



SAML V2.0 Errata

Approved Errata Committee Draft 04 20 October 2009

Specification URIs:

This Version:

<http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0-cd-04.html>
<http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0-cd-04.odt>
<http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0-cd-04.pdf>

Previous Version:

<http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0-cd-03.html>
<http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0-cd-03.odt>
<http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0-cd-03.pdf>

Latest Version:

<http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.html>
<http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.odt>
<http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf>

Technical Committee:

OASIS Security Services TC

Chair(s):

Hal Lockhart, Oracle Corporation
Thomas Hardjono, M.I.T.

Editor:

Scott Cantor, Internet2

Related Work:

<http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf>
<http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf>
<http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf>
<http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>
<http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf>
<http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf>
<http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf>

Abstract:

This document lists approved errata to the SAML V2.0 OASIS Standard.

Status:

This document was last revised or approved by the SSTC on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

39 TC members should send comments on this specification to the TC's email list.
40 Others should send comments to the TC by using the "Send A Comment" button on
41 the TC's web page at <http://www.oasis-open.org/committees/security>.
42 For information on whether any patents have been disclosed that may be essential to
43 implementing this specification, and any offers of patent licensing terms, please refer to the IPR
44 section of the TC web page (<http://www.oasis-open.org/committees/security/ipr.php>).
45 The non-normative errata page for this specification is located at [http://www.oasis-
open.org/committees/security](http://www.oasis-
46 open.org/committees/security).

47 Notices

48 Copyright © OASIS® 1993–2009. All Rights Reserved. OASIS trademark, IPR and other policies apply.
49 All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual
50 Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.
51 This document and translations of it may be copied and furnished to others, and derivative works that
52 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published,
53 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice
54 and this section are included on all such copies and derivative works. However, this document itself may
55 not be modified in any way, including by removing the copyright notice or references to OASIS, except as
56 needed for the purpose of developing any document or deliverable produced by an OASIS Technical
57 Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must
58 be followed) or as required to translate it into languages other than English.
59 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
60 or assigns.
61 This document and the information contained herein is provided on an "AS IS" basis and OASIS
62 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
63 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
64 OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
65 PARTICULAR PURPOSE.
66 OASIS requests that any OASIS Party or any other party that believes it has patent claims that would
67 necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard,
68 to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to
69 such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that
70 produced this specification.
71 OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of
72 any patent claims that would necessarily be infringed by implementations of this specification by a patent
73 holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR
74 Mode of the OASIS Technical Committee that produced this specification. OASIS may include such
75 claims on its website, but disclaims any obligation to do so.
76 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
77 might be claimed to pertain to the implementation or use of the technology described in this document or
78 the extent to which any license under such rights might or might not be available; neither does it represent
79 that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to
80 rights in any document or deliverable produced by an OASIS Technical Committee can be found on the
81 OASIS website. Copies of claims of rights made available for publication and any assurances of licenses
82 to be made available, or the result of an attempt made to obtain a general license or permission for the
83 use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS
84 Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any
85 information or list of intellectual property rights will at any time be complete, or that any claims in such list
86 are, in fact, Essential Claims.
87 The name "OASIS" is a trademark of [OASIS](#), the owner and developer of this specification, and should be
88 used only to refer to the organization and its official outputs. OASIS welcomes reference to, and
89 implementation and use of, specifications, while reserving the right to enforce its marks against
90 misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

92	1 Introduction.....	6
93	1.1 Normative References.....	6
94	2 Approved Errata.....	8
95	E0: Incorrect Section Reference.....	8
96	E1: Relay State for HTTP Redirect.....	8
97	E2: Metadata Clarifications for HTTP Artifact Binding.....	8
98	E4: No Role for SAML V1.1 Artifacts in SAML V2.0.....	8
99	E6: Clarify Constraints on Encrypted NameID.....	9
100	E7: Metadata for Agreeing to Sign Authentication Requests.....	9
101	E8: SLO and NameID Termination	9
102	E10: Logout Request Reason Mismatch with Schema	10
103	E11: Improperly Labeled Feature.....	10
104	E12: Clarification on ManageNameIDRequest.....	10
105	E13: Inaccurate Description of Authorization Decision	11
106	E14: AllowCreate.....	11
107	E15: NameID Policy Adherence.....	13
108	E17: Authentication Response IssuerName vs. Assertion IssuerName.....	13
109	E18: Reference to Identity Provider Discovery Service in ECP Profile.....	14
110	E19: Clarification on Error Processing.....	14
111	E20: ECP SSO Profile and Metadata.....	14
112	E21: PAOS Version.....	15
113	E22: Error in Profile/ECP.....	15
114	E24: HTTPS in URI Binding.....	15
115	E25: Metadata Feature in Conformance.....	15
116	E26: Ambiguities Around Multiple Assertions and Statements in the SSO Profile.....	16
117	E27: Incorrect Step Number in ECP Profile.....	19
118	E28: Profile Labeling in Conformance.....	19
119	E29: Incomplete Listing of Features in Conformance.....	19
120	E30: Key Replacement.....	19
121	E31: Various Minor Errors in Binding.....	19
122	E32: Missing Required Information in Profiles.....	20
123	E33: References to Assertion Request Protocol.....	20
124	E34: RequestedAttribute Section Heading.....	20
125	E35: Response Consumer URL Rules and Example.....	20
126	E36: Clarification on Action Element.....	21
127	E37: Clarification in Metadata on Indexed Endpoints.....	21
128	E38: Clarification Regarding Index on <LogoutRequest>.....	21
129	E39: Error in SAML Profile Example.....	22
130	E40: Holder of Key.....	22
131	E41: EndpointType ResponseLocation Clarification in Metadata.....	22
132	E42: Match Authorities to Queries in Conformance.....	23

133	E43: Key Location in saml:EncryptedData.....	23
134	E45: AuthnContext Comparison Order.....	26
135	E46: AudienceRestriction Clarifications.....	26
136	E47: Clarification on SubjectConfirmation.....	27
137	E48: Clarification on Encoding for Binary Values in LDAP Profile.....	28
138	E49: Clarification on Attribute Name Format	28
139	E50: Clarification on SSL Ciphersuites	28
140	E51: Schema Type of Contents of <AttributeValue>	29
141	E52: Clarification on NotOnOrAfter Attribute for Subject Confirmation.....	29
142	E53: Correction to LDAP/X.500 Profile Attribute.....	29
143	E54: Corrections to ECP URN	29
144	E55: Language Cleanup Around Name Identifier Management.....	30
145	E56: Confirmation Method Typo.....	31
146	E57: SAMLmime Reference.....	31
147	E58: KeyDescriptor Typos in Profiles.....	32
148	E59: SSO Response When Using HTTP-Artifact.....	32
149	E60: Incorrect URI for Unspecified NameID Format.....	32
150	E61: Reference to Non-Existent Element.....	32
151	E62: TLS Keys in KeyDescriptor.....	33
152	E63: IdP Discovery Cookie Interpretation.....	33
153	E64: Liberty Moniker Used Inappropriately.....	33
154	E65: Second-level StatusCode.....	33
155	E66: Metadata and DNSSEC.....	34
156	E68: Use of Multiple <KeyDescriptor> Elements.....	34
157	E69: Semantics of <ds:KeyInfo> in <KeyDescriptor>.....	35
158	E70: Obsolete reference to UUID URN namespace.....	35
159	E71: Missing namespace definition in Profiles.....	35
160	E74: Update XML Signature Reference.....	35
161	E75: Clarify Handling of SubjectConfirmation in AuthnRequest.....	36
162	E76: Clarify nested validUntil/cacheDuration.....	36
163	E77: Generalize scope of Metadata specification.....	36
164	E78: Reassignment of persistent identifiers.....	37
165	E79: Clarification of SessionNotOnOrAfter.....	37
166	E81: Algorithm statement in XML Signature profile.....	37
167	E82: Empty <ContactPerson> element.....	37
168	E83: Weaken claim made about Exclusive C14N.....	37
169	3 Acknowledgments.....	39
170		

1 Introduction

171

172 This document lists the approved errata to the SAML V2.0 OASIS Standard. Each one has been given an
173 *Err* designation. Numbers in the sequence are missing wherever a reported problem (a “proposed
174 erratum”, or PE) resulted in a TC decision not to issue an erratum to any V2.0 specification text, or where
175 an issue has not yet been disposed.

176 This document is ultimately intended to be confirmed as a formal Approved Errata document. To see the
177 full list of reported problems and additional background on the approved errata, see the Errata Working
178 Document for SAML V2.0 [SAMLErrWork].

179 As required by the OASIS Technical Committee Process, the approved errata represent changes that are
180 not “substantive”. The changes focus on clarifications to ambiguous or conflicting specification text, where
181 different compliant implementations might have reasonably chosen different interpretations. The intent of
182 the Security Services TC has been to resolve such issues in service of improved interoperability based on
183 implementation and deployment experience.

