



SAML V2.0 Deployment Profiles for X.509 Subjects

Committee Working Draft 0302

~~26 February 2008~~ 28 August 2007

Specification URIs:

This Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-cd-02.html>~~http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-draft-03.html~~

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-cd-02.odt>~~http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-draft-03.odt~~

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-cd-02.pdf>~~http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-draft-03.pdf~~

Previous Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-cd-02.html>~~http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-cd-01.html~~

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-cd-02.odt>~~http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-cd-01.odt~~

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-cd-02.pdf>~~http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-cd-01.pdf~~

Latest Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509.html>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509.odt>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509.pdf>

Latest Approved Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-cd-02.html>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-cd-02.odt>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-profiles-deploy-x509-cd-02.pdf>

Technical Committee:

OASIS Security Services TC

Chair(s):

Hal Lockhart, BEA Systems, Inc.

Brian Campbell, Ping Identity Corporation

38 **Editor(s):**
39 Tom Scavo, National Center for Supercomputing Applications (NCSA)

40 **Related Work:**
41 This specification is an alternative to the *SAML V2.0 Attribute Sharing Profile for X.509*
42 *Authentication-Based Systems* [SAMLASP].

43 **Declared XML Namespace(s):**
44 urn:oasis:names:tc:SAML:metadata:X509:query

45 **Abstract:**
46 This related set of SAML V2.0 deployment profiles specifies how a principal who has been issued
47 an X.509 identity certificate is represented as a SAML Subject, how an assertion regarding such a
48 principal is produced and consumed, and finally how two entities exchange attributes about such
49 a principal.

50 **Status:**
51 This document was last revised or approved by the SSTC on the above date. The level of
52 approval is also listed above. Check the current location noted above for possible later revisions
53 of this document. This document is updated periodically on no particular schedule.
54 TC members should send comments on this specification to the TC's email list. Others
55 should send comments to the TC by using the "Send A Comment" button on the TC's
56 web page at <http://www.oasis-open.org/committees/security>.
57 For information on whether any patents have been disclosed that may be essential to
58 implementing this specification, and any offers of patent licensing terms, please refer to the IPR
59 section of the TC web page (<http://www.oasis-open.org/committees/security/ipr.php>).
60 The non-normative errata page for this specification is located at [http://www.oasis-](http://www.oasis-open.org/committees/security)
61 [open.org/committees/security](http://www.oasis-open.org/committees/security).

Notices

62

63 | Copyright © OASIS Open 2007-~~2008~~. All Rights Reserved.

64 All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual
65 Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

66 This document and translations of it may be copied and furnished to others, and derivative works that
67 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published,
68 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice
69 and this section are included on all such copies and derivative works. However, this document itself may
70 not be modified in any way, including by removing the copyright notice or references to OASIS, except as
71 needed for the purpose of developing any document or deliverable produced by an OASIS Technical
72 Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be
followed) or as required to translate it into languages other than English.

66 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
67 or assigns.

67 This document and the information contained herein is provided on an "AS IS" basis and OASIS
68 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
69 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
70 OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
71 PARTICULAR PURPOSE.

68 OASIS requests that any OASIS Party or any other party that believes it has patent claims that would
69 necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to
70 notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such
71 patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced
72 this specification.

69 OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any
70 patent claims that would necessarily be infringed by implementations of this specification by a patent
71 holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR
72 Mode of the OASIS Technical Committee that produced this specification. OASIS may include such
73 claims on its website, but disclaims any obligation to do so.

70 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
71 might be claimed to pertain to the implementation or use of the technology described in this document or
72 the extent to which any license under such rights might or might not be available; neither does it represent
73 that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to
74 rights in any document or deliverable produced by an OASIS Technical Committee can be found on the
75 OASIS website. Copies of claims of rights made available for publication and any assurances of licenses
76 to be made available, or the result of an attempt made to obtain a general license or permission for the
77 use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS
78 Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any
79 information or list of intellectual property rights will at any time be complete, or that any claims in such list
80 are, in fact, Essential Claims.

71 The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be
72 used only to refer to the organization and its official outputs. OASIS welcomes reference to, and
73 implementation and use of, specifications, while reserving the right to enforce its marks against
74 misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

72 Table of Contents

73	1 Introduction.....	6
74	1.1 Terminology.....	6
75	1.2 Outline.....	7
76	1.3 Normative References.....	7
77	1.4 Non-Normative References.....	8
78	2 X.509 SAML Subject Profile.....	9
79	2.1 Required Information.....	9
80	2.2 Profile Description.....	9
81	2.3 <saml:Subject> Usage.....	9
82	2.3.1 <saml:NameID> Usage.....	9
83	2.3.2 <saml:EncryptedID> Usage.....	9
84	2.4 Example.....	10
85	3 SAML Attribute Query Deployment Profile for X.509 Subjects.....	11
86	3.1 Profile Overview (non-normative).....	11
87	3.2 Required Information.....	12
88	3.3 Profile Description.....	13
89	3.3.1 <samlp:AttributeQuery> Issued by Service Provider.....	13
90	3.3.2 <samlp:Response> Issued by Identity Provider.....	13
91	3.4 Use of SAML Request-Response Protocol.....	14
92	3.4.1 <samlp:AttributeQuery> Usage.....	14
93	3.4.2 <samlp:Response> Usage.....	14
94	3.5 Example.....	15
95	3.6 Use of Encryption.....	16
96	3.7 Use of Digital Signatures.....	17
97	3.8 Use of Metadata.....	17
98	3.8.1 Identity Provider Metadata.....	17
99	3.8.2 Service Provider Metadata.....	18
100	3.9 Security and Privacy Considerations.....	19
101	3.9.1 Background.....	19
102	3.9.2 General Security Requirements.....	19
103	3.9.3 User Privacy.....	19
104	3.10 Implementation Guidelines (non-normative).....	20
105	3.10.1 Discovery.....	20
106	3.10.2 Name Mapping.....	20
107	3.10.3 Canonicalization.....	20
108	3.10.4 Identity Provider Policy	20

109	3.10.5 Caching of Attributes	21
110	4 SAML Attribute Self-Query Deployment Profile for X.509 Subjects.....	22
111	4.1 Profile Overview (non-normative).....	22
112	4.2 Required Information.....	23
113	4.3 Profile Description.....	24
114	4.3.1 <samlp:AttributeQuery> Issued by Principal.....	24
115	4.3.2 <samlp:Response> Issued by Identity Provider.....	24
116	4.4 Use of SAML Request-Response Protocol.....	24
117	4.4.1 <samlp:AttributeQuery> Usage.....	24
118	4.4.2 <samlp:Response> Usage.....	24
119	4.4.3 Processing Rules.....	25
120	4.5 Example.....	25
121	4.6 Use of Metadata.....	27
122	4.6.1 Identity Provider Metadata.....	27
123	4.7 Security and Privacy Considerations.....	28
124	4.8 Implementation Guidelines (non-normative).....	28
125	4.8.1 Discovery.....	28
126	5 Implementation Conformance.....	30
127	6 Acknowledgments.....	31
128	7 Revision History.....	32
129		

1 Introduction

This related set of *SAML V2.0 Deployment Profiles for X.509 Subjects* describes how a principal who has been issued an X.509 identity certificate is represented as a SAML Subject, how an assertion regarding such a principal is produced and consumed, and finally how two entities exchange attributes about such a principal.

1.1 Terminology

This specification uses normative text to describe the use of SAML assertions and attribute queries for X.509 subjects.

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC 2119]:

...they MUST only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)...

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

Listings of XML schemas appear like this.

Example code listings appear like this.

Conventional XML namespace prefixes are used throughout the listings in this specification to stand for their respective namespaces as follows, whether or not a namespace declaration is present in the example:

Prefix	XML Namespace	Comments
saml:	urn:oasis:names:tc:SAML:2.0:assertion	This is the SAML V2.0 assertion namespace [SAMLCore]. This is the default namespace used throughout this document.
samlp:	urn:oasis:names:tc:SAML:2.0:protocol	This is the SAML V2.0 protocol namespace [SAMLCore].
md:	urn:oasis:names:tc:SAML:2.0:metadata	This is the SAML V2.0 metadata namespace [SAMLMeta].
query:	urn:oasis:names:tc:SAML:metadata:ext:query	This is the SAML metadata query extension namespace [SAMLMeta-Ext].
x509qry:	urn:oasis:names:tc:SAML:metadata:X509:query	This is the SAML X.509 query namespace defined by this document and its accompanying schema [X509Query-XSD].
ds:	http://www.w3.org/2000/09/xmldsig#	This is the W3C XML Signature namespace, defined in the XML-Signature Syntax and Processing specification and schema [XMLSig-XSD].
xenc:	http://www.w3.org/2001/04/xmlenc#	This is the W3C XML Encryption namespace, defined in the XML Encryption Syntax and Processing specification [XMLEnc] and schema [XMLEnc-XSD].
xs:	http://www.w3.org/2001/XMLSchema	This is the XML Schema namespace [Schema1].

