



SAML V2.0 Deployment Profiles for X.509 Subjects

OASIS Draft, 18 December 2006

Document identifier:

sstc-saml2-profiles-deploy-x509-draft-01

Location:

http://www.oasis-open.org/committees/documents.php?wg_abbrev=security

Editors:

Tom Scavo, NCSA/University of Illinois

Contributors:

Von Welch, NCSA/University of Illinois

Abstract:

This related set of SAML V2.0 deployment profiles specifies how a principal who has been issued an X.509 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.

Status:

This is a **Working Draft** of the Security Services Technical Committee.

Committee members should submit comments and potential errata to the security-services@lists.oasis-open.org list. Others should submit them by filling out the web form located at http://www.oasis-open.org/committees/comments/form.php?wg_abbrev=security. The committee will publish on its web page (<http://www.oasis-open.org/committees/security>) a catalog of any changes made to this document as a result of comments.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights web page for the Security Services TC (<http://www.oasis-open.org/committees/security/ipr.php>).

29 **Table of Contents**

30	1 Introduction.....	4
31	1.1 Notation.....	4
32	1.2 Terminology.....	5
33	1.3 Outline.....	5
34	2 X.509 SAML Subject Deployment Profile.....	6
35	2.1 Required Information.....	6
36	2.2 Profile Description.....	6
37	2.3 <saml:Subject> Usage.....	6
38	2.3.1 <saml:NameID> Usage.....	6
39	2.3.2 <saml:EncryptedID> Usage.....	6
40	2.4 Example.....	7
41	3 SAML Attribute Query Deployment Profile for X.509 Subjects.....	8
42	3.1 Profile Overview (non-normative).....	8
43	3.2 Required Information.....	9
44	3.3 Profile Description.....	10
45	3.3.1 <samlp:AttributeQuery> Issued by Service Provider.....	10
46	3.3.2 <samlp:Response> Issued by Identity Provider.....	10
47	3.4 Use of SAML Request-Response Protocol.....	11
48	3.4.1 <samlp:AttributeQuery> Usage.....	11
49	3.4.2 <samlp:Response> Usage.....	11
50	3.5 Example.....	12
51	3.6 Use of Encryption.....	13
52	3.7 Use of Digital Signatures.....	14
53	3.8 Use of Metadata.....	14
54	3.8.1 Identity Provider Metadata.....	14
55	3.8.2 Service Provider Metadata.....	15
56	3.9 Security and Privacy Considerations.....	16
57	3.9.1 Background.....	16
58	3.9.2 General Security Requirements.....	16
59	3.9.3 User Privacy.....	16
60	3.10 Implementation Guidelines (non-normative).....	16
61	3.10.1 Discovery.....	17
62	3.10.2 Name Mapping.....	17
63	3.10.3 Canonicalization.....	17
64	3.10.4 Identity Provider Policy	17
65	3.10.5 Caching of Attributes	17

66	4 SAML Attribute Self-Query Deployment Profile for X.509 Subjects.....	18
67	4.1 Profile Overview (non-normative).....	18
68	4.2 Required Information.....	19
69	4.3 Profile Description.....	20
70	4.3.1 <samlp:AttributeQuery> Issued by Principal.....	20
71	4.3.2 <samlp:Response> Issued by Identity Provider.....	20
72	4.4 Use of SAML Request-Response Protocol.....	20
73	4.4.1 <samlp:AttributeQuery> Usage.....	20
74	4.4.2 <samlp:Response> Usage.....	20
75	4.4.3 Processing Rules.....	21
76	4.5 Example.....	21
77	4.6 Use of Metadata.....	23
78	4.6.1 Identity Provider Metadata.....	23
79	4.7 Security and Privacy Considerations.....	24
80	4.8 Implementation Guidelines (non-normative).....	24
81	4.8.1 Discovery.....	24
82	5 References.....	25
83	5.1 Normative References.....	25
84	5.2 Non-Normative References.....	26
85	6 Revision History.....	27
86		

87 1 Introduction

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

92 1.1 Notation

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

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

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

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

103 `Listings of XML schemas appear like this.`

104 `Example code listings appear like this.`

106 Conventional XML namespace prefixes are used throughout the listings in this specification to stand for
107 their respective namespaces as follows, whether or not a namespace declaration is present in the
108 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 [XMLSig] 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].

109 This specification uses the following typographical conventions in text: <SAML*Element*>,
110 <ns:ForeignElement>, Attribute, **Datatype**, OtherKeyword.

111 1.2 Terminology

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

116 The term *X.509 certificate* as used in this specification refers to an X.509 v3 end entity certificate
117 [RFC3280] or a certificate based on an X.509 v3 end entity certificate (such as an X.509 proxy certificate
118 [RFC3820]).

119 1.3 Outline

120 Section 2 describes how a principal who has been issued an X.509 certificate is represented as a SAML
121 Subject. Section 3 describes in detail how a service provider and identity provider exchange attributes
122 about a principal who has been issued an X.509 certificate. Finally, section 4 describes the special case
123 where the requester is the subject of the query, that is, where the principal self-queries for attributes.