184 In this document, errata change instructions are presented with surrounding context as necessary to
185 make the intent clear. Original specification text is often presented as follows, with problem text
186 highlighted in bold:

187 This is an original specification sentence. **The second sentence needs to be changed, removed, or**
188 **replaced.**

189 New specification text is typically presented as follows, with new or changed text highlighted in bold:

190 This is a **highly** original specification sentence. **This is the wholly new content to replace the old second**
191 **sentence. It runs on and on and on.**

192 In a few cases, text needs only to be struck, in which case the change is shown as follows, with text to be
193 removed both highlighted in bold and struck through:

194 This is yet another original specification sentence which contains ~~an inappropriately~~ long description.

195 In addition to this normative document, non-normative “errata composite” documents may be provided
196 that combine the prescribed corrections with the original specification text, illustrating the changes with
197 margin change bars, struck-through original text, and highlighted new text. These documents, if available,
198 will be found at the same location as this approved form.

199 All cited line numbers refer to the PDF forms of the original OASIS Standard specifications in question,
200 not to line numbers in this document or in the errata composite documents.

201 1.1 Normative References

202 In general, the latest revisions of all errata-related documents will be linked from the TC home page at
203 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security.

204 **[SAMLAuthCtx]** J. Kemp et al. *Authentication Context for the OASIS Security Assertion Markup*
205 *Language (SAML) V2.0*. OASIS SSTC, March 2005. See [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf)
206 [open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf).

207 **[SAMLBind]** S. Cantor et al. *Bindings for the OASIS Security Assertion Markup Language*
208 *(SAML) V2.0*. OASIS SSTC, March 2005. See [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf)
209 [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).

210 **[SAMLConf]** P. Mishra et al. *Conformance Requirements for the OASIS Security Assertion*
211 *Mark Markup Language (SAML) V2.0*. OASIS SSTC, March 2005. See
212 <http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf>.

213 **[SAMLCore]** S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion*
214 *Markup Language (SAML) V2.0*. OASIS SSTC, March 2005. See
215 <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>.

216 **[SAMLErrWork]** S Cantor. *Errata Working Document for SAML V2.0*. OASIS SSTC, October
217 2009. Revision 51 corresponds to this Working Draft; see [http://www.oasis-
open.org/committees/download.php/34737/sstc-saml-errata-2.0-draft-51.pdf](http://www.oasis-
218 open.org/committees/download.php/34737/sstc-saml-errata-2.0-draft-51.pdf).
219 **[SAMLMeta]** S. Cantor et al. *Metadata for the OASIS Security Assertion Markup Language
220 (SAML) V2.0*. OASIS SSTC, March 2005. See [http://docs.oasis-
open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf](http://docs.oasis-
221 open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf).
222 **[SAMLProf]** J. Hughes et al. *Profiles for the OASIS Security Assertion Markup Language
223 (SAML) V2.0*. OASIS SSTC, March 2005. See [http://docs.oasis-
open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf](http://docs.oasis-
224 open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf).
225 **[SAMLSec]** F. Hirsch et al. *Security Considerations for the OASIS Security Assertion Markup
226 Language (SAML) V2.0*. OASIS SSTC, March 2005. See [http://docs.oasis-
open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf](http://docs.oasis-
227 open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf).

2 Approved Errata

228

229 Following are the approved errata to the SAML V2.0 OASIS Standard.

230

E0: Incorrect Section Reference

231 Change [SAMLCore] at line 2660 to refer to section **3.7.3** rather than **3.6.3** for `Reason` codes. This was a
232 typographical error.

233

E1: Relay State for HTTP Redirect

234 Change [SAMLBind] Section 3.4.3 at lines 551-553 to reflect the fact that, indeed, the `RelayState`
235 parameter is covered by the query string signature described in Section 3.4.4.1 (DEFLATE encoding).
236 Note that Section 3.5.3, which has similar original wording, remains correct for its case.

237 Original:

238 RelayState data MAY be included with a SAML protocol message transmitted with this binding. The value
239 MUST NOT exceed 80 bytes in length and SHOULD be integrity protected by the entity creating the
240 message. **Signing is not realistic given the space limitation, but because the value is exposed to
241 third-party tampering, the entity SHOULD insure that the value has not been tampered with by using
242 a checksum, a pseudo-random value, or similar means.**

243 New:

244 RelayState data MAY be included with a SAML protocol message transmitted with this binding. The value
245 MUST NOT exceed 80 bytes in length and SHOULD be integrity protected by the entity creating the
246 message, **either via a digital signature (see Section 3.4.4.1) or by some independent means.**

247

E2: Metadata Clarifications for HTTP Artifact Binding

248 Change [SAMLBind] Section 3.6.7 at lines 1188-1191 to clarify metadata requirements on profiles using
249 the HTTP Artifact binding.

250 Original:

251 Support for the HTTP Artifact binding SHOULD be reflected by indicating URL endpoints at which requests
252 and responses for a particular protocol or profile should be sent. Either a single endpoint or distinct request
253 and response endpoints MAY be supplied. **One or more indexed endpoints for processing
254 <samlp:ArtifactResolve> messages SHOULD also be described.**

255 New:

256 Support for **receiving messages using** the HTTP Artifact binding SHOULD be reflected by indicating URL
257 endpoints at which requests and responses for a particular protocol or profile should be sent. **Support for
258 sending messages using this binding SHOULD be accompanied by one or more indexed
259 <md:ArtifactResolutionService> endpoints for processing <samlp:ArtifactResolve> messages.**

260

E4: No Role for SAML V1.1 Artifacts in SAML V2.0

261 Change [SAMLBind] Section 3.6.4 at line 1067 to clarify that SAML V1.1 artifacts have no role in SAML
262 V2.0.

263 New:

264 The following describes the single artifact type defined by SAML V2.0. **Although the general artifact
265 structure resembles that used in prior versions of SAML and the type code of the single format
266 described below does not conflict with previously defined formats, there is explicitly no
267 correspondence between SAML V2.0 artifacts and those found in any previous specifications, and
268 artifact formats not defined specifically for use with SAML V2.0 MUST NOT be used with this
269 binding.**

E6: Clarify Constraints on Encrypted NameID

270

271 Change [SAMLCore] Section 3.4.1.1 at line 2139 to clarify that, if encrypted name identifiers are chosen,
272 no further description of the type of name identifier will be available in SAML messages..

273 New:

274 The special Format value `urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted` indicates
275 that the resulting assertion(s) MUST contain `<EncryptedID>` elements instead of plaintext. The underlying
276 name identifier's unencrypted form can be of any type supported by the identity provider for the requested
277 subject. **It is not possible for the service provider to specifically request that a particular kind of
278 identifier be returned if it asks for encryption. The `<md:NameIDFormat>` metadata element (see
279 [SAMLMeta]) or other out-of-band means MAY be used to determine what kind of identifier to
280 encrypt and return.**

E7: Metadata for Agreeing to Sign Authentication Requests

281

282 Change [SAMLMeta] Section 2.4.3 at line 710, 741-742, and 744-747 to remove ambiguity about how to
283 accomplish signing when the IdP SSO descriptor includes the setting `WantAuthnRequestsSigned` and the
284 SP SSO descriptor includes the setting `AuthnRequestsSigned`. .

285 New at line 710:

286 **The `WantAuthnRequestsSigned` attribute is intended to indicate to service providers whether or not
287 they can expect an unsigned `<AuthnRequest>` message to be accepted by the identity provider. The
288 identity provider is not obligated to reject unsigned requests nor is a service provider obligated to
289 sign its requests, although it might reasonably expect an unsigned request will be rejected. In some
290 cases, a service provider may not even know which identity provider will ultimately receive and
291 respond to its requests, so the use of this attribute in such a case cannot be strictly defined.**

292

293 **Furthermore, note that the specific method of signing that would be expected is binding dependent.
294 The HTTP Redirect binding (see [SAMLBind]) requires that the signature be applied to the URL-
295 encoded value rather than placed within the XML message, while other bindings generally permit the
296 signature to be within the message in the usual fashion.**

297

298 The following schema fragment defines the `<IDPSSODescriptor>` element and its
299 `IDPSSODescriptorType` complex type:

300 New at lines 741-742:

301 **Optional attribute that indicates whether the `<samlp:AuthnRequest>` messages sent by this service
302 provider will be signed. If omitted, the value is assumed to be false. **A value of false (or omission of this
303 attribute) does not imply that the service provider will never sign its requests or that a signed
304 request should be considered an error. However, an identity provider that receives an unsigned
305 `<samlp:AuthnRequest>` message from a service provider whose metadata contains this attribute
306 with a value of true MUST return a SAML error response and MUST NOT fulfill the request.****

307 New at lines 744-747:

308 **Optional attribute that indicates a requirement for the `<saml:Assertion>` elements received by this
309 service provider to be signed. If omitted, the value is assumed to be false. This requirement is in addition to
310 any requirement for signing derived from the use of a particular profile/binding combination. **Note that an
311 enclosing signature at the SAML binding or protocol layer does not suffice to meet this requirement,
312 for example signing a `<samlp:Response>` containing the assertion(s) or a TLS connection.****

E8: SLO and NameID Termination

313

314 Change [SAMLCore] Section 3.6.3 at lines 2479-2480 to clarify the rules around SP single logout
315 behavior when a name identifier has been terminated.