Prefix	XML Namespace	Comments
xsi:	http://www.w3.org/2001/XMLSchema-instance	This is the XML Schema namespace for schema-related markup that appears in XML instances [Schema1].

141 This specification uses the following typographical conventions in text: <UnqualifiedElement>,
 142 <ns:QualifiedElement>, Attribute, **Datatype**, OtherKeyword.

142 The term *identity provider* as used in this specification refers to a typical SAML attribute authority
 143 [SAMLGloss]. The term *service provider* refers to a SAML attribute requester. However, as used in this
 144 specification, a service provider is not a typical SAML service provider since it performs X.509
 145 authentication in lieu of consuming a SAML authentication assertion.

143 The term *X.509 identity certificate* as used in this specification refers to an X.509 end entity certificate
 144 [RFC3280] or a certificate based on an X.509 end entity certificate (such as an X.509 proxy certificate
 145 [RFC3820]).

144 1.2 Outline

145 Section 2 describes how a principal who has been issued an X.509 identity certificate is represented as a
 146 SAML Subject. Section 3 describes in detail how a service provider and identity provider exchange
 147 attributes about a principal who has been issued an X.509 identity certificate. Section 4 describes the
 148 special case where the requester is the subject of the query, that is, where the principal self-queries for
 149 attributes. Finally, section 5 specifies requirements that all conforming implementations must follow.

146 1.3 Normative References

- 147 **[FIPS 140-2]** Security Requirements for Cryptographic Modules, May 2001. See
 148 <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>
- 148 **[RFC 2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF
 149 RFC 2119, March 1997. See <http://www.ietf.org/rfc/rfc2119.txt>
- 149 **[RFC2246]** T. Dierks and C. Allen. *The TLS Protocol Version 1.0*. IETF RFC 2246, January
 150 1999. See <http://www.ietf.org/rfc/rfc2246.txt>
- 150 **[RFC2253]** M. Wahl et al. *Lightweight Directory Access Protocol (v3): UTF-8 String*
 151 *Representation of Distinguished Names*. IETF RFC 2253, December 1997. See
 152 <http://www.ietf.org/rfc/rfc2253.txt>
- 153 **[RFC3280]** R. Housley et al. *Internet X.509 Public Key Infrastructure: Certificate and*
 154 *Certificate Revocation List (CRL) Profile*. IETF RFC 3280, April 2002. See
 155 <http://www.ietf.org/rfc/rfc3280.txt>
- 154 **[SAMLBind]** S. Cantor et al. *Bindings for the OASIS Security Assertion Markup Language*
 155 *(SAML) V2.0*. OASIS Standard, March 2005. See [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf)
 156 [open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf)
- 155 **[SAMLCore]** S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion*
 156 *Markup Language (SAML) V2.0*. OASIS Standard, March 2005. See
 157 <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>
- 156 **[SAMLMeta]** S. Cantor et al. *Metadata for the OASIS Security Assertion Markup Language*
 157 *(SAML) V2.0*. OASIS Standard, March 2005. See [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf)
 158 [open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf)
- 157 **[SAMLMeta-Ext]** T. Scavo and S. Cantor. *Metadata Extension for SAML V2.0 and V1.x Query*
 158 *Requesters*. OASIS ~~StandardDraft, November 2007~~ ~~September 2006~~. Document
 159 ID sstc-saml-metadata-ext-query-OSed-02. See [http://docs.oasis-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-ext-query-os.pdf)
 160 [open.org/security/saml/Post2.0/sstc-saml-metadata-ext-query-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-ext-query-os.pdf)
 161 [os.pdf](http://docs.oasis-open.org/security/saml/SpecDrafts-Post2.0/sstc-saml-metadata-ext-query-cd-02.pdf)[http://docs.oasis-open.org/security/saml/SpecDrafts-Post2.0/sstc-saml-](http://docs.oasis-open.org/security/saml/SpecDrafts-Post2.0/sstc-saml-metadata-ext-query-cd-02.pdf)
 162 [metadata-ext-query-cd-02.pdf](http://docs.oasis-open.org/security/saml/SpecDrafts-Post2.0/sstc-saml-metadata-ext-query-cd-02.pdf)

163 **[SAMLProf]** S. Cantor et al. *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. See <http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf>

164

165

166 **[Schema1]** H. S. Thompson et al. *XML Schema Part 1: Structures*. World Wide Web Consortium Recommendation, May 2001. See <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

167

168

169 **[SSL3]** A. Freier et al. *The SSL Protocol Version 3.0*, IETF Internet-Draft, November 1996. See <http://wp.netscape.com/eng/ssl3/draft302.txt>

170

171 **[X509Query-XSD]** *Schema for SAML V2.0 Deployment Profiles for X.509 Subjects*. OASIS, December 2006. Document ID sstc-saml-metadata-x509-query.xsd. See http://www.oasis-open.org/committees/documents.php?wg_abbrev=security

172

173

174 **[XMLEnc]** D. Eastlake et al. *XML Encryption Syntax and Processing*. World Wide Web Consortium Recommendation, December 2002. See <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>

175

176

177 **[XMLEnc-XSD]** *XML Encryption Schema*. World Wide Web Consortium Recommendation, December 2002. See <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-schema.xsd>

178

179

180 **[XMLSig]** D. Eastlake et al. *XML-Signature Syntax and Processing*. World Wide Web Consortium Recommendation, February 2002. See <http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/>

181

182

183 **[XMLSig-XSD]** *Schema for XML Signatures*. World Wide Web Consortium Recommendation, February 2002. See <http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-core-schema.xsd>

184

185

186 **1.4 Non-Normative References**

187 **[MACEAttrib]** S. Cantor et al. *MACE-Dir SAML Attribute Profiles*. Internet2 MACE, ~~December 2007~~ ~~April 2006~~. See <http://middleware.internet2.edu/dir/docs/draft-internet2-mace-dir-saml-attributes-latest.pdf> <http://middleware.internet2.edu/dir/docs/internet2-mace-dir-saml-attributes-200604.pdf>

188

189

190

191

192 **[RFC3820]** S. Tuecke et al. *Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile*. IETF RFC 3820, June 2004. See <http://www.ietf.org/rfc/rfc3820.txt>

193

194 **[SAMLASP]** R. Randall et al. *SAML V2.0 Attribute Sharing Profile for X.509 Authentication-Based Systems*. OASIS Committee Draft, August 2007. Document ID sstc-saml-x509-authn-attr-profile-cd-04.

195

196

197 **[SAMLGloss]** J. Hodges et al. *Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. See <http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf>

198

199

200 **[SAMLSecure]** F. Hirsch et al. *Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. See <http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf>

201

202

203 **2 X.509 SAML Subject Profile**

204 The X.509 SAML Subject Profile describes how a principal who has been issued an X.509 identity
205 certificate is represented as a SAML V2.0 Subject.

205 **2.1 Required Information**

206 **Identification:**

207 urn:oasis:names:tc:SAML:2.0:profiles:query:attribute:X509-subject

207 **Contact information:** security-services-comment@lists.oasis-open.org

208 **Description:** Given below.

209 **Updates:** N/A

210 **Extends:** N/A

211 **2.2 Profile Description**

212 This deployment profile specifies a SAML V2.0 `<saml:Subject>` element that represents a principal
213 who has been issued an X.509 identity certificate. An entity that produces a `<saml:Subject>` element
214 according to this deployment profile MUST have previously determined that the principal does in fact
215 possess the corresponding private key.

213 **2.3 `<saml:Subject>` Usage**

214 The `<saml:Subject>` element MUST contain exactly one of `<saml:NameID>` or
215 `<saml:EncryptedID>`. The `<saml:Subject>` element MAY contain one or more
216 `<saml:SubjectConfirmation>` elements that are out of scope for this deployment profile.

215 **2.3.1 `<saml:NameID>` Usage**

216 If the `<saml:Subject>` element contains a `<saml:NameID>` element, the following requirements MUST
217 be satisfied:

- 217 • The value of the `<saml:NameID>` element is the Subject Distinguished Name (DN) from the
218 principal's X.509 identity certificate.
- 218 • The `<saml:NameID>` element MUST have a `Format` attribute whose value is
219 `urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName`. Thus the DN value
220 of the `<saml:NameID>` element MUST satisfy the rules of section 8.3.3 of [SAMLCore]. Moreover,
221 for the purposes of this deployment profile, the DN value MUST conform to RFC 2253 [RFC2253].
- 219 • As specified in [SAMLCore], the `NameQualifier` attribute of the `<saml:NameID>` element
220 SHOULD be omitted.

220 **2.3.2 `<saml:EncryptedID>` Usage**

221 If the `<saml:Subject>` element contains a `<saml:EncryptedID>` element, the content of the
222 enclosed `<xenc:EncryptedData>` element MUST be an encrypted `<saml:NameID>` element that
223 satisfies the requirements of the previous section.

