </xs:simpleType>
1572 </xs:attribute>
1573 </xs:restriction>
1574 </xs:complexContent>
1575 </xs:complexType>
1576 </xs:schema>
1577
1578

```

1579 5.2.7. Password

1580 The Password class is identified when a Principal authenticates to an identity provider through the presentation of a
 1581 password over an unprotected HTTP session.

1582 5.2.7.1. Associated Liberty URI

1583 <http://www.projectliberty.org/schemas/authctx/classes/Password>

1584 5.2.7.2. Class Schema

```
1585
1586 <?xml version="1.0" encoding="UTF-8"?>
1587 <xs:schema targetNamespace="urn:liberty:ac:2003-08"
1588   xmlns:xs="http://www.w3.org/2001/XMLSchema"
1589   xmlns="urn:liberty:ac:2003-08"
1590   version="1.2-08"
1591   finalDefault="extension">
1592   <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
1593   <xs:annotation>
1594     <xs:documentation>
1595       http://www.projectliberty.org/schemas/authctx/classes/Password</xs:documentation>
1596     </xs:annotation>
1597   <xs:complexType name="PasswordAuthenticatorType">
1598     <xs:complexContent>
1599       <xs:restriction base="AuthenticatorType">
1600         <xs:choice>
1601           <xs:element ref="Password"/>
1602         </xs:choice>
1603       </xs:restriction>
1604     </xs:complexContent>
1605   </xs:complexType>
1606
1607   <xs:complexType name="PasswordPasswordType">
1608     <xs:complexContent>
1609       <xs:restriction base="PasswordType">
1610         <xs:sequence>
1611           <xs:element ref="Length" minOccurs="1"/>
1612           <xs:element ref="Generation" minOccurs="0"/>
1613           <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1614         </xs:sequence>
1615       </xs:restriction>
1616     </xs:complexContent>
1617   </xs:complexType>
1618
1619   <xs:complexType name="PasswordLengthType">
1620     <xs:complexContent>
1621       <xs:restriction base="LengthType">
1622         <xs:attribute name="min" use="required">
1623           <xs:simpleType>
1624             <xs:restriction base="xs:integer">
1625               <xs:minInclusive value="3"/>
1626             </xs:restriction>
1627           </xs:simpleType>
1628         </xs:attribute>
1629         <xs:attribute name="max" type="xs:integer" use="optional"/>
1630       </xs:restriction>
1631     </xs:complexContent>
1632   </xs:complexType>
1633 </xs:schema>
1634
1635
1636
```

1637 5.2.8. PasswordProtectedTransport

1638 The PasswordProtectedTransport class is identified when a Principal authenticates to an identity provider through the
1639 presentation of a password over a protected session.

1640 5.2.8.1. Associated Liberty URI

1641 <http://www.projectliberty.org/schemas/authctx/classes/PasswordProtectedTransport>

1642 5.2.8.2. Class Schema

```
1643
1644 <?xml version="1.0" encoding="UTF-8"?>
1645 <xs:schema targetNamespace="urn:liberty:ac:2003-08"
1646   xmlns="urn:liberty:ac:2003-08"
1647   xmlns:xs="http://www.w3.org/2001/XMLSchema"
1648   version="1.2-08"
1649   finalDefault="extension">
1650   <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
1651   <xs:annotation>
1652     <xs:documentation>
1653       http://www.projectliberty.org/schemas/authctx/classes/PasswordProtectedTransport</xs:documentation>
1654     </xs:annotation>
1655   <xs:complexType name="PasswordProtectedTransportAuthenticatorType">
1656     <xs:complexContent>
1657       <xs:restriction base="AuthenticatorType">
1658         <xs:choice>
1659           <xs:element ref="Password"/>
1660         </xs:choice>
1661       </xs:restriction>
1662     </xs:complexContent>
1663   </xs:complexType>
1664
1665   <xs:complexType name="PasswordProtectedTransportPasswordType">
1666     <xs:complexContent>
1667       <xs:restriction base="PasswordType">
1668         <xs:sequence>
1669           <xs:element ref="Length"/>
1670           <xs:element ref="Generation" minOccurs="0"/>
1671           <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1672         </xs:sequence>
1673       </xs:restriction>
1674     </xs:complexContent>
1675   </xs:complexType>
1676
1677   <xs:complexType name="PasswordProtectedTransportLengthType">
1678     <xs:complexContent>
1679       <xs:restriction base="LengthType">
1680         <xs:attribute name="min" use="required">
1681           <xs:simpleType>
1682             <xs:restriction base="xs:integer">
1683               <xs:minInclusive value="3"/>
1684             </xs:restriction>
1685           </xs:simpleType>
1686         </xs:attribute>
1687         <xs:attribute name="max" type="xs:integer" use="optional"/>
1688       </xs:restriction>
1689     </xs:complexContent>
1690   </xs:complexType>
1691
1692   <xs:complexType name="PasswordProtectedTransportAuthenticatorTransportProtocolType">
1693     <xs:complexContent>
1694       <xs:restriction base="AuthenticatorTransportProtocolType">
1695         <xs:choice>
1696           <xs:element ref="SSL"/>
1697         </xs:choice>
1698       </xs:restriction>
1699     </xs:complexContent>
1700   </xs:complexType>
1701 </xs:schema>
1702
1703
```

1704 **5.2.9. PreviousSession**

1705 The PreviousSession class is identified when a Principal had authenticated to an identity provider at some point in the
1706 past using any authentication context supported by that identity provider. Consequently, a subsequent authentication
1707 event that the identity provider will assert to the service provider may be significantly separated in time from the
1708 Principals current resource access request.

1709 The context for the previously authenticated session is explicitly not included in this context class because the user
1710 has not authenticated during this session, and so the mechanism that the user employed to authenticate in a previous
1711 session should not be used as part of a decision on whether to now allow access to a resource.

1712 **5.2.9.1. Associated Liberty URI**

1713 <http://www.projectliberty.org/schemas/authctx/classes/PreviousSession>

1714 **5.2.9.2. Class Schema**