316 Original:

317 The receiving provider can perform any maintenance with the knowledge that the relationship represented
318 by the name identifier has been terminated. **It can choose to invalidate the active session(s) of a**
319 **principal for whom a relationship has been terminated.**

320 New:

321 The receiving provider can perform any maintenance with the knowledge that the relationship represented
322 by the name identifier has been terminated. **In general it SHOULD NOT invalidate any active session(s)**
323 **of the principal for whom the relationship has been terminated. If the receiving provider is an identity**
324 **provider, it SHOULD NOT invalidate any active session(s) of the principal established with other**
325 **service providers. A requesting provider MAY send a <LogoutRequest> message prior to initiating**
326 **a name identifier termination by sending a <ManageNameIDRequest> message if that is the**
327 **requesting provider's intent (e.g., the name identifier termination is initiated via an administrator**
328 **who wished to terminate all user activity). The requesting provider MUST NOT send a**
329 **<LogoutRequest> message after the <ManageNameIDRequest> message is sent.**

330 **E10: Logout Request Reason Mismatch with Schema**

331 Change [SAMLCore] Section 3.7.1 at line 2540 to resolve an apparent conflict between the specification
332 text and the schema. (Note that although in this case the schema could have been more specific, text in
333 SAML specifications is allowed to impose further restrictions on syntactic constraints imposed by a
334 schema, and this technique has been used here to resolve the issue without a substantive change.)

335 New:

336 An indication of the reason for the logout, in the form of a URI reference. **The Reason attribute is specified**
337 **as a string in the schema. This specification further restricts the schema by requiring that the**
338 **Reason attribute MUST be in the form of a URI reference.**

339 **E11: Improperly Labeled Feature**

340 Change [SAMLConf] in Section 3.2 (Table 2) to make the labels in feature rows 6 through 9 consistent.

341 Original labels:

342 Name Identifier Management, HTTP Redirect (IdP-initiated)
343 Name Identifier Management, SOAP (IdP-initiated)
344 Name Identifier Management, HTTP Redirect
345 Name Identifier Management, SOAP

346 New labels:

347 **Name Identifier Management (IdP-Initiated), HTTP Redirect**
348 **Name Identifier Management (IdP-Initiated), SOAP**
349 **Name Identifier Management (SP-Initiated), HTTP Redirect**
350 **Name Identifier Management (SP-Initiated), SOAP**

351 **E12: Clarification on ManageNameIDRequest**

352 Change [SAMLCore] Section 3.6 at lines 2412-2413 and 2438, and change [SAMLProf] Section 4.5 at
353 lines 1320-1321, to remove incorrect implications that the name identifier format can be changed in the
354 course of the protocol.

355 New [SAMLCore] at lines 2412-2413:

356 After establishing a name identifier for a principal, an identity provider wishing to change the value **and/or-**
357 **format** of the identifier that it will use when referring to the principal, or to indicate that a name identifier will
358 no longer be used to refer to the principal, informs service providers of the change by sending them a
359 **<ManageNameIDRequest> message.**

360 New [SAMLCore] at line 2438:

361 If the requester is the identity provider, the new value will appear in subsequent <NameID> elements as the
362 element's content. **In either case, if the <NewEncryptedID> is used, its encrypted content is just a**
363 **<NewID> element containing only the new value for the identifier (format and qualifiers cannot be**
364 **changed once established).**

365 New [SAMLProf] at lines 1320-23121:

366 Subsequently, the identity provider may wish to notify the service provider of a change in the **format and/or**
367 **value that it will use to identify the same principal in the future.**

368 **E13: Inaccurate Description of Authorization Decision**

369 Change [SAMLCore] Section 2 at lines 357-358 to complete the list of potential results from an
370 authorization decision.

371 New:

372 Authorization Decision: A request to allow the assertion subject to access the specified resource has been
373 granted or denied **or is indeterminate.**

374 **E14: AllowCreate**

375 Change [SAMLCore] at lines 2123-2129, 2130, 2143-2147, 2419-2420, and 2480, and change
376 [SAMLProf] at lines 521-524, to clarify the semantics of AllowCreate.

377 Original at [SAMLCore] Section 3.4.1.1, lines 2123-2129:

378 A Boolean value used to indicate whether the identity provider **is allowed**, in the course of fulfilling the
379 request, to create a new identifier **to represent the principal**. Defaults to "false". **When "false", the**
380 **requester constrains the identity provider to only issue an assertion to it if an acceptable identifier**
381 **for the principal has already been established. Note that this does not prevent the identity provider**
382 **from creating such identifiers outside the context of this specific request (for example, in advance**
383 **for a large number of principals).**

384 New at [SAMLCore] Section 3.4.1.1, lines 2123-2129:

385 A Boolean value used to indicate whether the **requester grants to** the identity provider, in the course of
386 fulfilling the request, **permission to create a new identifier or to associate an existing identifier**
387 **representing the principal with the relying party**. Defaults to "false" if not present or the entire element
388 **is omitted.**

389 New at [SAMLCore] Section 3.4.1.1, line 2130 (just after the above changes):

390 **The AllowCreate attribute may be used by some deployments to influence the creation of state**
391 **maintained by the identity provider pertaining to the use of a name identifier (or any other persistent,**
392 **uniquely identifying attributes) by a particular relying party, for purposes such as dynamic identifier**
393 **or attribute creation, tracking of consent, subsequent use of the Name Identifier Management**
394 **protocol (see Section 3.6), or other related purposes.**

395
396 **When "false", the requester tries to constrain the identity provider to issue an assertion only if such**
397 **state has already been established or is not deemed applicable by the identity provider to the use of**
398 **an identifier. Thus, this does not prevent the identity provider from assuming such information**
399 **exists outside the context of this specific request (for example, establishing it in advance for a large**
400 **number of principals).**

401
402 **A value of "true" permits the identity provider to take any related actions it wishes to fulfill the**
403 **request, subject to any other constraints imposed by the request and policy (the IsPassive**
404 **attribute, for example).**

405
406 **Generally, requesters cannot assume specific behavior from identity providers regarding the initial**
407 **creation or association of identifiers on their behalf, as these are details left to implementations or**
408 **deployments. Absent specific profiles governing the use of this attribute, it might be used as a hint**
409 **to identity providers about the requester's intention to store the identifier or link it to a local value.**

410
411
412
413
414
415
416
417
418
419
420

A value of “false” might be used to indicate that the requester is not prepared or able to do so and save the identity provider wasted effort.

Requesters that do not make specific use of this attribute SHOULD generally set it to “true” to maximize interoperability.

The use of the AllowCreate attribute MUST NOT be used and SHOULD be ignored in conjunction with requests for or assertions issued with name identifiers with a Format of urn:oasis:names:tc:SAML:2.0:nameid-format:transient (they preclude any such state in and of themselves).

421 Original at [SAMLCore] Section 3.6, lines 2419-2420:

422
423
424
425
426
427

A service provider also uses this message to register or change the SPProvidedID value to be included when the underlying name identifier is used to communicate with it, or to terminate the use of a name identifier between itself and the identity provider.

Note that this protocol is typically not used with “transient” name identifiers, since their value is not intended to be managed on a long-term basis.

428 New at [SAMLCore] Section 3.6, lines 2419-2420:

429
430
431
432
433
434

A service provider also uses this message to register or change the SPProvidedID value to be included when the underlying name identifier is used to communicate with it, or to terminate the use of a name identifier between itself and the identity provider.

This protocol MUST NOT be used in conjunction with the urn:oasis:names:tc:SAML:2.0:nameidformat:transient <NameID> Format.

435 New at [SAMLCore] Section 3.6.3, line 2480 (note that E8 and E55 specify additional changes to the
436 original text shown here):

437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463

If the <Terminate> element is included in the request, the requesting provider is indicating that (in the case of a service provider) it will no longer accept assertions from the identity provider or (in the case of an identity provider) it will no longer issue assertions to the service provider about the principal. The receiving provider can perform any maintenance with the knowledge that the relationship represented by the name identifier has been terminated. It can choose to invalidate the active session(s) of a principal for whom a relationship has been terminated.

If the receiving provider is maintaining state associated with the name identifier, such as the value of the identifier itself (in the case of a pair-wise identifier), an SPProvidedID value, the sender’s consent to the identifier’s creation/use, etc., then the receiver can perform any maintenance with the knowledge that the relationship represented by the name identifier has been terminated.

Any subsequent operations performed by the receiver on behalf of the sender regarding the principal (for example, a subsequent <AuthnRequest>) SHOULD be carried out in a manner consistent with the absence of any previous state.

Termination is potentially the cleanup step for any state management behavior triggered by the use of the AllowCreate attribute in the Authentication Request protocol (see Section 3.4). Deployments that do not make use of that attribute are likely to avoid the use of the <Terminate> element or would treat it as a purely advisory matter.

Note that in most cases (a notable exception being the rules surrounding the SPProvidedID attribute), there are no requirements on either identity providers or service providers regarding the creation or use of persistent state. Therefore, no explicit behavior is mandated when the <Terminate> element is received. However, if persistent state is present pertaining to the use of an identifier (such as if an SPProvidedID attribute was attached), the <Terminate> element provides a clear indication that this state SHOULD be deleted (or marked as obsolete in some fashion).

464 Original at [SAMLProf] Section 4.1.4.1, lines 521-524:

465 If the identity provider cannot or will not satisfy the request, it MUST respond with a <Response> message
466 containing an appropriate error status code or codes.