222 To encrypt the `<saml:NameID>` element, exactly one of the following procedures MUST be followed:

- 223 • The producer generates a new symmetric key to encrypt the `<saml:NameID>` element. After

224 performing the encryption, the producer places the resulting ciphertext in the
225 <xenc:EncryptedData> element. The symmetric key MUST be encrypted with the consumer's
226 public key and the resulting ciphertext MUST be placed in the <xenc:EncryptedKey> element.

- 225 • The producer uses a symmetric key previously established with the consumer to encrypt the
226 <saml:NameID> element. After performing the encryption, the producer places the resulting
227 ciphertext in the <xenc:EncryptedData> element. In this case, however, the
228 <saml:EncryptedID> element MUST NOT contain an <xenc:EncryptedKey> element.

226 A symmetric key transmitted in an <xenc:EncryptedKey> element MUST NOT be later reused by the
227 producer as a previously established symmetric key.

227 2.4 Example

228 An example of an unencrypted X.509 SAML Subject:

```
229 <!-- unencrypted X.509 SAML Subject -->  
230 <saml:Subject>  
231   <saml:NameID  
232     Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">  
233     C=US, O=NCSA-TEST, OU=User, CN=trscavo@uiuc.edu  
234   </saml:NameID>  
235 </saml:Subject>
```

236 An example of an encrypted X.509 SAML Subject:

```
237 <!-- encrypted X.509 SAML Subject -->  
238 <saml:Subject>  
239   <saml:EncryptedID  
240     xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">  
241     <xenc:EncryptedData  
242       Type="http://www.w3.org/2001/04/xmlenc#Element">  
243       ...  
244     </xenc:EncryptedData>  
245     <xenc:EncryptedKey  
246       Recipient="https://idp.example.org/saml">  
247       ...  
248     </xenc:EncryptedKey>  
249   </saml:EncryptedID>  
250 </saml:Subject>
```

251 3 SAML Attribute Query Deployment Profile for X.509 252 Subjects

252 The *SAML Attribute Query Deployment Profile for X.509 Subjects* specifies how a service provider and an
253 identity provider exchange attributes about a principal who has been issued an X.509 identity certificate.
254 As such, the profile relies on the X.509 SAML Subject Profile specified in section 2 of this document. Note
255 that the deployment profile specified in section 4 is an extension of this profile.

253 3.1 Profile Overview (non-normative)

254 Consider the use case where a principal attempts to access a secured resource at a service provider.
255 Principal authentication at the service provider is accomplished by presenting a trusted X.509 identity
256 certificate and by demonstrating proof of possession of the associated private key.

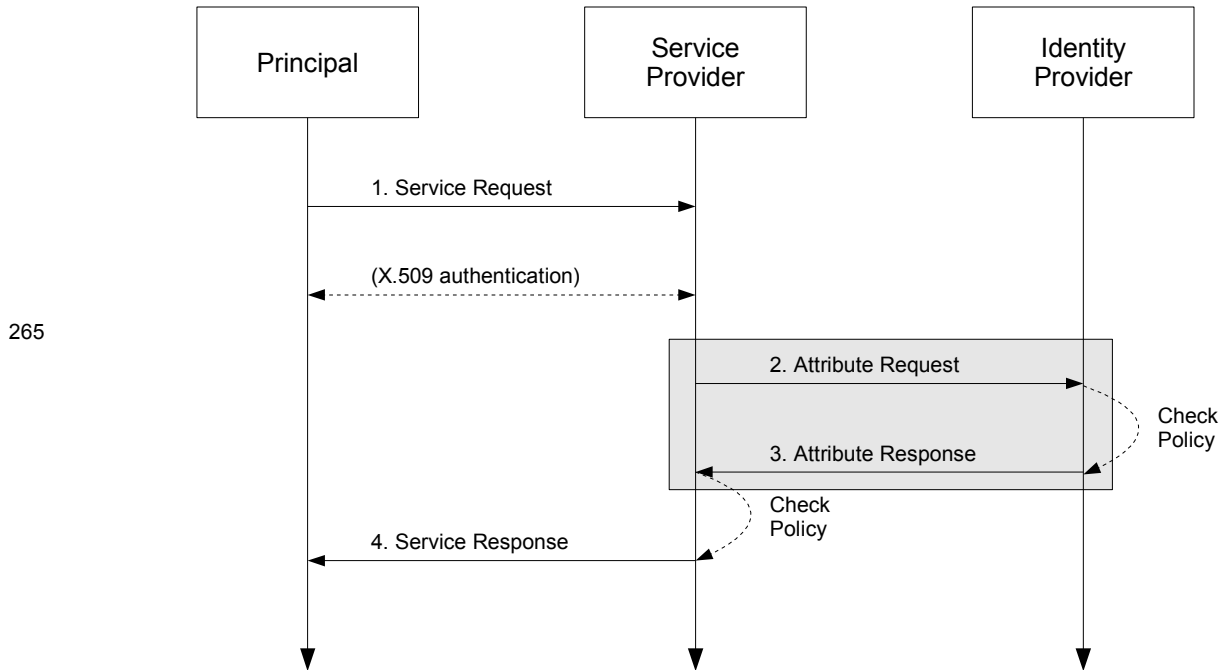
255 After the principal has been authenticated, the service provider requires additional information about the
256 principal in order to determine whether to grant access to the resource. To obtain this information, the
257 service provider uses the Subject Distinguished Name (DN) field (and perhaps other information) from the
258 principal's X.509 identity certificate to query an identity provider for attributes about the principal. Using the
259 attributes received from the identity provider, the service provider is able to make an informed access
260 control decision.

256 This use case is based upon the following assumptions:

- 257 • A principal possesses an X.509 identity credential.
- 258 • The principal wields a client that requests a service from a service provider.
- 259 • The client can access the principal's X.509 identity credential.
- 260 • The principal has an account with a SAML identity provider.
- 261 • The service provider knows the principal's preferred identity provider and is able to query that
262 identity provider for attributes.
- 262 • The identity provider is able to map an X.509 SAML Subject (as defined in section 2 of this
263 document) to one and only one principal in its security domain. In particular, the identity provider is
264 able to map the X.509 SAML Subject that represents this principal.

263 The sequence of steps for the full use case is shown below.

264 **Note:** The steps constrained by this profile are highlighted with a gray box. The other
265 steps are shown only for completeness; the profile does not constrain them.



265

266 **1. Service Request**

267 In step 1, the principal requests a secured resource from a service provider who requires that the
 268 principal be authenticated. The principal authenticates to the service provider with an X.509 identity
 269 certificate.

268 **2. Attribute Request**

269 In step 2, the service provider sends a SAML V2.0 `<samlp:AttributeQuery>` message to the
 270 identity provider using a SAML SOAP Binding. The Subject DN from the principal's X.509 identity
 271 certificate (presented in step 1) is used to construct the `<saml:Subject>` element.

270 **3. Attribute Response**

271 In step 3, after verifying that the service provider is a valid requester, the identity provider issues a
 272 `<samlp:Response>` message containing appropriate attributes pertaining to the principal. The
 273 attributes returned to the service provider are subject to policy at the identity provider.

272 **4. Service Response**

273 In step 4, based on the attributes received from the identity provider, the service provider returns the
 274 requested resource or an error, subject to policy.

274 Of the sequence of steps described above, it is steps 2 and 3 that are profiled in sections 3.3 and 3.4 of
 275 this deployment profile.

275 **3.2 Required Information**

276 **Identification:**

277 `urn:oasis:names:tc:SAML:2.0:profiles:query:attribute:X509`

277 **Contact information:** security-services-comment@lists.oasis-open.org

278 **Description:** Given below.

279 **Updates:** N/A

280 **Extends:** Assertion Query/Request Profile [SAMLProf]

281 **3.3 Profile Description**

282 This deployment profile describes the use of the SAML V2.0 Assertion Query and Request Protocol
283 [SAMLCore] in conjunction with the SAML V2.0 SOAP Binding [SAMLBind] to retrieve the attributes of a
284 principal who has authenticated using an X.509 identity certificate. The attribute exchange MUST conform
285 to the Assertion Query/Request Profile given in section 6 of [SAMLProf] unless otherwise specified below.

283 As outlined in section 3.1, a service provider sends a SAML V2.0 `<samlp:AttributeQuery>` message
284 directly to an identity provider. This message contains a name identifier that identifies a principal who has
285 authenticated to the service provider using an X.509 identity certificate. If the identity provider receiving the
286 request can:

- 284 • recognize the name identifier; and
- 285 • fulfill the request subject to any applicable policies;

286 the identity provider responds with a successful `<samlp:Response>` containing the relevant attributes for
287 the identified principal.

287 **3.3.1 `<samlp:AttributeQuery>` Issued by Service Provider**

288 To initiate the profile, the service provider uses a synchronous binding such as the SAML SOAP Binding
289 [SAMLBind] to send a SAML V2.0 `<samlp:AttributeQuery>` message to an Attribute Service
290 endpoint at the identity provider. SAML metadata (section 3.8) MAY be used to determine the endpoint
291 locations and bindings supported by the identity provider.