124 2 X.509 SAML Subject Deployment Profile

125 The X.509 SAML Subject Deployment Profile describes how a principal who has been issued an X.509
126 certificate is represented as a SAML V2.0 Subject.

127 2.1 Required Information

128 **Identification:**

129 urn:oasis:names:tc:SAML:2.0:profiles:deploy:X509:subject

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

131 **Description:** Given below.

132 **Updates:** N/A

133 **Extends:** N/A

134 2.2 Profile Description

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

138 2.3 `<saml:Subject>` Usage

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

142 2.3.1 `<saml:NameID>` Usage

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

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

153 2.3.2 `<saml:EncryptedID>` Usage

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

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

- 158 • The producer generates a new symmetric key to encrypt the `<saml:NameID>` element. After
159 performing the encryption, the producer places the resulting ciphertext in the

160 <xenc:EncryptedData> element. The symmetric key MUST be encrypted with the consumer's
161 public key and the resulting ciphertext MUST be placed in the <xenc:EncryptedKey> element.

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

166 2.4 Example

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

```
168 <!-- unencrypted X.509 SAML Subject -->  
169 <saml:Subject>  
170   <saml:NameID  
171     Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">  
172     C=US, O=NCSA-TEST, OU=User, CN=trscavo@uiuc.edu  
173   </saml:NameID>  
174 </saml:Subject>
```

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

```
176 <!-- encrypted X.509 SAML Subject -->  
177 <saml:Subject>  
178   <saml:EncryptedID  
179     xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">  
180     <xenc:EncryptedData  
181       Type="http://www.w3.org/2001/04/xmlenc#Element">  
182       ...  
183     </xenc:EncryptedData>  
184     <xenc:EncryptedKey  
185       Recipient="https://idp.example.org/saml">  
186       ...  
187     </xenc:EncryptedKey>  
188   </saml:EncryptedID>  
189 </saml:Subject>
```

3 SAML Attribute Query Deployment Profile for X.509 Subjects

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

3.1 Profile Overview (non-normative)

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

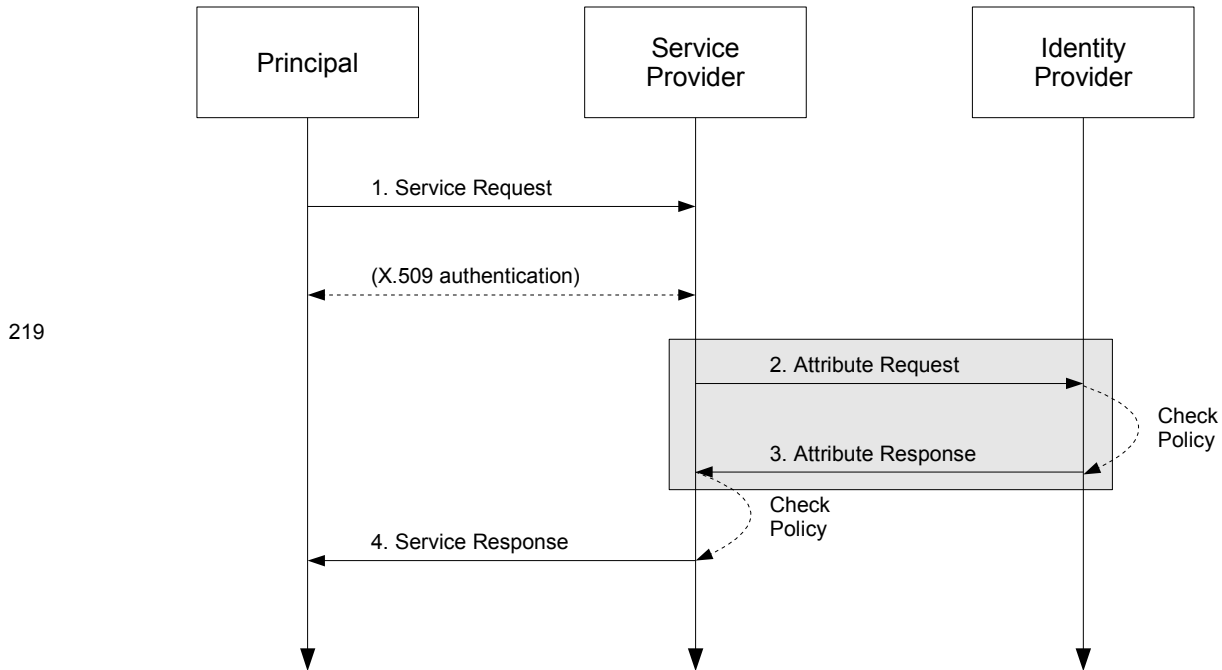
After the principal has been authenticated, the service provider requires additional information about the principal in order to determine whether to grant access to the resource. To obtain this information, the service provider uses the Subject Distinguished Name (DN) field (and perhaps other information)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

This use case is based upon the following assumptions:

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

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

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



219

220 **1. Service Request**

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

223 **2. Attribute Request**

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

227 **3. Attribute Response**

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

231 **4. Service Response**

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

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

236 **3.2 Required Information**

237 **Identification:**

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

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

240 **Description:** Given below.