467
468 **If the service provider wishes to permit the identity provider to establish a new identifier for the**
469 **principal if none exists, it MUST include a <NameIDPolicy> element with the AllowCreate attribute**
470 **set to "true". Otherwise, only a principal for whom the identity provider has previously established**
471 **an identifier usable by the service provider can be authenticated successfully.**

472 New at [SAMLProf] Section 4.1.4.1, lines 521-524:

473 If the identity provider cannot or will not satisfy the request, it MUST respond with a <Response> message
474 containing an appropriate error status code or codes.

475
476 **This profile does not provide any guidelines for the use of AllowCreate; see [SAMLCore] for**
477 **normative rules on using AllowCreate.**

478 **E15: NameID Policy Adherence**

479 Change [SAMLCore] Section 3.4.1.1 at line 2139 to clarify that the expressed name identifier policy must
480 be adhered to.

481 New (note that E6 specifies additional changes to the original text shown here):

482 The special Format value urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted indicates
483 that the resulting assertion(s) MUST contain <EncryptedID> elements instead of plaintext. The underlying
484 name identifier's unencrypted form can be of any type supported by the identity provider for the requested
485 subject.

486
487 **When a Format defined in Section Error: Reference source not found8.3 other than**
488 **urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified Or**
489 **urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted is used, then if the identity provider**
490 **returns any assertions:**

- 491
492 ● the Format value of the <NameID> within the <Subject> of any <Assertion> MUST be identical
493 to the Format value supplied in the <NameIDPolicy>, and
494
495 ● if SPNameQualifier is not omitted in <NameIDPolicy>, the SPNameQualifier value of the
496 <NameID> within the <Subject> of any <Assertion> MUST be identical to the SPNameQualifier
497 value supplied in the <NameIDPolicy>.

498 **E17: Authentication Response IssuerName vs. Assertion** 499 **IssuerName**

500 Change [SAMLProf] Section 4.1.4.2 at lines 541-543 to accurately reflect the conditions under which
501 issuer information is required and how issuer information at the different levels must correlate.

502 Original:

503 **The <Issuer> element MAY be omitted, but if present it MUST contain the unique identifier of the**
504 **issuing identity provider; the Format attribute MUST be omitted or have a value of**
505 **urn:oasis:names:tc:SAML:2.0:nameid-format:entity.**

506 New:

507 **If the <Response> message is signed or if an enclosed assertion is encrypted, then the <Issuer>**
508 **element MUST be present. Otherwise it MAY be omitted. If present it MUST contain the unique identifier**
509 **of the issuing identity provider; the Format attribute MUST be omitted or have a value of**
510 **urn:oasis:names:tc:SAML:2.0:nameid-format:entity.**

511
512
513
514
515
516
517
518

519
520
521
522
523
524
525
526
527
528
529

530
531
532
533
534
535
536
537

538
539
540

541
542
543

544
545
546
547
548
549
550
551
552
553
554
555
556

E18: Reference to Identity Provider Discovery Service in ECP Profile

Change [SAMLProf] Section 4.2.2 at lines 725-726 to remove the incorrect implication that an ECP is a direct participant in the identity provider discovery profile.

New:

In step 3, the ECP obtains the location of an endpoint at an identity provider for the authentication request protocol that supports its preferred binding. The means by which this is accomplished is implementation-dependent. ~~The ECP MAY use the SAML identity provider discovery profile described in Section 4.3.~~

E19: Clarification on Error Processing

Change [SAMLBind] Section 3.2.2.1 at lines 310-317 and Section 3.2.3.3 at line 378 to clarify SAML error processing and its relationship to SOAP error processing.

Original at Section 3.2.2.1, lines 310-317:

The SAML responder **MUST** return **either a SAML response element within the body of another SOAP message or generate a SOAP fault**. The SAML responder **MUST NOT** include more than one SAML response per SOAP message or include any additional XML elements in the SOAP body. **If a SAML responder cannot, for some reason, process a SAML request, it MUST generate a SOAP fault**. SOAP fault codes **MUST NOT** be sent for errors within the SAML problem domain, for example, inability to find an extension schema or as a signal that the subject is not authorized to access a resource in an authorization query. (SOAP 1.1 faults and fault codes are discussed in [SOAP11] Section 4.1.)

New at Section 3.2.2.1, lines 310-317:

The SAML responder **SHOULD** return a **SOAP message containing either a SAML response element in the body or a SOAP fault**. The SAML responder **MUST NOT** include more than one SAML response per SOAP message or include any additional XML elements in the SOAP body. SOAP fault codes **SHOULD NOT** be sent for errors within the SAML problem domain, for example, inability to find an extension schema or as a signal that the subject is not authorized to access a resource in an authorization query. **See Section 3.2.3.3 for more information about error handling**. (SOAP 1.1 faults and fault codes are discussed in [SOAP11] Section 4.1.)

Original at Section 3.2.3.3, line 378:

In the case of a SAML processing error, the SOAP HTTP server **MUST** respond with "200 OK" and include a SAML-specified `<samlp:Status>` element in the SAML response within the SOAP body.

New at Section 3.2.3.3, line 378:

In the case of a SAML processing error, the SOAP HTTP server **SHOULD** respond with "200 OK" and include a SAML-specified `<samlp:Status>` element in the SAML response within the SOAP body.

E20: ECP SSO Profile and Metadata

Change [SAMLProf] at line 1081 to add a new subsection, Section 4.2.6, in order to add metadata considerations to the ECP profile.

New (small portion of previous subsection shown):

The ECP **SHOULD** be authenticated to the identity provider, such as by maintaining an authenticated session. Any HTTP exchanges subsequent to the delivery of the `<AuthnRequest>` message and before the identity provider returns a `<Response>` **MUST** be securely associated with the original request.

4.2.6 Use of Metadata

The rules specified in the browser SSO profile in Section 4.1.6 apply here as well. Specifically, the indexed endpoint element `<md:AssertionConsumerService>` with a binding of `urn:oasis:names:tc:SAML:2.0:bindings:PAOS` **MAY** be used to describe the supported

557 binding and location(s) to which an identity provider may send responses to a service provider
 558 using this profile. IN addition, the endpoint <md:SingleSignOnService> with a binding of
 559 urn:oasis:names:tc:SAML:2.0:bindings:SOAP MAY be used to describe the supported
 560 binding and location(s) to which a service provider may send requests to an identity provider using
 561 this profile.

562 E21: PAOS Version

563 Change [SAMLBind] Section 3.3.3 at line 474 to clarify the PAOS version required. New:

- 564 ● The HTTP PAOS Header field MUST be present and specify the PAOS version with
 565 "urn:liberty:paos:2003-08" **at a minimum.**

566 E22: Error in Profile/ECP

567 Change [SAMLProf] Section 4.2.4.1 at line 907 to refer to the **AssertionConsumerServiceURL** attribute
 568 rather than the **AssertionServiceConsumerURL** attribute. This was a typographical error.

569 E24: HTTPS in URI Binding

570 Change [SAMLBind] Section 3.7 at lines 1349-1351 to make the HTTP support requirements more
 571 appropriate in the context of the URI binding.

572 Original:

573 Like SOAP, URI resolution can occur over multiple underlying transports. This binding has **transport-**
 574 **independent** aspects, but also calls out the **use of HTTP with SSL 3.0 [SSL3] or TLS 1.0 [RFC2246] as**
 575 **REQUIRED (mandatory to implement).**

576 New:

577 Like SOAP, URI resolution can occur over multiple underlying transports. This binding has **protocol-**
 578 **independent** aspects, but also calls out **as mandatory the implementation of HTTP URIs.**

579 E25: Metadata Feature in Conformance

580 Change [SAMLConf] in Section 3.2 (Tables 2 and 4) to add feature rows, and at line 231 to add two
 581 subsections, Sections 3.6 and 3.7, in order to reflect conformance aspects of the SAML metadata feature.

582 New in Table 2:

583	Feature	IdP	IdP Lite	SP	SP Lite	ECP
584	Metadata Structures	OPT	OPT	OPT	OPT	N/A
585	Metadata Interoperation	OPT	OPT	OPT	OPT	N/A

586 New in Table 4:

587	Feature	Authn	Attrib	Authz	Requester
588	Metadata Structures	OPT	OPT	OPT	OPT
589	Metadata Interoperation	OPT	OPT	OPT	OPT

590 New at line 231 (small portion of previous subsection shown):

591 If a SAML authority uses SSL 3.0 or TLS 1.0, it MUST use a server-side certificate.

592

593 3.6 Metadata Structures

594

595 **Implementations claiming conformance to SAML V2.0 may declare each operational mode's**
 596 **conformance to SAML V2.0 Metadata [SAMLMeta] through election of the Metadata Structures**
 597 **option.**
 598

599 With respect to each operational mode, such conformance entails the following:
600

601 ● Implementing SAML metadata according to the extensible SAML V2.0 Metadata format in all cases
602 where an interoperating peer has the option, as stated in SAML V2.0 specifications, of depending on
603 the existence of SAML V2.0 Metadata. Electing the Metadata Structures option has the effect of
604 requiring that such metadata be available to the interoperating peer. The Metadata Interoperation
605 feature, described below, provides a means of satisfying this requirement.

606
607 ● Referencing, consuming, and adhering to the SAML metadata, according to [SAMLMeta], of an
608 interoperating peer when the known metadata relevant to that peer and the particular operation, and
609 the current exchange, has expired or is no longer valid in cache, provided the metadata is available
610 and is not prohibited by policy or the particular operation and that specific exchange.