289 The service provider uses information obtained from the principal's X.509 identity certificate to construct
290 the query. As required by the X.509 SAML Subject Profile (section 2), the service provider MUST have
291 previously determined that the principal does in fact possess the corresponding private key. The details of
292 this step are out of scope for this deployment profile.

290 The service provider MUST authenticate itself to the identity provider. SSL 3.0 [SSL3] or TLS 1.0
291 [RFC2246] with client authentication MAY be used for this purpose and to provide integrity protection and
292 confidentiality. Also, the `<samlp:AttributeQuery>` element MAY be signed.

291 **3.3.2 `<samlp:Response>` Issued by Identity Provider**

292 The identity provider MUST process the request as outlined in [SAMLCore]. After processing the message
293 or upon encountering an error, the identity provider MUST return a `<samlp:Response>` message
294 containing an appropriate status code to the service provider to complete the SAML protocol exchange. If
295 the identity provider is successful in locating one or more attributes for this principal, they will be included
296 in the response.

293 The identity provider MUST be able to map the referenced X.509 Subject to one and only one principal in
294 its security domain. If the identity provider is not able to map the `<saml:Subject>` element to a local
295 principal, it MUST return an error.

294 The identity provider processes the `<samlp:AttributeQuery>` element and any enclosed
295 `<saml:Attribute>` elements before returning an assertion containing a
296 `<saml:AttributeStatement>` to the requester. If no `<saml:Attribute>` elements are included in
297 the query, the identity provider returns all attributes for this principal, subject to policy. SAML metadata
298 (section 3.8) MAY be used to determine the attribute requirements of the service provider. If the identity
299 provider is unable to resolve attributes for this principal (for any reason), it MUST return an error.

295 The identity provider MUST authenticate itself to the service provider. Also, either the
296 `<samlp:Response>` element or the `<saml:Assertion>` element (or both) MAY be signed.

296 **3.4 Use of SAML Request-Response Protocol**

297 As required by the Assertion Query/Request Profile [SAMLProf], the `<samlp:AttributeQuery>`
298 element MUST contain a `<saml:Issuer>` element.

298 **3.4.1 `<samlp:AttributeQuery>` Usage**

299 The request MUST contain a `<samlp:AttributeQuery>` element that conforms to the following rules:

- 300 • The `<saml:Subject>` element MUST conform to the X.509 SAML Subject Profile defined in
301 section 2 of this document.
- 301 • The `<saml:Subject>` element MUST NOT contain a `<saml:SubjectConfirmation>`
302 element.
- 302 • The `<samlp:AttributeQuery>` element MAY include one or more `<saml:Attribute>`
303 elements.

303 **3.4.2 `<samlp:Response>` Usage**

304 If the request is successful, the `<samlp:Response>` element MUST conform to the following rules. Any
305 assertion(s) included in the response may be encrypted or unencrypted. See section 2 of the SAML V2.0
306 Assertions and Protocols specification [SAMLCore] for general requirements regarding SAML assertions.

305 For each `<saml:Assertion>` element the following conditions MUST be satisfied:

- 306 • The `<saml:Subject>` element (which strongly matches the subject of the query [SAMLCore])
307 SHOULD NOT contain a `<saml:SubjectConfirmation>` element.
- 307 • The `<saml:Assertion>` element MUST contain a `<saml:Conditions>` element with
308 `NotBefore` and `NotOnOrAfter` attributes.
- 308 • The `<saml:Assertion>` element SHOULD contain a `<saml:Audience>` element whose value
309 is identical to the value of the `<saml:Issuer>` element in the request.
- 309 • Other conditions (including other `<saml:Audience>` elements) MAY be included as required by
310 the service provider or at the discretion of the identity provider.
- 310 • The `<saml:Assertion>` element MUST contain at least one `<saml:AttributeStatement>`
311 element and SHOULD contain *only* `<saml:AttributeStatement>` elements.

311 For each `<saml:EncryptedAssertion>` element, the content of the enclosed
312 `<xenc:EncryptedData>` element MUST be an encrypted `<saml:Assertion>` element that satisfies
313 the above requirements.

312 To encrypt the `<saml:Assertion>` element, exactly one of the following procedures MUST be followed:

- 313 • The identity provider generates a new symmetric key to encrypt the `<saml:Assertion>` element.
314 After performing the encryption, the identity provider places the resulting ciphertext in the
315 `<xenc:EncryptedData>` element. The symmetric key MUST be encrypted with the service
316 provider's public key and the resulting ciphertext placed in the `<xenc:EncryptedKey>` element.
- 314 • The identity provider uses a symmetric key previously established with the service provider to
315 encrypt the `<saml:Assertion>` element. After encrypting the `<saml:Assertion>` element
316 using this key, the identity provider places the resulting ciphertext in the `<xenc:EncryptedData>`
317 element. In this case, however, the `<saml:EncryptedAssertion>` element MUST NOT contain
318 an `<xenc:EncryptedKey>` element.

315 See section 3.6 for additional rules regarding encryption.

316 If the request is unsuccessful and the identity provider wishes to return an error, the `<samlp:Response>`

317 element MUST NOT contain a <saml:Assertion> element. Possible error responses include the
318 following:

- 318 • The identity provider MAY return one of the status codes
319 urn:oasis:names:tc:SAML:2.0:status:UnknownAttrProfile or
320 urn:oasis:names:tc:SAML:2.0:status:InvalidAttrNameOrValue as suggested in
321 section 3.3.2.3 of [SAMLCore].
- 319 • If the identity provider does not recognize the <saml:NameID> element or otherwise is unable to
320 map the <saml:NameID> element to a local principal name, it MAY return the following status
321 code:
322 urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal

320 3.5 Example

321 For example, the requester issues the following attribute query:

```
322 <samlp:AttributeQuery
323   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
324   xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
325   ID="aaf23196-1773-2113-474a-fe114412ab72"
326   Version="2.0"
327   IssueInstant="2006-07-17T22:26:40Z">
328   <saml:Issuer>https://sp.example.org/saml</saml:Issuer>
329   <saml:Subject>
330     <saml:NameID
331       Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">
332       C=US, O=NCSA-TEST, OU=User, CN=trscavo@uiuc.edu
333     </saml:NameID>
334   </saml:Subject>
335   <saml:Attribute
336     xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
337     x500:Encoding="LDAP"
338     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
339     Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
340     FriendlyName="eduPersonPrincipalName">
341   </saml:Attribute>
342   <saml:Attribute
343     xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
344     x500:Encoding="LDAP"
345     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
346     Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
347     FriendlyName="eduPersonAffiliation">
348   </saml:Attribute>
349 </samlp:AttributeQuery>
```

346 After processing the request, the identity provider issues the following response:

```
347 <samlp:Response
348   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
349   xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
350   InResponseTo="aaf23196-1773-2113-474a-fe114412ab72"
351   ID="b07b804c-7c29-ea16-7300-4f3d6f7928ac"
352   Version="2.0"
353   IssueInstant="2006-07-17T22:26:41Z">
354   <saml:Issuer>https://idp.example.org/saml</saml:Issuer>
355   <samlp:Status>
356     <samlp:StatusCode
357       Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
358   </samlp:Status>
359   <saml:Assertion
360     xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
361     xmlns:xs="http://www.w3.org/2001/XMLSchema"
362     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
363     xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
364     ID="a144e8f3-adad-594a-9649-924517abe933">
```

```

365     Version="2.0"
366     IssueInstant="2006-07-17T22:26:41Z">
367     <saml:Issuer>https://idp.example.org/saml</saml:Issuer>
368     <saml:Subject>
369         <saml:NameID
370             Format="urn:oasis:names:tc:SAML:1.1:nameid-
371 format:X509SubjectName">
371             C=US, O=NCSA-TEST, OU=User, CN=trscavo@uiuc.edu
372         </saml:NameID>
373     </saml:Subject>
374     <!-- assertion lifetime constrained by principal's X.509 cert -->
375     <saml:Conditions
376         NotBefore="2006-07-17T22:21:41Z"
377         NotOnOrAfter="2006-07-17T22:51:41Z">
378         <saml:AudienceRestriction>
379             <saml:Audience>https://sp.example.org/saml</saml:Audience>
380         </saml:AudienceRestriction>
381     </saml:Conditions>
382     <saml:AttributeStatement>
383         <saml:Attribute
384             xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
385             x500:Encoding="LDAP"
386             NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
387             Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
388             FriendlyName="eduPersonPrincipalName">
389             <saml:AttributeValue xsi:type="xs:string">
390                 trscavo@uiuc.edu
391             </saml:AttributeValue>
392         </saml:Attribute>
393         <saml:Attribute
394             xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
395             x500:Encoding="LDAP"
396             NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
397             Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
398             FriendlyName="eduPersonAffiliation">
399             <saml:AttributeValue xsi:type="xs:string">
400                 member
401             </saml:AttributeValue>
402             <saml:AttributeValue xsi:type="xs:string">
403                 staff
404             </saml:AttributeValue>
405         </saml:Attribute>
406     </saml:AttributeStatement>
407 </saml:Assertion>
408 </samlp:Response>

```

409 The attributes in the above example (eduPersonAffiliation and eduPersonPrincipalName)
410 conform to the MACE-Dir Attribute Profile for SAML 2.0 [MACEAttrib] and are for illustration purposes
411 only.