```
1715
1716 <?xml version="1.0" encoding="UTF-8"?>
1717 <xs:schema targetNamespace="urn:liberty:ac:2003-08"
1718   xmlns="urn:liberty:ac:2003-08"
1719   xmlns:xs="http://www.w3.org/2001/XMLSchema"
1720   version="1.2-08"
1721   finalDefault="extension">
1722   <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
1723   <xs:annotation>
1724     <xs:documentation>
1725       http://www.projectliberty.org/schemas/authctx/classes/PreviousSession</xs:documentation>
1726   </xs:annotation>
1727   <xs:complexType name="PreviousSessionAuthenticatorType">
1728     <xs:complexContent>
1729       <xs:restriction base="AuthenticatorType">
1730         <xs:choice>
1731           <xs:element ref="PreviousSession"/>
1732         </xs:choice>
1733       </xs:restriction>
1734     </xs:complexContent>
1735   </xs:complexType>
1736 </xs:schema>
1737
1738
1739
```

1740 **5.2.10. Smartcard**

1741 The Smartcard class is identified when a Principal authenticates to an identity provider using a smartcard.

1742 **5.2.10.1. Associated Liberty URI**

1743 <http://www.projectliberty.org/schemas/authctx/classes/Smartcard>

1744 **5.2.10.2. Class Schema**

```
1745
1746 <?xml version="1.0" encoding="UTF-8"?>
1747 <xs:schema targetNamespace="urn:liberty:ac:2003-08"
1748   xmlns="urn:liberty:ac:2003-08"
1749   xmlns:xs="http://www.w3.org/2001/XMLSchema"
1750   version="1.2-08"
1751   finalDefault="extension">
1752   <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
1753   <xs:annotation>
1754     <xs:documentation> http://www.projectliberty.org/schemas/authctx/classes/Smartcard
```

```

1755     </xs:documentation>
1756     </xs:annotation>
1757     <xs:complexType name="SmartCardPrincipalAuthenticationMechanismType">
1758         <xs:complexContent>
1759             <xs:restriction base="PrincipalAuthenticationMechanismType">
1760                 <xs:choice>
1761                     <xs:element ref="Smartcard"/>
1762                 </xs:choice>
1763             </xs:restriction>
1764         </xs:complexContent>
1765     </xs:complexType>
1766 </xs:schema>
1767
1768

```

1769 5.2.11. SmartcardPKI

1770 The SmartcardPKI class is identified when a Principal authenticates to an identity provider through a two-factor
1771 authentication mechanism using a smartcard with enclosed private key and a PIN.

1772 5.2.11.1. Associated Liberty URI

1773 <http://www.projectliberty.org/schemas/authctx/classes/SmartcardPKI>

1774 5.2.11.2. Class Schema