611 3.7 Metadata Interoperation 612

613 Election of the Metadata Interoperation option requires the implementation to offer, in addition to
614 any other mechanism, the well-known location publication and resolution mechanism described in
615 the SAML metadata specification [SAMLMeta].
616

617 E26: Ambiguities Around Multiple Assertions and Statements in 618 the SSO Profile

619 Change [SAMLProf] Section 4.1.4.2 at lines 541-572, Section 4.1.4.3 at lines 576-591, and Section
620 4.1.4.5 at lines 600-601 to resolve ambiguities around the usage of multiple assertions and multiple
621 statements within an assertion in the SSO profile.

622 Original at Section 4.1.4.2, lines 541-572:

- 623 • The <Issuer> element MAY be omitted, but if present it MUST contain the unique identifier of the
624 issuing identity provider; the Format attribute MUST be omitted or have a value of
625 urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- 626 • It MUST contain at least one <Assertion>. Each assertion's <Issuer> element MUST contain the
627 unique identifier of the **issuing** identity provider; the Format attribute MUST be omitted or have a value
628 of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- 629 • **The set of one or more assertions MUST contain at least one <AuthnStatement> that reflects**
630 **the authentication of the principal to the identity provider.**
- 631 • **At least one assertion containing an <AuthnStatement> MUST contain a <Subject> element**
632 **with at least one <SubjectConfirmation> element containing a Method of**
633 **urn:oasis:names:tc:SAML:2.0:cm:bearer. If the identity provider supports the Single**
634 **Logout profile, defined in Section 4.4, any such authentication statements MUST include a**
635 **SessionIndex attribute to enable per-session logout requests by the service provider.**
- 636 • **The bearer <SubjectConfirmation> element described above MUST contain a**
637 **<SubjectConfirmationData> element that contains a Recipient attribute containing the**
638 **service provider's assertion consumer service URL and a NotOnOrAfter attribute that limits the**
639 **window during which the assertion can be delivered. It MAY contain an Address attribute limiting**
640 **the client address from which the assertion can be delivered. It MUST NOT contain a NotBefore**
641 **attribute. If the containing message is in response to an <AuthnRequest>, then the**
642 **InResponseTo attribute MUST match the request's ID.**
- 643 • Other statements **and confirmation methods** MAY be included in the assertion(s) at the discretion of
644 the identity provider. In particular, <AttributeStatement> elements MAY be included. The
645 <AuthnRequest> MAY contain an AttributeConsumingServiceIndex XML attribute referencing
646 information about desired or required attributes in [SAMLMeta]. The identity provider MAY ignore this, or
647 send other attributes at its discretion.
- 648 • **The assertion(s) containing a bearer subject confirmation MUST contain an**
649 **<AudienceRestriction> including the service provider's unique identifier as an <Audience>.**

- 650 • Other conditions (and other <Audience> elements) MAY be included as requested by the service
651 provider or at the discretion of the identity provider. (Of course, all such conditions MUST be understood
652 by and accepted by the service provider in order for the assertion to be considered valid.) The identity
653 provider is NOT obligated to honor the requested set of <Conditions> in the <AuthnRequest>, if
654 any.
- 655 • The identity provider is NOT obligated to honor the requested set of <Conditions> in the
656 <AuthnRequest>, if any.

657 New at Section 4.1.4.2, lines 541-572 (note that E17 specifies additional changes to the first bullet item
658 shown here):

- 659 • The <Issuer> element MAY be omitted, but if present it MUST contain the unique identifier of the
660 issuing identity provider; the Format attribute MUST be omitted or have a value of
661 urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- 662 • It MUST contain at least one <Assertion>. Each assertion's <Issuer> element MUST contain the
663 unique identifier of the **responding** identity provider; the Format attribute MUST be omitted or have a
664 value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity. **Note that this profile**
665 **assumes a single responding identity provider, and all assertions in a response MUST be issued**
666 **by the same entity.**
- 667 • **If multiple assertions are included, then each assertion's <Subject> element MUST refer to the**
668 **same principal. It is allowable for the content of the <Subject> elements to differ (e.g. using**
669 **different <NameID> or alternative <SubjectConfirmation> elements).**
- 670 • **Any assertion issued for consumption using this profile MUST contain a <Subject> element**
671 **with at least one <SubjectConfirmation> element containing a Method of**
672 **urn:oasis:names:tc:SAML:2.0:cm:bearer. Such an assertion is termed a bearer**
673 **assertion. Bearer assertions MAY contain additional <SubjectConfirmation> elements.**
- 674 • **Assertions without a bearer <SubjectConfirmation> MAY also be included; processing of**
675 **additional assertions or <SubjectConfirmation> elements is outside the scope of this**
676 **profile.**
- 677 • **At least one bearer <SubjectConfirmation> element MUST contain a**
678 **<SubjectConfirmationData> element that itself MUST contain a Recipient attribute**
679 **containing the service provider's assertion consumer service URL and a NotOnOrAfter**
680 **attribute that limits the window during which the assertion can be [PE52]confirmed by the relying**
681 **party. It MAY also contain an Address attribute limiting the client address from which the**
682 **assertion can be delivered. It MUST NOT contain a NotBefore attribute. If the containing**
683 **message is in response to an <AuthnRequest>, then the InResponseTo attribute MUST**
684 **match the request's ID.**
- 685 • **The set of one or more bearer assertions MUST contain at least one <AuthnStatement> that**
686 **reflects the authentication of the principal to the identity provider. Multiple <AuthnStatement>**
687 **elements MAY be included, but the semantics of multiple statements is not defined by this**
688 **profile.**
- 689 • **If the identity provider supports the Single Logout profile, defined in Section Error: Reference**
690 **source not found, any authentication statements MUST include a SessionIndex attribute to**
691 **enable per-session logout requests by the service provider.**
- 692 • Other statements MAY be included in the **bearer** assertion(s) at the discretion of the identity provider. In
693 particular, <AttributeStatement> elements MAY be included. The <AuthnRequest> MAY contain
694 an AttributeConsumingServiceIndex XML attribute referencing information about desired or
695 required attributes in [SAMLMeta]. The identity provider MAY ignore this, or send other attributes at its
696 discretion.
- 697 • **Each bearer** assertion MUST contain an <AudienceRestriction> including the service provider's
698 unique identifier as an <Audience>.

699 • Other conditions (and other <Audience> elements) MAY be included as requested by the service
700 provider or at the discretion of the identity provider. (Of course, all such conditions MUST be understood
701 by and accepted by the service provider in order for the assertion to be considered valid.) The identity
702 provider is NOT obligated to honor the requested set of <Conditions> in the <AuthnRequest>, if
703 any.

704 • The identity provider is NOT obligated to honor the requested set of <Conditions> in the
705 <AuthnRequest>, if any.

706 Original at Section 4.1.4.3, lines 576-591:

707 • Verify that the Recipient attribute in any bearer <SubjectConfirmationData> matches the assertion
708 consumer service URL to which the <Response> or artifact was delivered

709 • Verify that the NotOnOrAfter attribute in any bearer <SubjectConfirmationData> has not passed,
710 subject to allowable clock skew between the providers

711 • Verify that the InResponseTo attribute in the bearer <SubjectConfirmationData> equals the ID of
712 its original <AuthnRequest> message, unless the response is unsolicited (see Section 4.1.5), in which
713 case the attribute MUST NOT be present

714 • Verify that any assertions relied upon are valid in other respects.

715 • If any bearer <SubjectConfirmationData> includes an Address attribute, the service provider MAY
716 check the user agent's client address against it.

717 • Any assertion which is not valid, or whose subject confirmation requirements cannot be met SHOULD be
718 discarded and SHOULD NOT be used to establish a security context for the principal.

719 • If an <AuthnStatement> used to establish a security context for the principal contains a
720 SessionNotOnOrAfter attribute, the security context SHOULD be discarded once this time is reached,
721 unless the service provider reestablishes the principal's identity by repeating the use of this profile.

722 New at Section 4.1.4.3, lines 576-591:

723 • Verify that the Recipient attribute in **the** bearer <SubjectConfirmationData> matches the assertion
724 consumer service URL to which the <Response> or artifact was delivered

725 • Verify that the NotOnOrAfter attribute in **the** bearer <SubjectConfirmationData> has not passed,
726 subject to allowable clock skew between the providers

727 • Verify that the InResponseTo attribute in the bearer <SubjectConfirmationData> equals the ID of
728 its original <AuthnRequest> message, unless the response is unsolicited (see Section 4.1.5), in which
729 case the attribute MUST NOT be present

730 • Verify that any assertions relied upon are valid in other respects. **Note that while multiple bearer
731 <SubjectConfirmation> elements may be present, the successful evaluation of a single such
732 element in accordance with this profile is sufficient to confirm an assertion. However, each
733 assertion, if more than one is present, MUST be evaluated independently.**

734 • If **any the** bearer <SubjectConfirmationData> includes an Address attribute, the service provider
735 MAY check the user agent's client address against it.

736 • Any assertion which is not valid, or whose subject confirmation requirements cannot be met SHOULD be
737 discarded and SHOULD NOT be used to establish a security context for the principal.