412 3.6 Use of Encryption

413 If the service provider encrypts the <saml:NameID> element in the query, the identity provider SHOULD
414 encrypt any resulting assertions. Moreover, if the service provider uses a previously established symmetric
415 key, the identity provider SHOULD use the same symmetric key to encrypt the assertion. In the case
416 where the service provider generates a new symmetric key, the identity provider MUST treat this key as a
417 previously established key, that is, the identity provider SHOULD use the same symmetric key to encrypt
418 the assertion and MUST NOT encrypt this key into the <xenc:EncryptedKey> element.

419 An encryption algorithm satisfying FIPS 140-2 Security Requirements [FIPS 140-2] SHALL be used for all
420 encryption operations.

421 3.7 Use of Digital Signatures

422 If the service provider encrypts the `<saml:NameID>` element in the query, the
423 `<samlp:AttributeQuery>` element MUST be signed *after* the encryption operation takes place. If the
424 identity provider encrypts a `<saml:Assertion>` element in the response, the `<saml:Assertion>`
425 element MUST be signed *before* the encryption operation takes place. Whether or not an assertion is
426 encrypted, the `<saml:Response>` element MAY be signed.

423 A signing algorithm satisfying FIPS 140-2 Security Requirements [FIPS 140-2] SHALL be used for all
424 digital signature operations on encrypted elements or elements with encrypted content.

424 3.8 Use of Metadata

425 The identity provider and the service provider MAY use metadata for locating endpoints, communicating
426 key information, and so forth. The use of SAML V2.0 metadata [SAMLMeta], which is RECOMMENDED,
427 is profiled in sections 3.8.1 and 3.8.2 below.

426 3.8.1 Identity Provider Metadata

427 An identity provider that uses SAML V2.0 metadata MUST include an
428 `<md:AttributeAuthorityDescriptor>` element that satisfies the following rules:

- 428 • The containing `<md:EntityDescriptor>` element MUST have an `entityID` attribute whose
429 value is the same unique identifier given as the `<saml:Issuer>` element in assertions issued by
430 the identity provider.
- 429 • The `<md:AttributeAuthorityDescriptor>` element MUST include an
430 `<md:NameIDFormat>` element with value `"urn:oasis:names:tc:SAML:1.1:nameid-`
431 `format:X509SubjectName"`.
- 430 • One or more `<saml:Attribute>` elements MAY be included in the
431 `<md:AttributeAuthorityDescriptor>` element. Since a service provider may choose not to
432 query the identity provider based on the attributes in this list, this list SHOULD be comprehensive or
433 otherwise omitted.

431 To distinguish between this deployment profile and other uses of `X509SubjectName`, an identity provider
432 requires the means to explicitly call out its support of this deployment profile. An XML attribute has been
433 specified for this purpose [X509Query-XSD]:

```
432 <xs:attribute  
433   name="supportsX509Query" type="boolean" use="optional"/>
```

434 Use of this attribute is OPTIONAL. An identity provider that chooses to use this attribute, however, MUST
435 do so as follows:

- 435 • The `<md:AttributeAuthorityDescriptor>` element MUST include at least one
436 `<md:AttributeService>` element having attribute `supportsX509Query` set to `"true"`.
- 436 • At least one `<md:AttributeService>` element having attribute `supportsX509Query` set to
437 `"true"` MUST have its `Binding` attribute set to
438 `"urn:oasis:names:tc:SAML:2.0:bindings:SOAP"`.

437 An example of identity provider metadata follows:

```
438 <!-- An Identity Provider supporting this deployment profile -->  
439 <md:EntityDescriptor  
440   xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"  
441   entityID="https://idp.example.org/saml">  
442  
443   <md:AttributeAuthorityDescriptor  
444     protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
```

```

446 <md:AttributeService
447   x509qry:supportsX509Query="true"
448   xmlns:x509qry="urn:oasis:names:tc:SAML:metadata:X509:query"
449   Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
450   Location="https://idp.example.org:8443/saml-idp/AA"/>
451
452 <md:NameIDFormat>
453   urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
454 </md:NameIDFormat>
455
456 <!-- see [MACEAttr] -->
457 <md:AttributeProfile>
458   urn:mace:dir:profiles:attribute:samlv2
459 </md:AttributeProfile>
460
461 </md:AttributeAuthorityDescriptor>
462
463 </md:EntityDescriptor>

```

464 3.8.2 Service Provider Metadata

465 A service provider that uses SAML V2.0 metadata **MUST** include an `<md:RoleDescriptor>` element
466 that satisfies the following rules:

- 467 • The containing `<md:EntityDescriptor>` element **MUST** have an `entityID` attribute whose
468 value is the same unique identifier used as the `<saml:Issuer>` element in attribute queries
469 issued by the service provider.
- 467 • The type of the `<md:RoleDescriptor>` element **MUST** be derived from type
468 **query:AttributeQueryDescriptorType** [SAMLMeta-Ext].
- 468 • The `<md:RoleDescriptor>` element **MUST** include an `<md:NameIDFormat>` element with
469 value "urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName".
- 469 • One or more `<md:RequestedAttribute>` elements **MAY** be included in the
470 `<md:AttributeConsumingService>` element.

470 An example of service provider metadata follows:

```

471 <!-- A Service Provider supporting this profile -->
472 <md:EntityDescriptor
473   xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
474   entityID="https://sp.example.org/saml">
475
476   <md:RoleDescriptor
477     xmlns:query="urn:oasis:names:tc:SAML:metadata:ext:query"
478     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
479     xsi:type="query:AttributeQueryDescriptorType"
480     protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
481
482     <md:NameIDFormat>
483       urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
484     </md:NameIDFormat>
485
486     <md:AttributeConsumingService isDefault="true" index="0">
487       <md:ServiceName xml:lang="en">
488         Grid Service Provider
489       </md:ServiceName>
490       <md:RequestedAttribute
491         NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
492         Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
493         FriendlyName="eduPersonPrincipalName">
494       </md:RequestedAttribute>
495       <md:RequestedAttribute
496         NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
497         Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"

```

```
498         FriendlyName="eduPersonAffiliation">
499         </md:RequestedAttribute>
500         </md:AttributeConsumingService>
501
502     </md:RoleDescriptor>
503
504 </md:EntityDescriptor>
```

505 The attributes in the above example (`eduPersonAffiliation` and `eduPersonPrincipalName`)
506 conform to the MACE-Dir Attribute Profile for SAML 2.0 [MACEAttrib] and are for illustration purposes
507 only.

506 3.9 Security and Privacy Considerations

507 The motivation for this deployment profile is to specify a secure means of obtaining SAML attributes in
508 conjunction with X.509 authentication.

508 3.9.1 Background

509 The SAML Security and Privacy specification [SAMLSecure] provides general background material
510 relevant to all SAML bindings and profiles. Section 6.1 of [SAMLSecure], in particular, considers the
511 security requirements of the SAML SOAP Binding, and is therefore pertinent to this deployment profile. In
512 addition, section 3.1.2 of the SAML Bindings specification [SAMLBind] provides further security guidelines
513 regarding SAML bindings.

514 3.9.2 General Security Requirements

515 SAML profiles often involve a system entity that relies on an earlier act of user authentication. For
516 example, the SAML Web Browser SSO Profile [SAMLProf] relies on an authentication service that
517 validates a credential (typically a username/password) for a user. The authentication service must be
518 securely linked to an identity provider that issues SAML authentication assertions based on that user's act
519 of authentication. Similarly, this deployment profile assumes that the system entity that performs the
520 X.509 authentication is operating in a secure environment that includes the attribute requester.

521 In this deployment profile, an end user presents an X.509 identity certificate to authenticate at the service
522 provider. The system entity that performs this authentication (i.e., validates the certificate and its trust
523 chain) must be securely linked to the SAML attribute requester that subsequently initiates this deployment
524 profile. The latter must have a secure means of obtaining the X.509 subject name (and other information)
525 from the certificate and issuing a SAML V2.0 `<samlp:AttributeQuery>` for that subject to the
526 appropriate asserting party. The mechanism by which these system entities are linked is out of scope for
527 this deployment profile.

528 Local policy settings at the attribute authority will determine whether or not the asserting party is permitted
529 to return attributes for the requested subject.

530 3.9.3 User Privacy

531 Since a DN persists for the life of the certificate, a service provider may query for attributes at any time.
532 To prevent service providers from querying for attributes after the certificate has expired, an identity
533 provider SHOULD check the lifetime of the referenced certificate before issuing an assertion regarding an
534 X.509 Subject. If the certificate has expired, an error should be returned.

535 As a further privacy measure, the principal may use a short-lived X.509 identity certificate. For example,
536 an X.509 proxy certificate [RFC3820]) may be used.