241 **Updates:** N/A

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

243 **3.3 Profile Description**

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

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

- 252 • recognize the name identifier; and
- 253 • fulfill the request subject to any applicable policies;

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

256 **3.3.1 `<samlp:AttributeQuery>` Issued by Service Provider**

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

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

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

268 **3.3.2 `<samlp:Response>` Issued by Identity Provider**

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

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

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

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

285 3.4 Use of SAML Request-Response Protocol

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

288 3.4.1 `<samlp:AttributeQuery>` Usage

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

- 290 • The `<saml:Subject>` element MUST conform to the X.509 SAML Subject Deployment Profile
291 defined in section 2 of this document.
- 292 • The `<saml:Subject>` element MUST NOT contain a `<saml:SubjectConfirmation>`
293 element.
- 294 • The `<samlp:AttributeQuery>` element MAY include one or more `<saml:Attribute>`
295 elements.

296 3.4.2 `<samlp:Response>` Usage

297 If the request is successful, the `<samlp:Response>` element MUST contain at least one assertion. The
298 assertion(s) may be encrypted or unencrypted. See section 2 of the SAML V2.0 Assertions and Protocols
299 specification [SAMLCore] for general requirements regarding SAML assertions.

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

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

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

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

324 See section 3.6 for additional rules regarding encryption.

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

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

- 328 • If the identity provider does not support this profile, it MAY return the following status code:
329 urn:oasis:names:tc:SAML:2.0:status:UnknownAttrProfile
- 330 • If the identity provider does not recognize the <saml:NameID> element or otherwise is unable to
331 map the <saml:NameID> element to a local principal name, it MAY return the following status
332 code:
333 urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal

334 3.5 Example

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

```
336 <samlp:AttributeQuery
337   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
338   xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
339   ID="aaf23196-1773-2113-474a-fe114412ab72"
340   Version="2.0"
341   IssueInstant="2006-07-17T22:26:40Z">
342   <saml:Issuer>https://sp.example.org/saml</saml:Issuer>
343   <saml:Subject>
344     <saml:NameID
345       Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">
346       C=US, O=NCSA-TEST, OU=User, CN=trscavo@uiuc.edu
347     </saml:NameID>
348   </saml:Subject>
349   <saml:Attribute
350     xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
351     x500:Encoding="LDAP"
352     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
353     Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
354     FriendlyName="eduPersonPrincipalName">
355   </saml:Attribute>
356   <saml:Attribute
357     xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
358     x500:Encoding="LDAP"
359     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
360     Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
361     FriendlyName="eduPersonAffiliation">
362   </saml:Attribute>
363 </samlp:AttributeQuery>
```

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

```
365 <samlp:Response
366   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
367   xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
368   InResponseTo="aaf23196-1773-2113-474a-fe114412ab72"
369   ID="b07b804c-7c29-ea16-7300-4f3d6f7928ac"
370   Version="2.0"
371   IssueInstant="2006-07-17T22:26:41Z">
372   <saml:Issuer>https://idp.example.org/saml</saml:Issuer>
373   <samlp:Status>
374     <samlp:StatusCode
375       Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
376   </samlp:Status>
377   <saml:Assertion
378     xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
379     xmlns:xs="http://www.w3.org/2001/XMLSchema"
380     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
381     xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
382     ID="a144e8f3-adad-594a-9649-924517abe933"
383     Version="2.0"
384     IssueInstant="2006-07-17T22:26:41Z">
```

```

385     <saml:Issuer>https://idp.example.org/saml</saml:Issuer>
386     <saml:Subject>
387         <saml:NameID
388             Format="urn:oasis:names:tc:SAML:1.1:nameid-
389 format:X509SubjectName">
390             C=US, O=NCSA-TEST, OU=User, CN=trscavo@uiuc.edu
391         </saml:NameID>
392     </saml:Subject>
393     <!-- assertion lifetime constrained by principal's X.509 cert -->
394     <saml:Conditions
395         NotBefore="2006-07-17T22:21:41Z"
396         NotOnOrAfter="2006-07-17T22:51:41Z">
397         <saml:AudienceRestriction>
398             <saml:Audience>https://sp.example.org/saml</saml:Audience>
399         </saml:AudienceRestriction>
400     </saml:Conditions>
401     <saml:AttributeStatement>
402         <saml:Attribute
403             xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
404             x500:Encoding="LDAP"
405             NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
406             Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
407             FriendlyName="eduPersonPrincipalName">
408             <saml:AttributeValue xsi:type="xs:string">
409                 trscavo@uiuc.edu
410             </saml:AttributeValue>
411         </saml:Attribute>
412         <saml:Attribute
413             xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
414             x500:Encoding="LDAP"
415             NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
416             Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
417             FriendlyName="eduPersonAffiliation">
418             <saml:AttributeValue xsi:type="xs:string">
419                 member
420             </saml:AttributeValue>
421             <saml:AttributeValue xsi:type="xs:string">
422                 staff
423             </saml:AttributeValue>
424         </saml:Attribute>
425     </saml:AttributeStatement>
426 </saml:Assertion>
427 </samlp:Response>