738 • If an <AuthnStatement> used to establish a security context for the principal contains a
739 SessionNotOnOrAfter attribute, the security context SHOULD be discarded once this time is reached,
740 unless the service provider reestablishes the principal's identity by repeating the use of this profile. **Note
741 that if multiple <AuthnStatement> elements are present, the SessionNotOnOrAfter value closest
742 to the present time SHOULD be honored.**

743 Original at Section 4.1.4.5, lines 600-601:

744 If the HTTP POST binding is used to deliver the <Response>, the enclosed assertion(s) MUST be signed.

749 New at Section 4.1.4.5, lines 600-601:

750 If the HTTP POST binding is used to deliver the <Response>, **each assertion MUST be protected by a**
751 **digital signature. This can be accomplished by signing each individual <Assertion> element or by**
752 **signing the <Response> element.**

753 **E27: Incorrect Step Number in ECP Profile**

754 Change [SAMLProf] Section 4.2.4.3 at line 947 to change the reference to the step number from **5** to **7**.
755 This was a typographical error.

756 **E28: Profile Labeling in Conformance**

757 Change [SAMLConf] Section 2 at Table 1 to make its labeling and categorization of profiles more
758 consistent.

759 Combine the profile rows labeled **Artifact Resolution**, **Authentication Query**, **Attribute Query**, and
760 **Authorization Decision Query** into a single profile row labeled **Assertion Query/Request** in column 1,
761 with the breakdown of these four protocol types moved to column 2 (message flows) for that row.

762 Remove the profile rows labeled **SAML URI binding** and **Metadata**.

763 **E29: Incomplete Listing of Features in Conformance**

764 Change [SAMLConf] Section 3.2 at Table 2 to include missing feature rows. New:

Feature	IdP	IdP Lite	SP	SP Lite	ECP
Request for Assertion by Identifier	OPT	N/A	N/A	N/A	N/A
SAML URI Binding	OPT	N/A	N/A	N/A	N/A

768 **E30: Key Replacement**

769 Change [SAMLCore] Section 6.1 at line 3110 to improve wording around key replacement. Original:

770 Encrypted data and **optionally one** or more encrypted keys **MUST** replace the plaintext information in the
771 same location within the XML instance.

772 New:

773 Encrypted data and **zero** or more encrypted keys **MUST** replace the plaintext information in the same
774 location within the XML instance.

775 **E31: Various Minor Errors in Binding**

776 Change [SAMLBind] Section 3.3.5 at line 511, Section 3.5.3 at line 785, and Section 3.6.5 at lines 1136
777 and 1397 to clean up various minor wording errors.

778 At Section 3.3.5, line 511, capitalize the word **RECOMMENDED**.

779 Original at Section 3.5.3, line 785:

780 If no such **value** is included with a SAML request message, or if the SAML response message is being
781 generated without a corresponding request ...

782 New at Section 3.5.3, line 785:

783 If no such **RelayState data** is included with a SAML request message, or if the SAML response message is
784 being generated without a corresponding request ...

785 Original at Section 3.6.5, line 1136:

786 The SAML requester determines the SAML responder by examining the artifact, and issues a
787 <samlp:ArtifactResolve> request containing the artifact to the SAML responder using a **direct** SAML
788 binding, as in step 3.

789 New at Section 3.6.5, line 1136:

790 The SAML requester determines the SAML responder by examining the artifact, and issues a
791 <samlp:ArtifactResolve> request containing the artifact to the SAML responder using a **synchronous**
792 SAML binding, as in step 3.

793 Original at Section 3.6.5, line 1397:

794 Note that the use of wildcards **is not allowed for on** such queries.

795 New at Section 3.6.5, line 1397:

796 Note that **the URI syntax does not support** the use of wildcards in such **ID** queries.

797 **E32: Missing Required Information in Profiles**

798 Change [SAMLProf] at line 1092. New subsection added at line 1092 as Section 4.3.1, incrementing the
799 subsection numbers of the existing Sections 4.3.1 through 4.3.3:

800 **4.3.1 Required Information**

801 **Identification:** urn:oasis:names:tc:SAML:2.0:profiles:SSO:idp-discovery

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

803 **Description:** Given below.

804 **Updates:** None.

805 **E33: References to Assertion Request Protocol**

806 Change [SAMLMeta] Section 2.4.3 at line 700, Section 2.4.5 at line 838, Section 2.4.6 at line 871, and
807 Section 2.4.7 at line 904 to change references to the **Assertion Request** protocol to **Assertion**
808 **Query/Request**. This is just a typographical error.

809 **E34: RequestedAttribute Section Heading**

810 Change [SAMLMeta] at line 809 to make the Section **2.4.4.2** heading be a level below, at **2.4.4.1.1**, for
811 consistency in reflecting element nesting in the document outline.

812 **E35: Response Consumer URL Rules and Example**

813 Change [SAMLProf] Section 4.2.4.1 at lines 906-908, and Section 4.2.4.3 at line 964, to make the
814 example conform to the rules for a response consumer URL and explain these rules more clearly.

815 Original at Section 4.2.4.1, lines 906-908:

816 Specifies where the ECP is to send an error response. Also used to verify the correctness of the identity
817 provider's response, by cross checking this location against the **AssertionServiceConsumerURL** in the
818 ECP response header block. This value **MUST** be the same as the AssertionServiceConsumerURL (or the
819 URL referenced in metadata) conveyed in the <AuthnRequest>.

820 New at lines Section 4.2.4.1, 906-908:

821 Specifies where the ECP is to send an error response. Also used to verify the correctness of the identity
822 provider's response, by cross checking this location against the **AssertionConsumerServiceURL** in the
823 ECP response header block. This value **MUST** be the same as the AssertionServiceConsumerURL (or the
824 URL referenced in metadata) conveyed in the <AuthnRequest> **and SHOULD NOT be a relative URL**.

825 Original at Section 4.2.4.3, line 964:

```
826 <paos:Request xmlns:paos="urn:liberty:paos:2003-08"  
827 responseConsumerURL="http://identity-service.example.com/abc"
```

828 New at Section 4.2.4.3, line 964:

```
829 <paos:Request xmlns:paos="urn:liberty:paos:2003-08"  
830 responseConsumerURL="  
831 https://ServiceProvider.example.com/ecp_assertion_consumer"
```

832 E36: Clarification on Action Element

833 Change [SAMLCore] Section 2.7.4.2 at lines 1359-1363 to remove the incorrect specification text that
834 says the action namespace is optional (the schema mandates it, and in cases of disagreement, the
835 schema takes precedence).

836 Original:

```
837 Namespace [Optional]  
838 A URI reference representing the namespace in which the name of the specified action is to be interpreted.  
839 If this element is absent, the namespace urn:oasis:names:tc:SAML:1.0:action:rwedc-negation  
840 specified in Section 8.1.2 is in effect.
```

841 New:

```
842 Namespace [Required]  
843 A URI reference representing the namespace in which the name of the specified action is to be interpreted.
```

844 E37: Clarification in Metadata on Indexed Endpoints

845 Change [SAMLMeta] Section 2.2.3 at line 272 to clarify what it means for two endpoints to be “like”.

846 Original:

```
847 In any such sequence of like endpoints based on this type, the default endpoint is the first such endpoint  
848 with the isDefault attribute set to true.
```

849 New:

```
850 In any such sequence of indexed endpoints that share a common element name and namespace (i.e. all  
851 instances of <md:AssertionConsumerService> within a role), the default endpoint is the first such  
852 endpoint with the isDefault attribute set to true.
```

853 E38: Clarification Regarding Index on <LogoutRequest>

854 Change [SAMLCore] Section 3.7.1 at line 2546 and [SAMLProf] Section 4.4.4.1 at lines 1302-1304 to
855 clarify requirements around session indexes in logout requests.

856 Original at [SAMLCore] Section 3.7.1, line 2546:

```
857 <SessionIndex> [Optional]  
858 The identifier that indexes this session at the message recipient.
```

859 New at [SAMLCore] Section 3.7.1, line 2546:

```
860 <SessionIndex> [Optional]  
861 The index of the session between the principal identified by the <saml:BaseID>, <saml:NameID>, or <saml:EncryptedID> element, and the session authority. This must correlate to the  
862 SessionIndex attribute, if any, in the <saml:AuthnStatement> of the assertion used to establish  
863 the session that is being terminated.
```

865 New at [SAMLProf] Section 4.4.4.1, lines 1302-1304:

866 If the requester is a session participant, it MUST include at least one <SessionIndex> element in the
867 request. (Note that the session participant always receives a SessionIndex attribute in the
868 <saml:AuthnStatement> elements that it receives to initiate the session, per Section 4.1.4.2 of
869 the Web Browser SSO Profile.) If the requester is a session authority (or acting on its behalf), then it MAY
870 omit any such elements to indicate the termination of all of the principal's applicable sessions.

871 E39: Error in SAML Profile Example

872 **Note:** E39 corrects text in a section that is affected by E53, which deprecates the entire
873 section. Please see E53 for details.

874 Change [SAMLProf] Section 8.5.6 at lines 2095-2098 to move the ldapprof:Encoding attribute to the
875 correct location.