537 **3.10 Implementation Guidelines (non-normative)**

538 The following non-normative guidelines are provided for the convenience of implementers.

539 **3.10.1 Discovery**

540 The service provider must determine the principal's preferred identity provider. This is called *identity*
541 *provider discovery*.

542 Some possible approaches to identity provider discovery in the context of this deployment profile are
543 discussed briefly below:

- 544 • The identity provider's unique identifier may be preconfigured at the service provider. This is useful,
545 for instance, if there is only one identity provider per deployment.
- 546 • The subject DN of the principal's X.509 identity certificate may include a reference to the identity
547 provider. New deployments are discouraged from decorating long-lived DNs in this manner,
548 however, since this practice may lessen interoperability with existing PKIs. For short-lived X.509
549 identity certificates, this practice may be satisfactory.
- 550 • The issuer DN or the issuer alternative name may provide clues about the principal's preferred
551 identity provider. This technique may not be practical, however, since SAML authorities do not
552 typically issue X.509 credentials.
- 553 • A reference to the identity provider may be inserted into a non-critical X.509 extension [RFC3280] at
554 the time the credential is issued. For long-term credentials, this practice may not be feasible, but
555 for short-term credentials, this technique may be satisfactory.

556 This deployment profile does not specify a particular method of identity provider discovery.

557 **3.10.2 Name Mapping**

558 An identity provider that consumes a `<saml:Subject>` element produced according to this deployment
559 profile must be able to map the referenced X.509 Subject to one and only one principal in its security
560 domain. If the identity provider issued the X.509 credential in the first place, or otherwise has access to
561 the principal's X.509 identity certificate, this should be straightforward. Otherwise a persistent certificate
562 registration process to facilitate the mapping of X.509 Subjects to principals may be used.

563 **3.10.3 Canonicalization**

564 According to this deployment profile, the format of the DNs used to construct the `<saml:Subject>`
565 element is dictated by [SAMLCore]. Since the latter allows some flexibility in the precise format of a DN
566 (by virtue of its dependence on [RFC2253]), it may be necessary for an identity provider to canonicalize
567 the DN during the course of mapping it to a local principal name. Note that the details of the
568 canonicalization process are of concern only to the identity provider. As long as the service provider
569 provides a DN whose canonicalization is recognized by the identity provider, the correct mapping will
570 occur.

571 **3.10.4 Identity Provider Policy**

572 Service providers may explicitly enumerate the required attributes in queries or may issue so-called
573 "empty queries" that essentially request all available attributes. Regardless of the attribute requirements
574 called out in the query (or in metadata, if used for this purpose), it is the identity provider that determines
575 the actual attributes returned to the service provider. Thus a responsible identity provider will initiate and
576 enforce policy that strictly limits the attributes released to service providers.

577 **3.10.5 Caching of Attributes**

578 A service provider will most likely provide a capability to cache user attributes returned in assertions. If so,
579 cache expiration settings should be configurable by administrators.

580 4 SAML Attribute Self-Query Deployment Profile for 581 X.509 Subjects

582 The *SAML Attribute Self-Query Deployment Profile for X.509 Subjects* specifies how a principal who has
583 been issued an X.509 identity certificate self-queries an identity provider for attributes. The profile extends
584 the SAML Attribute Query Deployment Profile for X.509 Subjects specified in section 3 of this document.
585 Where the two profiles conflict, this deployment profile takes precedence.

586 4.1 Profile Overview (non-normative)

587 In this scenario, a principal self-queries an identity provider for attributes. The principal uses the Subject
588 Distinguished Name (DN) field (and perhaps other information) from its X.509 identity certificate to
589 formulate the query. Principal authentication is accomplished by presenting a trusted X.509 identity
590 certificate (the same certificate used to construct the query) and by demonstrating proof of possession of
591 the associated private key. After the principal has been authenticated, the identity provider binds the
592 principal's public key to an assertion, which is issued directly to the principal.

593 The principal subsequently requests a secured resource at the service provider. The principal presents
594 the previously obtained assertion to the service provider and demonstrates proof of possession of the
595 corresponding private key. Using the attributes in the assertion, the service provider is able to make an
596 informed access control decision.

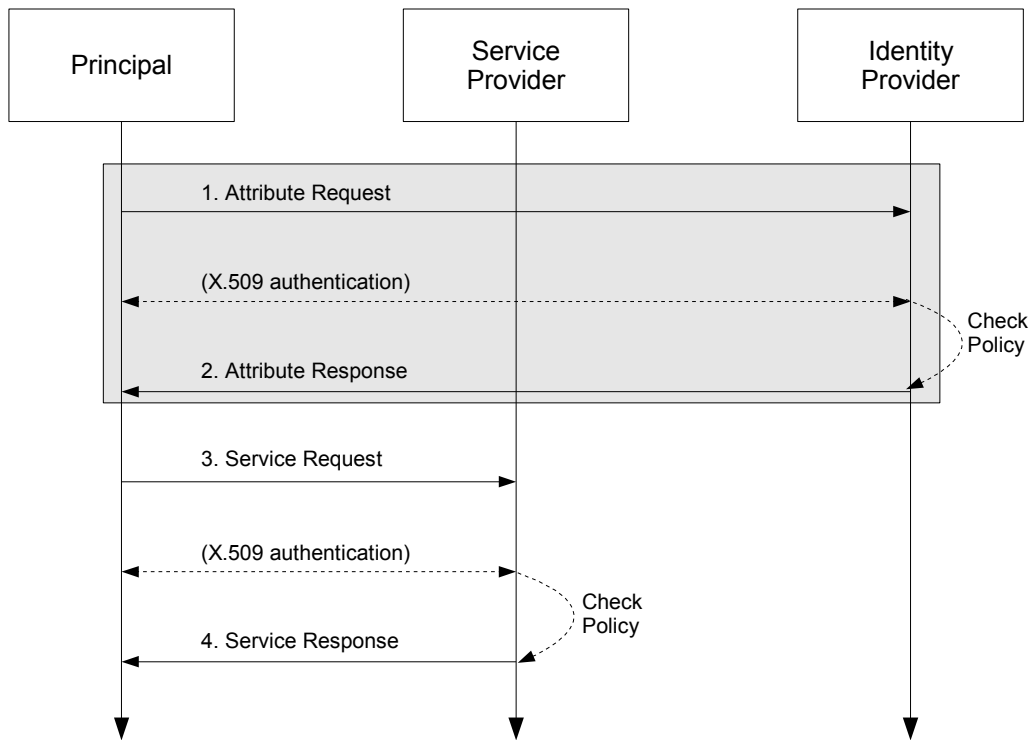
597 This use case is based on the following assumptions:

- 598 • A principal possesses an X.509 credential.
- 599 • The principal wields a client that can both query an identity provider for attributes and request a
600 service from a service provider.
- 601 • The client can access the principal's X.509 credential.
- 602 • The principal has an account with a SAML identity provider.
- 603 • The client knows the principal's preferred identity provider and the attribute requirements of the
604 target service provider.
- 605 • The identity provider is able to map an X.509 SAML Subject (as defined in section 2 of this
606 document) to one and only one principal in its security domain. In particular, the identity provider is
607 able to map the X.509 SAML Subject that represents this principal.

608 Note that in the case of a self-query, the client possesses significantly more functionality than the client
609 alluded to in section 3.1.

610 The sequence of steps for the full use case is shown below.

611 **Note:** The steps constrained by this profile are highlighted with a gray box. The other
612 steps are shown only for completeness; the profile does not constrain them.



613

614 **1. Attribute Request**

615 In step 1, the principal sends a SAML V2.0 `<samlp:AttributeQuery>` message to the identity
 616 provider using a SAML SOAP Binding. The Subject DN from the principal's X.509 identity certificate is
 617 used to construct the `<saml:Subject>` element of the query. The identity provider requires that the
 618 principal be authenticated. The principal authenticates to the identity provider using the same X.509
 619 credential used to construct the query.

620 **2. Attribute Response**

621 In step 2, after verifying that the principal is a valid requester, the identity provider issues a
 622 `<samlp:Response>` message containing appropriate attributes. The attributes returned to the
 623 principal are subject to policy at the identity provider.

624 **3. Service Request**

625 In step 3, the principal requests a secured resource at the service provider. The principal presents the
 626 assertion obtained at step 2 to the service provider. The service provider requires that the principal be
 627 authenticated. The principal authenticates to the service provider using the same X.509 credential
 628 used to authenticate to the identity provider at step 1.

629 **4. Service Response**

630 In step 4, based on the attributes in the pushed assertion, the service provider returns the requested
 631 resource or an error, subject to policy.

632 Of the sequence of steps described above, it is steps 1 and 2 that are profiled in sections 4.3 and 4.4 of
 633 this deployment profile.

634 **4.2 Required Information**

635 **Identification:**

636 `urn:oasis:names:tc:SAML:2.0:profiles:query:attribute:X509-self`

637 **Contact information:** security-services-comment@lists.oasis-open.org

638 **Description:** Given below.

639 **Updates:** N/A

640 **Extends:** SAML Attribute Query Deployment Profile for X.509 Subjects (section 3)

641 **4.3 Profile Description**

642 This deployment profile extends the SAML Attribute Query Deployment Profile for X.509 Subjects
643 described in section 3.3.