```

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

431 3.6 Use of Encryption

432 If the service provider encrypts the `<saml:NameID>` element in the query, the identity provider SHOULD
433 encrypt any resulting assertions. Moreover, if the service provider uses a previously established symmetric
434 key, the identity provider SHOULD use the same symmetric key to encrypt the assertion. In the case
435 where the service provider generates a new symmetric key, the identity provider MUST treat this key as a
436 previously established key, that is, the identity provider SHOULD use the same symmetric key to encrypt
437 the assertion and MUST NOT encrypt this key into the `<xenc:EncryptedKey>` element.

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

440 3.7 Use of Digital Signatures

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

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

448 3.8 Use of Metadata

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

452 3.8.1 Identity Provider Metadata

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

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

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

```
468 <xs:attribute  
469   name="supportsX509Query" type="boolean" use="optional"/>
```

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

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

477 An example of identity provider metadata follows:

```
478 <!-- An Identity Provider supporting this deployment profile -->  
479 <md:EntityDescriptor  
480   xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"  
481   entityID="https://idp.example.org/saml">  
482  
483   <md:AttributeAuthorityDescriptor  
484     protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
```

```

486     <md:AttributeService
487       x509qry:supportsX509Query="true"
488       xmlns:x509qry="urn:oasis:names:tc:SAML:metadata:X509:query"
489       Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
490       Location="https://idp.example.org:8443/saml-idp/AA"/>
491
492     <md:NameIDFormat>
493       urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
494     </md:NameIDFormat>
495
496     <!-- see [MACEAttr] -->
497     <md:AttributeProfile>
498       urn:mace:dir:profiles:attribute:samlv2
499     </md:AttributeProfile>
500
501   </md:AttributeAuthorityDescriptor>
502
503 </md:EntityDescriptor>

```

504 3.8.2 Service Provider Metadata

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

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

516 An example of service provider metadata follows:

```

517 <!-- A Service Provider supporting this profile -->
518 <md:EntityDescriptor
519   xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
520   entityID="https://sp.example.org/saml">
521
522   <md:RoleDescriptor
523     xmlns:query="urn:oasis:names:tc:SAML:metadata:ext:query"
524     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
525     xsi:type="query:AttributeQueryDescriptorType"
526     protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
527
528     <md:NameIDFormat>
529       urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
530     </md:NameIDFormat>
531
532     <md:AttributeConsumingService isDefault="true" index="0">
533       <md:ServiceName xml:lang="en">
534         Grid Service Provider
535       </md:ServiceName>
536       <md:RequestedAttribute
537         NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
538         Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
539         FriendlyName="eduPersonPrincipalName">
540       </md:RequestedAttribute>
541       <md:RequestedAttribute
542         NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
543         Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"

```

```
544     FriendlyName="eduPersonAffiliation">
545     </md:RequestedAttribute>
546     </md:AttributeConsumingService>
547
548     </md:RoleDescriptor>
549
550 </md:EntityDescriptor>
```

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

554 **3.9 Security and Privacy Considerations**

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

557 **3.9.1 Background**

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

563 **3.9.2 General Security Requirements**

564 SAML profiles often involve a system entity that relies on an earlier act of user authentication. For
565 example, the SAML Web Browser SSO Profile [SAMLProf] relies on an authentication service that
566 validates a username/password for a user. The authentication service must be securely linked to an
567 identity provider that issues SAML authentication assertions based on that user's act of authentication.
568 Similarly, this profile assumes that the system entity that performs the X.509 authentication is operating in
569 a secure environment that includes the attribute requester.

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

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

578 **3.9.3 User Privacy**

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

583 **3.10 Implementation Guidelines (non-normative)**

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

585 **3.10.1 Discovery**

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

588 Some possible approaches to identity provider discovery in the context of this profile are discussed briefly
589 below:

- 590 • The identity provider's unique identifier may be preconfigured at the service provider. This is useful,
591 for instance, if there is only one identity provider per deployment.
- 592 • The subject DN of the principal's X.509 certificate may include a reference to the identity provider.
593 New deployments are discouraged from decorating long-lived DNs in this manner, however, since
594 this practice may lessen interoperability with existing PKIs. For short-lived DNs, this practice may be
595 satisfactory.
- 596 • The issuer DN or the issuer alternative name may provide clues about the principal's preferred
597 identity provider. This technique may not be practical, however, since SAML authorities do not
598 typically issue X.509 credentials.
- 599 • A reference to the identity provider may be inserted into a non-critical X.509 extension [RFC3280] at
600 the time the credential is issued. For long-term credentials, this practice may not be feasible,
601 however. For short-term credentials, this technique may be satisfactory.