876 Original:

```
877 <saml:Attribute  
878   xmlns:xacmlprof="urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML"  
879   xmlns:ldapprof="urn:oasis:names:tc:SAML:2.0:profiles:attribute:LDAP"  
880  xacmlprof:DataType="http://www.w3.org/2001/XMLSchema#string"  
881   ldapprof:Encoding="LDAP"  
882   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"  
883   Name="urn:oid:2.5.4.42" FriendlyName="givenName">  
884   <saml:AttributeValue xsi:type="xs:string">By-Tor</saml:AttributeValue>  
885 </saml:Attribute>
```

886 New:

```
887 <saml:Attribute  
888   xmlns:xacmlprof="urn:oasis:names:tc:SAML:2.0:profiles:attribute:XACML"  
889   xmlns:ldapprof="urn:oasis:names:tc:SAML:2.0:profiles:attribute:LDAP"  
890  xacmlprof:DataType="http://www.w3.org/2001/XMLSchema#string"  
891   NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"  
892   Name="urn:oid:2.5.4.42" FriendlyName="givenName">  
893   <saml:AttributeValue xsi:type="xs:string"  
894   ldapprof:Encoding="LDAP">By-Tor</saml:AttributeValue>  
895 </saml:Attribute>
```

896 E40: Holder of Key

897 Change [SAMLProf] Section 3.1 at lines 335-337 to align the description of Holder of Key in the profiles
898 specification with the language in the core specification.

899 Original:

900 As described in [XMLSig], each <ds:KeyInfo> element holds a key or information that enables an
901 application to obtain a key. The holder of a specified key is considered to be **the subject** of the assertion by
902 the asserting party.

903 New (note that E47 specifies additional changes to the original text shown here):

904 As described in [XMLSig], each <ds:KeyInfo> element holds a key or information that enables an
905 application to obtain a key. The holder of a specified key is considered to be **an acceptable attesting entity**
906 **for** the assertion by the asserting party.

907 E41: EndpointType ResponseLocation Clarification in Metadata

908 Change [SAMLMeta] Section 2.2.2 at line 242 to clarify correct behavior when the response location is
909 omitted from the metadata.

910 New:

911 The `ResponseLocation` attribute is used to enable different endpoints to be specified for receiving request
 912 and response messages associated with a protocol or profile, not as a means of load-balancing or
 913 redundancy (multiple elements of this type can be included for this purpose). When a role contains an
 914 element of this type pertaining to a protocol or profile for which only a single type of message (request or
 915 response) is applicable, then the `ResponseLocation` attribute is unused. **If the `ResponseLocation`**
 916 **attribute is omitted, any response messages associated with a protocol or profile may be assumed**
 917 **to be handled at the URI indicated by the `Location` attribute.**

918 E42: Match Authorities to Queries in Conformance

919 Change [SAMLConf] Section 3.2 at Table 4 to indicate more precisely the relationship between SAML
 920 authorities and queries for types of assertion statements that those authorities do not specialize in
 921 producing.

922 Original:

923 Feature	Authn	Attrib	Authz	Requester
924 Authentication Query, SOAP	MUST	OPT	OPT	OPT
925 Attribute Query, SOAP	OPT	MUST	OPT	OPT
926 Authorization Decision Query, SOAP	OPT	OPT	MUST	OPT

927 New:

928 Feature	Authn	Attrib	Authz	Requester
929 Authentication Query, SOAP	MUST	N/A	N/A	OPT
930 Attribute Query, SOAP	N/A	MUST	N/A	OPT
931 Authorization Decision Query, SOAP	N/A	N/A	MUST	OPT

932 E43: Key Location in `saml:EncryptedData`

933 Change [SAMLCore] at line 3116 by replacing the existing Section 6.2 with new Sections 6.2 and 6.3 to
 934 reflect correct application and usage of the XML Encryption standard and to add several examples to fully
 935 demonstrate this.

936 Original:

937 6.2 Combining Signatures and Encryption

938 Use of XML Encryption and XML Signature MAY be combined. When an assertion is to be signed
 939 and encrypted, the following rules apply. A relying party MUST perform signature validation and
 940 decryption in the reverse order that signing and encryption were performed.

941 • When a signed `<Assertion>` element is encrypted, the signature MUST first be calculated and
 942 placed within the `<Assertion>` element before the element is encrypted.

943 • When a `<BaseID>`, `<NameID>`, or `<Attribute>` element is encrypted, the encryption MUST be
 944 performed first and then the signature calculated over the assertion or message containing the
 945 encrypted element.

946 New:

947 6.2 Key and Data Referencing Guidelines

948 If an encrypted key is NOT included in the XML instance, then the relying party must be able to
 949 locally determine the decryption key, per [XMLEnc].

950 Implementations of SAML MAY implicitly associate keys with the corresponding data they are used
 951 to encrypt, through the positioning of `<xenc:EncryptedKey>` elements next to the associated
 952 `<xenc:EncryptedData>` element, within the enclosing SAML parent element. However, the
 953 following set of explicit referencing guidelines are suggested to facilitate interoperability.

954 If the encrypted key is included in the XML instance, then it SHOULD be referenced within the
 955 associated `<xenc:EncryptedData>` element, or alternatively embedded within the
 956 `<xenc:EncryptedData>` element. When an `<xenc:EncryptedKey>` element is used, the

957 <ds:KeyInfo> element within <xenc:EncryptedData> SHOULD reference the
958 <xenc:EncryptedKey> element using a <ds:RetrievalMethod> element of Type
959 http://www.w3.org/2001/04/xmlenc#EncryptedKey.

960 In addition, an <xenc:EncryptedKey> element SHOULD contain an <xenc:ReferenceList>
961 element containing a <xenc:DataReference> that references the corresponding
962 <xenc:EncryptedData> element(s) that the key was used to encrypt.

963 In scenarios where the encrypted element is being "multicast" to multiple recipients, and the key
964 used to encrypt the message must be in turn encrypted individually and independently for each of
965 the multiple recipients, the <xenc:CarriedKeyName> element SHOULD be used to assign a
966 common name to each of the <xenc:EncryptedKey> elements so that a <ds:KeyName> can be
967 used from within the <xenc:EncryptedData> element's <ds:KeyInfo> element.

968 Within the <xenc:EncryptedData> element, the <ds:KeyName> can be thought of as an "alias" that
969 is used for backwards referencing from the <xenc:CarriedKeyName> element in each individual
970 <xenc:EncryptedKey> element. While this accommodates a "multicast" approach, each recipient
971 must be able to understand (at least one) <ds:KeyName>. The Recipient attribute is used to
972 provide a hint as to which key is meant for which recipient.

973 The SAML implementation has the discretion to accept or reject a message where multiple
974 Recipient attributes or <ds:KeyName> elements are understood. It is RECOMMENDED that
975 implementations simply use the first key they understand and ignore any additional keys.

976 6.3 Examples

977 In the following example, the parent element (<EncryptedID>) contains <xenc:EncryptedData>
978 and (referenced) <xenc:EncryptedKey> elements as siblings (note that the key can in fact be
979 anywhere in the same instance, and the key references the <xenc:EncryptedData> element):

```
980 <saml:EncryptedID xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">  
981   <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"   
982     Id="Encrypted_DATA_ID"  
983     Type="http://www.w3.org/2001/04/xmlenc#Element">  
984     <xenc:EncryptionMethod  
985       Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>  
986     <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">  
987       <ds:RetrievalMethod URI="#Encrypted_KEY_ID"  
988         Type="http://www.w3.org/2001/04/xmlenc#EncryptedKey"/>  
989     </ds:KeyInfo>  
990     <xenc:CipherData>  
991       <xenc:CipherValue>Nk4W4mx...</xenc:CipherValue>  
992     </xenc:CipherData>  
993   </xenc:EncryptedData>  
994  
995   <xenc:EncryptedKey xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"   
996     Id="Encrypted_KEY_ID">  
997     <xenc:EncryptionMethod  
998       Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>  
999     <xenc:CipherData>  
1000       <xenc:CipherValue>PzA5X...</xenc:CipherValue>  
1001     </xenc:CipherData>  
1002     <xenc:ReferenceList>  
1003       <xenc:DataReference URI="#Encrypted_DATA_ID"/>  
1004     </xenc:ReferenceList>  
1005   </xenc:EncryptedKey>
```

1006 In the following <EncryptedAttribute> example, the <xenc:EncryptedKey> element is contained
1007 within the <xenc:EncryptedData> element, so there is no explicit referencing:

```
1008 <saml:EncryptedAttribute  
1009   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">  
1010   <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"   
1011     Id="Encrypted_DATA_ID"  
1012     Type="http://www.w3.org/2001/04/xmlenc#Element">
```

```

1013 <xenc:EncryptionMethod
1014   Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
1015 <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
1016   <xenc:EncryptedKey Id="Encrypted_KEY_ID">
1017     <xenc:EncryptionMethod
1018       Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
1019     <xenc:CipherData>
1020       <xenc:CipherValue>SDFSDF... </xenc:CipherValue>
1021     </xenc:CipherData>
1022   </xenc:EncryptedKey>
1023 </ds:KeyInfo>
1024 <xenc:CipherData>
1025   <xenc:CipherValue>Nk4W4mx...</xenc:CipherValue>
1026 </xenc:CipherData>
1027 </xenc:EncryptedData>
1028 </saml:EncryptedAttribute>