644 As outlined in section 4.1, a principal sends a SAML V2.0 `<samlp:AttributeQuery>` message directly
645 to an identity provider. The principal authenticates to the identity provider using an X.509 identity
646 certificate. If the identity provider receiving the request can:

- 647 • recognize the name identifier; and
- 648 • determine that the requester is the principal; and
- 649 • fulfill the request subject to any applicable policies;

650 the identity provider responds with a successful `<samlp:Response>` containing the relevant attributes for
651 the principal. To determine that the requester is the principal, the identity provider MUST authenticate the
652 principal.

653 **4.3.1 `<samlp:AttributeQuery>` Issued by Principal**

654 To initiate the profile, the principal uses a synchronous binding such as the SAML SOAP Binding
655 [SAMLBind] to send a SAML V2.0 `<samlp:AttributeQuery>` message as described in section 3.3.
656 The principal uses information obtained from its X.509 identity certificate to construct the query. The
657 principal MUST authenticate itself to the identity provider using the same X.509 credential used to
658 construct the query. SSL 3.0 [SSL3] or TLS 1.0 [RFC2246] with client authentication MAY be used for this
659 purpose and to provide integrity protection and confidentiality.

660 **4.3.2 `<samlp:Response>` Issued by Identity Provider**

661 The identity provider MUST process the request as outlined in section 3.3.

662 **4.4 Use of SAML Request-Response Protocol**

663 As required by the Assertion Query/Request Profile [SAMLProf], the `<samlp:AttributeQuery>`
664 element MUST contain a `<saml:Issuer>` element. Since the requester is the principal, the
665 `<saml:Issuer>` element MUST be identical to the `<saml:NameID>` element, that is, both MUST satisfy
666 the rules of the X.509 SAML Subject Profile (section 2).

667 **4.4.1 `<samlp:AttributeQuery>` Usage**

668 The request MUST contain a `<samlp:AttributeQuery>` element that conforms to the rules of
669 section 3.4.1.

670 **4.4.2 `<samlp:Response>` Usage**

671 If the request is successful, the `<samlp:Response>` element MUST conform to the rules of section 3.4.2
672 except as noted below:

- 673 • The `<saml:Subject>` element MUST contain a `<saml:SubjectConfirmation>` element

- 674 whose Method attribute has value "urn:oasis:names:tc:SAML:2.0:cm:holder-of-key".
- 675 • A <saml:SubjectConfirmationData> element MUST be present and it MUST contain a
 - 676 <ds:KeyInfo> element that refers to the principal's X.509 identity certificate.
 - 677 • On the <saml:Conditions> element, the value of the NotBefore attribute (resp., the
 - 678 NotOnOrAfter attribute) MUST be greater than or equal to (resp., less than or equal to) the
 - 679 NotBefore field (resp., the NotOnOrAfter field) of the certificate.
 - 680 • The <saml:Assertion> element MUST be signed.
 - 681 • The <saml:Assertion> element MAY include a <saml:AuthnStatement> element.

682 4.4.3 Processing Rules

683 In addition to the assertion processing rules outlined in [SAMLCore], the service provider MUST verify the
684 following:

- 685 • The <saml:SubjectConfirmationData> element MUST be present and it MUST contain a
- 686 <ds:KeyInfo> element that refers to the principal's X.509 identity certificate.
- 687 • The value of the NotBefore attribute (resp., the NotOnOrAfter attribute) MUST be greater than
- 688 or equal to (resp., less than or equal to) the NotBefore field (resp., the NotOnOrAfter field) of
- 689 the certificate.

690 The certificate referred to in the above processing rules MUST be the same certificate used to construct
691 the <saml:Subject> of the query.

692 4.5 Example

693 For example, the principal issues the following attribute query:

```
694 <samlp:AttributeQuery
695   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
696   xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
697   ID="aaf23196-1773-2113-474a-fe114412ab72"
698   Version="2.0"
699   IssueInstant="2006-07-17T20:31:40Z">
700   <saml:Issuer
701     Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">
702     C=US, O=NCSA-TEST, OU=User, CN=trscavo@uiuc.edu
703   </saml:Issuer>
704   <saml:Subject>
705     <saml:NameID
706       Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">
707       C=US, O=NCSA-TEST, OU=User, CN=trscavo@uiuc.edu
708     </saml:NameID>
709   </saml:Subject>
710   <saml:Attribute
711     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
712     Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
713     FriendlyName="eduPersonPrincipalName">
714   </saml:Attribute>
715   <saml:Attribute
716     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
717     Name="urn:oid:2.5.4.42"
718     FriendlyName="givenName">
719   </saml:Attribute>
720   <saml:Attribute
721     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
722     Name="urn:oid:2.5.4.4"
723     FriendlyName="sn">
724   </saml:Attribute>
725   <saml:Attribute
```

```
726     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
727     Name="urn:oid:1.3.6.1.4.1.1466.115.121.1.26"
728     FriendlyName="mail">
729   </saml:Attribute>
730 </samlp:AttributeQuery>
```

731 After processing the request, the identity provider issues a response containing an assertion such as the
732 one listed below. Note that the assertion was obtained by a principal who authenticated to an identity
733 provider via TLS [RFC2246] client authentication, as indicated in the <saml:AuthnStatement>
734 element.

```
735 <!-- SAML Assertion for an X.509 Subject -->
736 <saml:Assertion
737   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
738   xmlns:xs="http://www.w3.org/2001/XMLSchema"
739   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
740   xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
741   ID="_33776a319493ad607b7ab3e689482e45"
742   Version="2.0"
743   IssueInstant="2006-07-17T20:31:41Z">
744   <saml:Issuer>https://idp.example.org/saml</saml:Issuer>
745   <ds:Signature>...</ds:Signature>
746   <saml:Subject>
747     <saml:NameID
748       Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">
749       C=US, O=NCSA-TEST, OU=User, CN=trscavo@uiuc.edu
750     </saml:NameID>
751     <saml:SubjectConfirmation
752       Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
753       <saml:SubjectConfirmationData
754         <ds:KeyInfo>
755           <ds:X509Data>
756             <!-- principal's X.509 cert -->
757             <ds:X509Certificate>
758 MIIcDCCAXACCQDE+9eiWrm62jANBqkqhkiG9w0BAQQFADBFMQswCQYDVQGGewJV
759 UzESMBAGA1UEChMJKNTQs1URVNUMQ0wCwYDVQQLEwRvc2VYMRMwEQYDVQDEwpt
760 UC1TZXJ2aWNlMB4XDTA2MDcxNzIwMjE0MVoXDTA2MDcxODIwMjE0MVoVszELMAK
761 GAlUEBhmCVVMxExAgAQBgNVBAoTCU5DU0EtVEVTVFVDENMASGA1UECxMEVXN1cjE
762 ZMBCGAlUEAwQdHJzY2F2b0B1aXVjLmVkdTCBnzANBqkqhkiG9w0BAQEFAAObjQ
763 AwgYkcgYEAv9QMe4lRl3XbWPcflbCjGK9gty6zBJmp+tsaJINM0VaBaZ3t+tSX
764 knelYife nCc2O3yaX76aq53QMxy+5wKQYe8Rzdw28Nv3a73wFjXJXoUhGkvE
765 rCscs9EfIWCc g2bH0g8uSh+Fbv3lHih4lBJ5MCS2buJfsR7d1r/xsadU2RcCA
766 WEAATANBgkqhkiG9w0BAQQFAAOCQAEdyIcMTob7TVkelfJ7+I1j0LO24UlKvbL
767 zd2OPvcFTcv6fVHx Ejk0QxaZXJhreZ6+rIdiMXrEz1RdJEsNMxtDW8++sVp6avo
768 B5EX1y3ez+CEAIL4g cJvKZUR4dMryWshWIBHKFFul+r7urUgvWI12KbMeE9KP+
769 kiiiiTskLcKgFzngw1J selmHhTcTcRcDocn5yO2+d3dog52vSOTVFDsBuvDixO2
770 hv679JR6Hlqjtk4GExp E9iVI0wdPE038uQIJJTXlhsMMLvUGVh/c0ReJbn92V
771 j4dI/yy6PtY/8ncYLYNkjg oVN0J/yMoktn9lTlFyTiuY4OuJsZR01+zWLy9g==
772           </ds:X509Certificate>
773         </ds:X509Data>
774       </ds:KeyInfo>
775     </saml:SubjectConfirmationData>
776   </saml:SubjectConfirmation>
777 </saml:Subject>
778 <!-- assertion lifetime constrained by principal's X.509 cert -->
779 <saml:Conditions
780   NotBefore="2006-07-17T20:31:41Z"
781   NotOnOrAfter="2006-07-18T20:21:41Z">
782 </saml:Conditions>
783 <saml:AuthnStatement
784   AuthnInstant="2006-07-17T20:31:41Z">
785   <saml:AuthnContext>
786     <saml:AuthnContextClassRef>
787       urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient
788     </saml:AuthnContextClassRef>
789   </saml:AuthnContext>
790 </saml:AuthnStatement>
```

```

791 <saml:AttributeStatement>
792   <saml:Attribute
793     xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
794     x500:Encoding="LDAP"
795     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
796     Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
797     FriendlyName="eduPersonPrincipalName">
798     <saml:AttributeValue xsi:type="xs:string">
799       trscavo@uiuc.edu
800     </saml:AttributeValue>
801   </saml:Attribute>
802   <saml:Attribute
803     xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
804     x500:Encoding="LDAP"
805     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
806     Name="urn:oid:2.5.4.42"
807     FriendlyName="givenName">
808     <saml:AttributeValue xsi:type="xs:string">
809       Tom
810     </saml:AttributeValue>
811   </saml:Attribute>
812   <saml:Attribute
813     xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
814     x500:Encoding="LDAP"
815     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
816     Name="urn:oid:2.5.4.4"
817     FriendlyName="sn">
818     <saml:AttributeValue xsi:type="xs:string">
819       Scavo
820     </saml:AttributeValue>
821   </saml:Attribute>
822   <saml:Attribute
823     xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
824     x500:Encoding="LDAP"
825     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
826     Name="urn:oid:1.3.6.1.4.1.1466.115.121.1.26"
827     FriendlyName="mail">
828     <saml:AttributeValue xsi:type="xs:string">
829       trscavo@gmail.com
830     </saml:AttributeValue>
831   </saml:Attribute>
832 </saml:AttributeStatement>
833 </saml:Assertion>