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

603 **3.10.2 Name Mapping**

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

609 **3.10.3 Canonicalization**

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

616 **3.10.4 Identity Provider Policy**

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

622 **3.10.5 Caching of Attributes**

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

625 4 SAML Attribute Self-Query Deployment Profile for 626 X.509 Subjects

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

631 4.1 Profile Overview (non-normative)

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

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

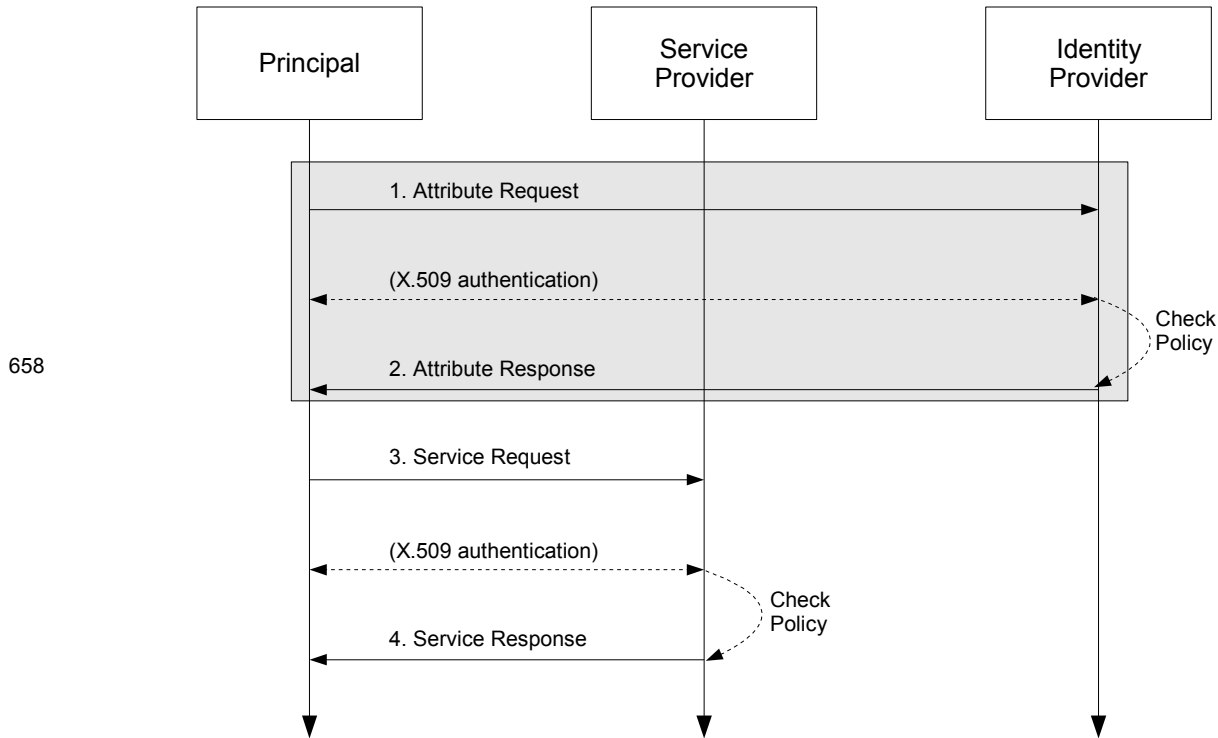
642 This use case is based on the following assumptions:

- 643 • A principal possesses an X.509 credential.
- 644 • The principal wields a client that can both query an identity provider for attributes and request a
645 service from a service provider.
- 646 • The client can access the principal's X.509 credential.
- 647 • The principal has an account with a SAML identity provider.
- 648 • The client knows the principal's preferred identity provider and the attribute requirements of the
649 target service provider.
- 650 • The identity provider is able to map an X.509 SAML Subject (as defined in section 2 of this
651 document) to one and only one principal in its security domain. In particular, the identity provider is
652 able to map the X.509 SAML Subject that represents this principal.

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

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

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



658

659 **1. Attribute Request**

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

665 **2. Attribute Response**

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

669 **3. Service Request**

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

674 **4. Service Response**

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

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

679 **4.2 Required Information**

680 **Identification:**

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

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

683 **Description:** Given below.

684 **Updates:** N/A

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

686 **4.3 Profile Description**

687 This profile extends the SAML Attribute Query Deployment Profile for X.509 Subjects described in
688 section 3.3.

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

- 692 • recognize the name identifier; and
- 693 • determine that the requester is the principal; and
- 694 • fulfill the request subject to any applicable policies;

695 the identity provider responds with a successful `<samlp:Response>` containing the relevant attributes for
696 the principal.