```

1775
1776 <?xml version="1.0" encoding="UTF-8"?>
1777 <xs:schema targetNamespace="urn:liberty:ac:2003-08"
1778     xmlns="urn:liberty:ac:2003-08"
1779     xmlns:xs="http://www.w3.org/2001/XMLSchema"
1780     version="1.2-08"
1781     finalDefault="extension">
1782     <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
1783     <xs:annotation>
1784         <xs:documentation>
1785             http://www.projectliberty.org/schemas/authctx/classes/SmartcardPKI</xs:documentation>
1786     </xs:annotation>
1787
1788     <xs:complexType name="SmartCardPKIPrincipalAuthenticationMechanismType">
1789         <xs:complexContent>
1790             <xs:restriction base="PrincipalAuthenticationMechanismType">
1791                 <xs:sequence>
1792                     <xs:element ref="ActivationPin"/>
1793                     <xs:element ref="Smartcard"/>
1794                     <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1795                 </xs:sequence>
1796             </xs:restriction>
1797         </xs:complexContent>
1798     </xs:complexType>
1799
1800     <xs:complexType name="SmartCardPKIAuthenticatorType">
1801         <xs:complexContent>
1802             <xs:restriction base="AuthenticatorType">
1803                 <xs:choice>
1804                     <xs:element ref="AsymmetricDecryption"/>
1805                     <xs:element ref="AsymmetricKeyAgreement"/>
1806                     <xs:element ref="DigSig"/>
1807                 </xs:choice>
1808             </xs:restriction>
1809         </xs:complexContent>
1810     </xs:complexType>
1811
1812     <xs:complexType name="SmartCardPKIKeyActivationType">
1813         <xs:complexContent>

```

```

1814         <xs:restriction base="KeyActivationType">
1815             <xs:choice>
1816                 <xs:element ref="ActivationPin" />
1817             </xs:choice>
1818         </xs:restriction>
1819     </xs:complexContent>
1820 </xs:complexType>
1821
1822 <xs:complexType name="SmartcardPKIKeyStorageType">
1823     <xs:complexContent>
1824         <xs:restriction base="KeyStorageType">
1825             <xs:attribute name="medium" use="required">
1826                 <xs:simpleType>
1827                     <xs:restriction base="xs:NMTOKEN">
1828                         <xs:enumeration value="smartcard" />
1829                     </xs:restriction>
1830                 </xs:simpleType>
1831             </xs:attribute>
1832         </xs:restriction>
1833     </xs:complexContent>
1834 </xs:complexType>
1835
1836
1837 <xs:complexType name="SmartCardPKIPrivateKeyProtectionType">
1838     <xs:complexContent>
1839         <xs:restriction base="PrivateKeyProtectionType">
1840             <xs:sequence>
1841                 <xs:element ref="KeyActivation" />
1842                 <xs:element ref="KeyStorage" />
1843                 <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded" />
1844             </xs:sequence>
1845         </xs:restriction>
1846     </xs:complexContent>
1847 </xs:complexType>
1848
1849 </xs:schema>
1850
1851
    
```

1852 5.2.12. SoftwarePKI

1853 The Software-PKI class is identified when a Principal uses an X.509 certificate stored in software to authenticate to
 1854 the identity provider.

1855 5.2.12.1. Associated Liberty URI

1856 <http://www.projectliberty.org/schemas/authctx/classes/SoftwarePKI>

1857 5.2.12.2. Class Schema

```

1858
1859 <?xml version="1.0" encoding="UTF-8"?>
1860 <xs:schema targetNamespace="urn:liberty:ac:2003-08"
1861     xmlns="urn:liberty:ac:2003-08"
1862     xmlns:xs="http://www.w3.org/2001/XMLSchema"
1863     version="1.2-08"
1864     finalDefault="extension">
1865     <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd" />
1866     <xs:annotation>
1867         <xs:documentation>
1868             http://www.projectliberty.org/schemas/authctx/classes/SoftwarePKI</xs:documentation>
1869         </xs:annotation>
1870
1871     <xs:complexType name="SoftwarePKIPrincipalAuthenticationMechanismType">
1872         <xs:complexContent>
    
```

```

1873         <xs:restriction base="PrincipalAuthenticationMechanismType">
1874             <xs:sequence>
1875                 <xs:element ref="ActivationPin"/>
1876                 <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1877             </xs:sequence>
1878         </xs:restriction>
1879     </xs:complexContent>
1880 </xs:complexType>
1881
1882 <xs:complexType name="SoftwarePKIAuthenticatorType">
1883     <xs:complexContent>
1884         <xs:restriction base="AuthenticatorType">
1885             <xs:choice>
1886                 <xs:element ref="AsymmetricDecryption"/>
1887                 <xs:element ref="AsymmetricKeyAgreement"/>
1888                 <xs:element ref="DigSig"/>
1889             </xs:choice>
1890         </xs:restriction>
1891     </xs:complexContent>
1892 </xs:complexType>
1893
1894 <xs:complexType name="SoftwarePKIKeyActivationType">
1895     <xs:complexContent>
1896         <xs:restriction base="KeyActivationType">
1897             <xs:choice>
1898                 <xs:element ref="ActivationPin"/>
1899             </xs:choice>
1900         </xs:restriction>
1901     </xs:complexContent>
1902 </xs:complexType>
1903
1904 <xs:complexType name="SoftwarePKIPrivateKeyProtectionType">
1905     <xs:complexContent>
1906         <xs:restriction base="PrivateKeyProtectionType">
1907             <xs:sequence>
1908                 <xs:element ref="KeyActivation"/>
1909                 <xs:element ref="KeyStorage"/>
1910                 <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1911             </xs:sequence>
1912         </xs:restriction>
1913     </xs:complexContent>
1914 </xs:complexType>
1915
1916 <xs:complexType name="SoftwarePKIKeyStorageType">
1917     <xs:complexContent>
1918         <xs:restriction base="KeyStorageType">
1919             <xs:attribute name="medium" use="required">
1920                 <xs:simpleType>
1921                     <xs:restriction base="xs:NMTOKEN">
1922                         <xs:enumeration value="memory"/>
1923                     </xs:restriction>
1924                 </xs:simpleType>
1925             </xs:attribute>
1926         </xs:restriction>
1927     </xs:complexContent>
1928 </xs:complexType>
1929
1930 </xs:schema>
1931
1932
    