```

834 The attributes in the above example (eduPersonPrincipalName, givenName, sn, and mail) conform
835 to the MACE-Dir Attribute Profile for SAML 2.0 [MACEAttrib] and are for illustration purposes only.

836 4.6 Use of Metadata

837 As outlined in section 3.8, the use of SAML V2.0 metadata [SAMLMeta] is RECOMMENDED, but since a
838 principal is not expected to publish metadata about itself, only the use of identity provider metadata is
839 profiled below. Note, however, that the principal may wield a client that relies on service provider metadata
840 (see, e.g., section 4.8.1), in which case the rules in section 3.8.2 apply as well.

841 4.6.1 Identity Provider Metadata

842 An identity provider that uses SAML V2.0 metadata MUST include an
843 <md:AttributeAuthorityDescriptor> element that satisfies the rules given in section 3.8.1, except
844 that in this case the identity provider uses XML attribute supportsX509SelfQuery instead of
845 supportsX509Query [X509Query-XSD]:

```
846 <xsi:attribute
```

847 `name="supportsX509SelfQuery" type="boolean" use="optional"/>`

848 As before, use of this attribute is OPTIONAL.

849 An example of identity provider metadata follows:

```
850 <!-- An Identity Provider supporting both deployment profiles -->
851 <md:EntityDescriptor
852   xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
853   entityID="https://idp.example.org/saml">
854
855   <md:AttributeAuthorityDescriptor
856     protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
857
858     <md:AttributeService
859       x509qry:supportsX509Query="true"
860       x509qry:supportsX509SelfQuery="true"
861       xmlns:x509qry="urn:oasis:names:tc:SAML:metadata:X509:query"
862       Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
863       Location="https://idp.example.org:8443/saml-idp/AA"/>
864
865     <md:NameIDFormat>
866       urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
867     </md:NameIDFormat>
868
869     <!-- see [MACEAttr] -->
870     <md:AttributeProfile>
871       urn:mace:dir:profiles:attribute:samlv2
872     </md:AttributeProfile>
873
874   </md:AttributeAuthorityDescriptor>
875
876 </md:EntityDescriptor>
```

877 Note that this identity provider supports both X.509 attribute query deployment profiles at the same
878 endpoint location.

879 4.7 Security and Privacy Considerations

880 Except for section 3.9.2, the security and privacy considerations outlined in section 3.9 apply equally as
881 well in the case of self-query. As a further privacy measure, a principal may limit the self-query to non-
882 identity attributes (such as givenName) and push the resulting assertion to the service provider who
883 subsequently queries the identity provider for additional attributes (according to the deployment profile in
884 section 3). In this way, a service provider receives only those attributes that are actually required for
885 access.

886 4.8 Implementation Guidelines (non-normative)

887 In addition to the guidelines outlined in section 3.10, the following non-normative guidelines are provided
888 for the convenience of implementers.

889 4.8.1 Discovery

890 In the SAML Attribute Query Deployment Profile for X.509 Subjects (section 3), we encounter the problem
891 of identity provider discovery (section 3.10.1). In the case where the principal self-queries for attributes, we
892 encounter a different problem, which we call *service provider discovery*. In both cases, we assume the
893 client knows the principal's preferred identity provider, so identity provider discovery is a non-issue in the
894 case of self-queries, but in that case the client is faced with a self-query for unknown attributes.

895 If the client had access to the published metadata of potential service providers, and that metadata
896 included the attribute requirements of the service providers, the client would be able to formulate specific
897 attribute queries targeted for specific service providers.

898 This deployment profile does not specify a particular method of service provider discovery.

899 5 Implementation Conformance

900 ~~A client implementation of this specification shall be a conforming *Extended Mode X.509 Attribute Query*~~
901 ~~*Requester* or a conforming *Extended Mode X.509 Attribute Self-Query Requester* (or both). On the server~~
902 ~~side, an implementation of this specification shall be a conforming *Extended Mode X.509 Attribute Query*~~
903 ~~*Responder* or a conforming *Extended Mode X.509 Attribute Self-Query Responder*, respectively.~~An-
904 ~~implementation of this specification shall be a conforming *Extended Mode X.509 Attribute-*~~
905 ~~*Query/Requester* or a conforming *Extended Mode X.509 Attribute Self-Query/Requester* (or both). An~~
906 ~~*Extended Mode X.509 Attribute Self-Query/Requester* is a functional superset of an *Extended Mode X.509-*~~
907 ~~*Attribute Query/Requester*.~~

908 An Extended Mode X.509 Attribute Query ~~/Requester~~ or Responder MUST conform to the relevant
909 normative statements in section 3. An Extended Mode X.509 Attribute Self-Query ~~/Requester~~ or
910 Responder MUST conform to the relevant normative statements in section 4, which includes references to
911 normative portions of section 3.

912 **6 Acknowledgments**

913 The editors would like to acknowledge the contributions of the OASIS Security Services Technical
914 Committee, whose voting members at the time of publication were:

- 915 • ~~George Fletcher, AOL~~
- 916 • Hal Lockhart, BEA Systems, Inc.
- 917 • ~~Steve Anderson, BMC Software~~
- 918 • ~~Christopher Laskowski, Booz Allen Hamilton~~
- 919 • Rob Philpott, EMC Corporation
- 920 • ~~Carolina Canales Valenzuela, Ericsson~~
- 921 • ~~Ashish Patel, France Telecom~~
- 922 • ~~Greg Whitehead, Hewlett-Packard~~
- 923 • ~~Heather Hinton, IBM~~
- 924 • ~~Anthony Nadalin, IBM~~
- 925 • Eric Tiffany, ~~IEEE Industry Standards and Technology Org (IEEE-ISTO)~~Liberty Alliance Project
- 926 • Scott Cantor, Internet2
- 927 • Bob Morgan, Internet2
- 928 • Tom Scavo, National Center for Supercomputing Applications (NCSA)
- 929 • ~~Peter Davis, Neustar, Inc.~~
- 930 • Jeff Hodges, Neustar, Inc.
- 931 • Frederick Hirsch, Nokia Corporation
- 932 • ~~Abbie Barbir, Nortel Networks Limited~~
- 933 • Paul Madsen, NTT Corporation
- 934 • Ari Kermaier, Oracle Corporation
- 935 • ~~Prateek Mishra, Oracle Corporation~~
- 936 • Brian Campbell, Ping Identity Corporation
- 937 • ~~Anil Saldhana, Red Hat~~
- 938 • ~~Eve Maler, Sun Microsystems~~
- 939 • Emily Xu, Sun Microsystems
- 940 • ~~Kent Spaulding, Tripod Technology Group, Inc.~~
- 941 • David Staggs, Veterans Health Administration

942 The editors would also like to acknowledge the contributions of the following individuals:

- 943 • Von Welch, National Center for Supercomputing Applications (NCSA)

944

7 Revision History

<i>Document ID</i>	<i>Date</i>	<i>Committer</i>	<i>Comment</i>
sstc-saml2-profiles-deploy-x509-draft-01	18 Dec 2006	T. Scavo	Initial draft.
sstc-saml2-profiles-deploy-x509-draft-02	26 Mar 2007	T. Scavo	
sstc-saml2-profiles-deploy-x509-cd-01	07 May 2007	T. Scavo	Committee Draft
sstc-saml2-profiles-deploy-x509-cd-02	28 Aug 2007	T. Scavo	Committee Draft
sstc-saml2-profiles-deploy-x509-draft-03	26 Feb 2008	T. Scavo	

945