697 To determine that the requester is the principal, the identity provider MUST authenticate the principal. The
698 identity provider MAY choose to respond successfully to unauthenticated queries, but this is out of scope
699 for this profile.

700 **4.3.1 `<samlp:AttributeQuery>` Issued by Principal**

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

707 **4.3.2 `<samlp:Response>` Issued by Identity Provider**

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

709 **4.4 Use of SAML Request-Response Protocol**

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

714 **4.4.1 `<samlp:AttributeQuery>` Usage**

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

717 **4.4.2 `<samlp:Response>` Usage**

718 If the request is successful, the `<samlp:Response>` element MUST conform to the rules of section 3.4.2

719 except as noted below:

- 720 • The <saml:Subject> element MUST contain a <saml:SubjectConfirmation> element
721 whose Method attribute has value "urn:oasis:names:tc:SAML:2.0:cm:holder-of-key".
- 722 • A <saml:SubjectConfirmationData> element MUST be present and it MUST contain a
723 <ds:KeyInfo> element that refers to the principal's X.509 certificate.
- 724 • On the <saml:Conditions> element, the value of the NotBefore attribute (resp., the
725 NotOnOrAfter attribute) MUST be greater than or equal to (resp., less than or equal to) the
726 NotBefore field (resp., the NotOnOrAfter field) of the certificate.
- 727 • The <saml:Assertion> element MUST be signed.
- 728 • The <saml:Assertion> element MAY include a <saml:AuthnStatement> element.

729 4.4.3 Processing Rules

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

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

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

739 4.5 Example

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

```
741 <samlp:AttributeQuery
742   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
743   xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
744   ID="aaf23196-1773-2113-474a-fe114412ab72"
745   Version="2.0"
746   IssueInstant="2006-07-17T20:31:40Z">
747   <saml:Issuer
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:Issuer>
751   <saml:Subject>
752     <saml:NameID
753       Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">
754       C=US, O=NCSA-TEST, OU=User, CN=trscavo@uiuc.edu
755     </saml:NameID>
756   </saml:Subject>
757   <saml:Attribute
758     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
759     Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
760     FriendlyName="eduPersonPrincipalName">
761   </saml:Attribute>
762   <saml:Attribute
763     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
764     Name="urn:oid:2.5.4.42"
765     FriendlyName="givenName">
766   </saml:Attribute>
767   <saml:Attribute
768     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
769     Name="urn:oid:2.5.4.4"
```



```

835     </saml:AuthnContextClassRef>
836   </saml:AuthnContext>
837 </saml:AuthnStatement>
838 <saml:AttributeStatement>
839   <saml:Attribute
840     xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
841     x500:Encoding="LDAP"
842     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
843     Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
844     FriendlyName="eduPersonPrincipalName">
845     <saml:AttributeValue xsi:type="xs:string">
846       trscavo@uiuc.edu
847     </saml:AttributeValue>
848   </saml:Attribute>
849   <saml:Attribute
850     xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
851     x500:Encoding="LDAP"
852     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
853     Name="urn:oid:2.5.4.42"
854     FriendlyName="givenName">
855     <saml:AttributeValue xsi:type="xs:string">
856       Tom
857     </saml:AttributeValue>
858   </saml:Attribute>
859   <saml:Attribute
860     xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
861     x500:Encoding="LDAP"
862     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
863     Name="urn:oid:2.5.4.4"
864     FriendlyName="sn">
865     <saml:AttributeValue xsi:type="xs:string">
866       Scavo
867     </saml:AttributeValue>
868   </saml:Attribute>
869   <saml:Attribute
870     xmlns:x500="urn:oasis:names:tc:SAML:2.0:profiles:attribute:X500"
871     x500:Encoding="LDAP"
872     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
873     Name="urn:oid:1.3.6.1.4.1.1466.115.121.1.26"
874     FriendlyName="mail">
875     <saml:AttributeValue xsi:type="xs:string">
876       trscavo@gmail.com
877     </saml:AttributeValue>
878   </saml:Attribute>
879 </saml:AttributeStatement>
880 </saml:Assertion>

```

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

883 4.6 Use of Metadata

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

888 4.6.1 Identity Provider Metadata

889 An identity provider that uses SAML V2.0 metadata MUST include an
890 <md:AttributeAuthorityDescriptor> element that satisfies the rules given in section 3.8.1, except
891 that in this case the identity provider uses XML attribute supportsX509SelfQuery instead of

892 supportsX509Query [X509Query-XSD]:

```
893 <xs:attribute  
894   name="supportsX509SelfQuery" type="boolean" use="optional"/>
```