```

1933 5.2.13. TimeSyncToken

1934 The TimeSyncToken class is identified when a Principal authenticates through a time synchronization token.

1935 5.2.13.1. Associated Liberty URI

1936 <http://www.projectliberty.org/schemas/authctx/classes/TimeSyncToken>

1937 5.2.13.2. Class Schema

```
1938
1939 <?xml version="1.0" encoding="UTF-8"?>
1940 <xs:schema targetNamespace="urn:liberty:ac:2003-08"
1941   xmlns:xs="http://www.w3.org/2001/XMLSchema"
1942   xmlns="urn:liberty:ac:2003-08"
1943   finalDefault="extension">
1944   <xs:include schemaLocation="lib-arch-authentication-context-v1.2-08.xsd"/>
1945   <xs:annotation>
1946     <xs:documentation> http://www.projectliberty.org/schemas/authctx/classes/TimeSyncToken</xs:
1947 documentation>
1948   </xs:annotation>
1949   <xs:complexType name="TimeSyncTokenPrincipalAuthenticationMechanismType">
1950     <xs:complexContent>
1951       <xs:restriction base="PrincipalAuthenticationMechanismType">
1952         <xs:choice>
1953           <xs:element ref="Token"/>
1954         </xs:choice>
1955       </xs:restriction>
1956     </xs:complexContent>
1957   </xs:complexType>
1958   <xs:complexType name="TimeSyncTokenTokenType">
1959     <xs:complexContent>
1960       <xs:restriction base="TokenType">
1961         <xs:sequence>
1962           <xs:element ref="TimeSyncToken"/>
1963           <xs:element ref="Extension" minOccurs="0" maxOccurs="unbounded"/>
1964         </xs:sequence>
1965       </xs:restriction>
1966     </xs:complexContent>
1967   </xs:complexType>
1968   <xs:complexType name="TimeSyncTokenTimeSyncTokenType">
1969     <xs:complexContent>
1970       <xs:restriction base="TimeSyncTokenType">
1971         <xs:attribute name="DeviceType" use="required">
1972           <xs:simpleType>
1973             <xs:restriction base="xs:NMTOKEN">
1974               <xs:enumeration value="hardware"/>
1975             </xs:restriction>
1976           </xs:simpleType>
1977         </xs:attribute>
1978         <xs:attribute name="SeedLength" use="required">
1979           <xs:simpleType>
1980             <xs:restriction base="xs:integer">
1981               <xs:enumeration value="64"/>
1982             </xs:restriction>
1983           </xs:simpleType>
1984         </xs:attribute>
1985         <xs:attribute name="DeviceInHand" use="required">
1986           <xs:simpleType>
1987             <xs:restriction base="xs:NMTOKEN">
1988               <xs:enumeration value="true"/>
1989             </xs:restriction>
1990           </xs:simpleType>
1991         </xs:attribute>
1992       </xs:restriction>
1993     </xs:complexContent>
1994   </xs:complexType>
1995 </xs:schema>
1996
1997
```

1998 5.3. Authentication Context Classes Extensibility

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

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

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

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

2015 The processing rules for both Service and Identity Provider for Authentication Context Classes are listed in [[LibertyProtSchema](#)].
2016

2017 **References**

2018 **References**

- 2019 [LibertyProtSchema] Cantor, Scott, Kemp, John, eds. "Liberty ID-FF Protocols and Schema Specification," Version
2020 1.2-errata-v2.0, Liberty Alliance Project (12 September 2004). <http://www.projectliberty.org/specs>
- 2021 [RFC2119] Bradner, S., eds. "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, The Internet
2022 Engineering Task Force (March 1997). <http://www.ietf.org/rfc/rfc2119.txt> [March 1997].
- 2023 [Schema1] Thompson, Henry S., Beech, David, Maloney, Murray, Mendelsohn, Noah, eds. (May
2024 2002). "XML Schema Part 1: Structures," Recommendation, World Wide Web Consortium
2025 <http://www.w3.org/TR/xmlschema-1/>