895 As before, use of this attribute is OPTIONAL.

896 An example of identity provider metadata follows:

```
897 <!-- An Identity Provider supporting both deployment profiles -->  
898 <md:EntityDescriptor  
899   xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"  
900   entityID="https://idp.example.org/saml">  
901  
902   <md:AttributeAuthorityDescriptor  
903     protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">  
904  
905     <md:AttributeService  
906       x509qry:supportsX509Query="true"  
907       x509qry:supportsX509SelfQuery="true"  
908       xmlns:x509qry="urn:oasis:names:tc:SAML:metadata:X509:query"  
909       Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"  
910       Location="https://idp.example.org:8443/saml-idp/AA"/>  
911  
912     <md:NameIDFormat>  
913       urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName  
914     </md:NameIDFormat>  
915  
916     <!-- see [MACEAttr] -->  
917     <md:AttributeProfile>  
918       urn:mace:dir:profiles:attribute:samlv2  
919     </md:AttributeProfile>  
920  
921   </md:AttributeAuthorityDescriptor>  
922  
923 </md:EntityDescriptor>
```

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

926 4.7 Security and Privacy Considerations

927 TBD

928 4.8 Implementation Guidelines (non-normative)

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

931 4.8.1 Discovery

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

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

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

941 5 References

942 5.1 Normative References

- 943 **[FIPS 140-2]** Security Requirements for Cryptographic Modules, May 2001. See
944 <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>
- 945 **[RFC 2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF
946 RFC 2119, March 1997. See <http://www.ietf.org/rfc/rfc2119.txt>
- 947 **[RFC2246]** T. Dierks and C. Allen. *The TLS Protocol Version 1.0*. IETF RFC 2246, January
948 1999. See <http://www.ietf.org/rfc/rfc2246.txt>
- 949 **[RFC2253]** M Wahl et al. *Lightweight Directory Access Protocol (v3): UTF-8 String*
950 *Representation of Distinguished Names*. IETF RFC 2253, December 1997. See
951 <http://www.ietf.org/rfc/rfc2253.txt>
- 952 **[RFC3280]** R. Housley et al. *Internet X.509 Public Key Infrastructure: Certificate and*
953 *Certificate Revocation List (CRL) Profile*. IETF RFC 3280, April 2002. See
954 <http://www.ietf.org/rfc/rfc3280.txt>
- 955 **[SAMLBind]** S. Cantor et al. *Bindings for the OASIS Security Assertion Markup Language*
956 *(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)
957 [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)
- 958 **[SAMLCore]** S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion*
959 *Markup Language (SAML) V2.0*. OASIS Standard, March 2005. See
960 <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>
- 961 **[SAMLMeta]** S. Cantor et al. *Metadata for the OASIS Security Assertion Markup Language*
962 *(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)
963 [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)
- 964 **[SAMLMeta-Ext]** T. Scavo and S. Cantor. *Metadata Extension for SAML V2.0 and V1.x Query*
965 *Requesters*. OASIS, September 2006. Document ID sstc-saml-metadata-ext-
966 query-cd-02. See [http://docs.oasis-open.org/security/saml/SpecDrafts-](http://docs.oasis-open.org/security/saml/SpecDrafts-Post2.0/sstc-saml-metadata-ext-query-cd-02.pdf)
967 [Post2.0/sstc-saml-metadata-ext-query-cd-02.pdf](http://docs.oasis-open.org/security/saml/SpecDrafts-Post2.0/sstc-saml-metadata-ext-query-cd-02.pdf)
- 968 **[SAMLProf]** S. Cantor et al. *Profiles for the OASIS Security Assertion Markup Language*
969 *(SAML) V2.0*. OASIS Standard, March 2005. See [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)
970 [open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)
- 971 **[Schema1]** H. S. Thompson et al. *XML Schema Part 1: Structures*. World Wide Web
972 Consortium Recommendation, May 2001. See [http://www.w3.org/TR/2001/REC-](http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/)
973 [xmlschema-1-20010502/](http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/)
- 974 **[SSL3]** A. Freier et al. *The SSL Protocol Version 3.0*, IETF Internet-Draft, November
975 1996. See <http://wp.netscape.com/eng/ssl3/draft302.txt>
- 976 **[X509Query-XSD]** *Schema for SAML V2.0 Deployment Profiles for X.509 Subjects*. OASIS,
977 December 2006. Document ID sstc-saml-metadata-x509-query.xsd. See
978 http://www.oasis-open.org/committees/documents.php?wg_abbrev=security
- 979 **[XMLEnc]** D. Eastlake et al. *XML Encryption Syntax and Processing*. World Wide Web
980 Consortium Recommendation, December 2002. See
981 <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>
- 982 **[XMLEnc-XSD]** *XML Encryption Schema*. World Wide Web Consortium Recommendation,
983 December 2002. See [http://www.w3.org/TR/2002/REC-xmlenc-core-](http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-schema.xsd)
984 [20021210/xenc-schema.xsd](http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/xenc-schema.xsd)
- 985 **[XMLSig]** D. Eastlake et al. *XML-Signature Syntax and Processing*. World Wide Web
986 Consortium Recommendation, February 2002. See
987 <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/>
- 988 **[XMLSig-XSD]** *Schema for XML Signatures*. World Wide Web Consortium Recommendation,

989 February 2002. See [http://www.w3.org/TR/2002/REC-xmlsig-core-](http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/xmlsig-core-schema.xsd)
990 [20020212/xmlsig-core-schema.xsd](http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/xmlsig-core-schema.xsd)
991

992 **5.2 Non-Normative References**

- 993 **[MACEAttrib]** S. Cantor et al. *MACE-Dir SAML Attribute Profiles*. Internet2 MACE, April 2006.
994 See [http://middleware.internet2.edu/dir/docs/internet2-mace-dir-saml-attributes-](http://middleware.internet2.edu/dir/docs/internet2-mace-dir-saml-attributes-200604.pdf)
995 [200604.pdf](http://middleware.internet2.edu/dir/docs/internet2-mace-dir-saml-attributes-200604.pdf)
- 996 **[RFC3820]** S. Tuecke et al. *Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate*
997 *Profile*. IETF RFC 3820, June 2004. See <http://www.ietf.org/rfc/rfc3820.txt>
- 998 **[SAMLGloss]** J. Hodges et al. *Glossary for the OASIS Security Assertion Markup Language*
999 *(SAML) V2.0*. OASIS Standard, March 2005. See [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf)
1000 [open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf)
- 1001 **[SAMLSecure]** F. Hirsch et al. *Security and Privacy Considerations for the OASIS Security*
1002 *Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. See
1003 <http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf>

1004 **6 Revision History**

1005 TBA

<i>Document ID</i>	<i>Date</i>	<i>Committer</i>	<i>Comment</i>
Draft-01	18 Dec 2006	T. Scavo	Initial draft.

1006

1008 **A. Acknowledgments**

1009 The editors would like to acknowledge the contributions of the OASIS Security Services Technical
1010 Committee, whose voting members at the time of publication were:
1011

- 1012 • Christopher Laskowski, Booz Allen Hamilton
- 1013 • Rebekah Metz, Booz Allen Hamilton
- 1014 • Hal Lockhart, BEA Systems, Inc.
- 1015 • Steve Anderson, BMC Software
- 1016 • Sharon Boeyen, Entrust
- 1017 • Thomas Wisniewski, Entrust
- 1018 • Carolina Canales-Valenzuela, Ericsson
- 1019 • Dana Kaufman, Forum Systems, Inc.
- 1020 • Ashish Patel, France Telecom
- 1021 • Greg Whitehead, Hewlett-Packard
- 1022 • Guy Denton, IBM
- 1023 • Heather Hinton, IBM
- 1024 • Anthony Nadalin, IBM
- 1025 • Eric Tiffany, IEEE Industry Standards and Technology Org (IEEE-ISTO)
- 1026 • Scott Cantor, Internet2
- 1027 • Bob Morgan, Internet2
- 1028 • Tom Scavo, National Center for Supercomputing Applications (NCSA)
- 1029 • Peter Davis, Neustar, Inc.
- 1030 • Jeff Hodges, Neustar, Inc.
- 1031 • Frederick Hirsch, Nokia Corporation
- 1032 • Abbie Barbir, Nortel Networks Limited
- 1033 • Paul Madsen, NTT Corporation
- 1034 • Ari Kermaier, Oracle Corporation
- 1035 • Prateek Mishra, Oracle Corporation
- 1036 • Brian Campbell, Ping Identity Corporation
- 1037 • Rob Philpott, RSA Security
- 1038 • Jahan Moreh, Sigaba Corp.
- 1039 • Bhavna Bhatnagar, Sun Microsystems
- 1040 • Eve Maler, Sun Microsystems
- 1041 • Emily Xu, Sun Microsystems
- 1042 • David Staggs, Veterans Health Administration

1043 **Appendix B. Notices**

1044 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
1045 might be claimed to pertain to the implementation or use of the technology described in this document or
1046 the extent to which any license under such rights might or might not be available; neither does it represent
1047 that it has made any effort to identify any such rights. Information on OASIS's procedures with respect to
1048 rights in OASIS specifications can be found at the OASIS website. Copies of claims of rights made
1049 available for publication and any assurances of licenses to be made available, or the result of an attempt
1050 made to obtain a general license or permission for the use of such proprietary rights by implementors or
1051 users of this specification, can be obtained from the OASIS Executive Director.

1052 OASIS invites any interested party to bring to its attention any copyrights, patents or patent applications, or
1053 other proprietary rights which may cover technology that may be required to implement this specification.
1054 Please address the information to the OASIS Executive Director.

1055 **Copyright © OASIS Open 2006. All Rights Reserved.**

1056 This document and translations of it may be copied and furnished to others, and derivative works that
1057 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and
1058 distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and
1059 this paragraph are included on all such copies and derivative works. However, this document itself may
1060 not be modified in any way, such as by removing the copyright notice or references to OASIS, except as
1061 needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights
1062 defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it
1063 into languages other than English.

1064 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
1065 or assigns.

1066 This document and the information contained herein is provided on an "AS IS" basis and OASIS
1067 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
1068 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR
1069 ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.