```

1029 **The final example shows an assertion encrypted for multiple recipients, using the**
1030 **<xenc:CarriedKeyName> approach:**

```

1031 <saml:EncryptedAssertion
1032   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
1033   <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
1034     Id="Encrypted_DATA_ID"
1035     Type="http://www.w3.org/2001/04/xmlenc#Element">
1036     <xenc:EncryptionMethod
1037       Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
1038     <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
1039       <ds:KeyName>MULTICAST_KEY_NAME</ds:KeyName>
1040     </ds:KeyInfo>
1041     <xenc:CipherData>
1042       <xenc:CipherValue>Nk4W4mx...</xenc:CipherValue>
1043     </xenc:CipherData>
1044   </xenc:EncryptedData>
1045
1046   <xenc:EncryptedKey xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
1047     Id="Encrypted_KEY_ID_1" Recipient="https://sp1.org">
1048     <xenc:EncryptionMethod
1049       Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
1050     <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
1051       <ds:KeyName>KEY_NAME_1</ds:KeyName>
1052     </ds:KeyInfo>
1053     <xenc:CipherData>
1054       <xenc:CipherValue>xyzABC...</xenc:CipherValue>
1055     </xenc:CipherData>
1056     <xenc:ReferenceList>
1057       <xenc:DataReference URI="#Encrypted_DATA_ID"/>
1058     </xenc:ReferenceList>
1059
1060     <xenc:CarriedKeyName>MULTICAST_KEY_NAME</xenc:CarriedKeyName>
1061   </xenc:EncryptedKey>
1062
1063   <xenc:EncryptedKey xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
1064     Id="Encrypted_KEY_ID_2" Recipient="https://sp2.org">
1065     <xenc:EncryptionMethod
1066       Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
1067     <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
1068       <ds:KeyName>KEY_NAME_2</ds:KeyName>
1069     </ds:KeyInfo>
1070     <xenc:CipherData>
1071       <xenc:CipherValue>abcXYZ...</xenc:CipherValue>
1072     </xenc:CipherData>
1073     <xenc:ReferenceList>

```

```
1074     <xenc:DataReference URI="#Encrypted_DATA_ID"/>
1075     </xenc:ReferenceList>
1076
1077     <xenc:CarriedKeyName>MULTICAST_KEY_NAME</xenc:CarriedKeyName>
1078     </xenc:EncryptedKey>
1079 </saml:EncryptedAssertion>
```

1080 E45: AuthnContext Comparison Order

1081 Change [SAMLCore] Section 3.3.2.2.1 at lines 1815-1819 and 1826 to clarify the lack of orderedness in
1082 the comparison of a set of authentication contexts.

1083 Original at Section 3.3.2.2.1, lines 1815-1819:

1084 Either a set of class references or a set of declaration references can be used. The set of supplied
1085 references MUST be evaluated as an ordered set, where the first element is the most preferred
1086 authentication context class or declaration. If none of the specified classes or declarations can be satisfied in
1087 accordance with the rules below, then the responder MUST return a <Response> message with a second-
1088 level <StatusCode> of urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext.

1089 New at Section 3.3.2.2.1, lines 1815-1819:

1090 Either a set of class references or a set of declaration references can be used. **If ordering is relevant to**
1091 **the evaluation of the request, then** the set of supplied references MUST be evaluated as an ordered set,
1092 where the first element is the most preferred authentication context class or declaration. If none of the
1093 specified classes or declarations can be satisfied in accordance with the rules below, then the responder
1094 MUST return a <Response> message with a second-level <StatusCode> of
1095 urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext. **For example, ordering is significant**
1096 **when using this element in an <AuthnRequest> message but not in an <AuthnQuery> message.**

1097 Original at Section 3.3.2.2.1, line 1826:

1098 If Comparison is set to "better", then the resulting authentication context in the authentication statement
1099 MUST be stronger (as deemed by the responder) than **any** of the authentication contexts specified.

1100 New at Section 3.3.2.2.1, line 1826:

1101 If Comparison is set to "better", then the resulting authentication context in the authentication statement
1102 MUST be stronger (as deemed by the responder) than **one** of the authentication contexts specified.

1103 E46: AudienceRestriction Clarifications

1104 Change [SAMLCore] Section 2.5.1.4 at lines 924-925 to clarify the logical sense with respect to individual
1105 audience elements within an audience-restriction condition grouping.

1106 Original:

1107 Note that multiple <AudienceRestriction> elements MAY be included in a single assertion, and each
1108 MUST be evaluated independently. The effect of this requirement and the preceding definition is that within
1109 a given **condition**, the **audiences** form a disjunction (an "OR") while multiple **conditions** form a conjunction
1110 (an "AND").

1111 New:

1112 Note that multiple <AudienceRestriction> elements MAY be included in a single assertion, and each
1113 MUST be evaluated independently. The effect of this requirement and the preceding definition is that within
1114 a given <AudienceRestrictions>, the <Audience> **elements** form a disjunction (an "OR") while
1115 multiple <AudienceRestrictions> **elements** form a conjunction (an "AND").

1116 **E47: Clarification on SubjectConfirmation**

1117 Change [SAMLCore] Section 2.4.1.1 at line 698, and change [SAMLProf] Section 3.1 at lines 336 and 341
1118 and Section 3.3 at lines 361-363, in order to clarify behavior around the subject confirmation element and
1119 the intent of the embedded secondary identifier.

1120 New at [SAMLCore] Section 2.4.1.1, line 698 (add text just before the schema listing introduction):

1121 **If the <SubjectConfirmation> element in an assertion subject contains an identifier the issuer**
1122 **authorizes the attesting entity to wield the assertion on behalf of that subject. A relying party MAY**
1123 **apply additional constraints on the use of such an assertion at its discretion, based upon the**
1124 **identities of both the subject and the attesting entity.**

1125 **If an assertion is issued for use by an entity other than the subject, then that entity SHOULD be**
1126 **identified in the <SubjectConfirmation> element.**

1127 The following schema fragment defines the <SubjectConfirmation> element and its
1128 SubjectConfirmationType complex type:

1129 Original at [SAMLProf] Section 3.1, line 336:

1130 As described in [XMLSig], each <ds:KeyInfo> element holds a key or information that enables an
1131 application to obtain a key. The holder of a **specified key** is considered to be the subject of the assertion by
1132 the asserting party.

1133 New at [SAMLProf] Section 3.1, line 336 (note that E40 specified additional changes to the original text
1134 shown here):

1135 As described in [XMLSig], each <ds:KeyInfo> element holds a key or information that enables an
1136 application to obtain a key. The holder of **one or more of the specified keys** is considered to be the subject
1137 of the assertion by the asserting party.

1138 New at [SAMLProf] Section 3.1, line 341 (add text just before the example):

1139 **If the <SubjectConfirmation> element in an assertion subject contains an identifier the issuer**
1140 **authorizes the attesting entity to wield the assertion on behalf of that subject. A relying party MAY**
1141 **apply additional constraints on the use of such an assertion at its discretion, based upon the**
1142 **identities of both the subject and the attesting entity.**

1143 **If an assertion is issued for use by an entity other than the subject, then that entity SHOULD be**
1144 **identified in the <SubjectConfirmation> element.**

1145 Example: The holder of the key named "By-Tor" or the holder of the key named "Snow Dog" can confirm
1146 itself as the subject.

1147 Original at [SAMLProf] Section 3.3, lines 361-363:

1148 The subject of the assertion is **the bearer of the assertion**, subject to optional constraints on confirmation
1149 using the attributes that MAY be present in the <SubjectConfirmationData> element, as defined by
1150 [SAMLCore].

1151 New at [SAMLProf] Section 3.3, lines 361-363:

1152 The subject of the assertion is **considered to be an acceptable attesting entity for the assertion by the**
1153 **asserting party**, subject to optional constraints on confirmation using the attributes that MAY be present in
1154 the <SubjectConfirmationData> element, as defined by [SAMLCore].

1155 **If the intended bearer is known by the asserting party to be an entity other than the subject, then the**
1156 **asserting party SHOULD identify that entity to the relying party by including a SAML identifier**
1157 **representing it in the enclosing <SubjectConfirmation> element.**

1158 **If multiple attesting entities are to be permitted to use the assertion based on bearer semantics, then**
1159 **multiple <SubjectConfirmation> elements SHOULD be included.**

E48: Clarification on Encoding for Binary Values in LDAP Profile

1160

1161 **Note:** E48 corrects text in a section that is affected by E53, which deprecates the entire
1162 section. Please see E53 for details.

1163 Change [SAMLProf] at line 1762. Original:

1164 For all other LDAP syntaxes, the attribute value is encoded, as the content of the <AttributeValue>
1165 element, by base64-encoding [RFC2045] the **encompassing** ASN.1 OCTET STRING-encoded LDAP
1166 attribute value. The `xsi:type` XML attribute **MUST** be set to `xs:base64Binary`. The profile-specific
1167 `Encoding` XML attribute is provided, with a value of "LDAP".

1168 New:

1169 For all other LDAP syntaxes, the attribute value is encoded, as the content of the <AttributeValue>
1170 element, by base64-encoding [RFC2045] the **contents of the** ASN.1 OCTET STRING-encoded LDAP
1171 attribute value (**not including the ASN.1 OCTET STRING wrapper**). The `xsi:type` XML attribute **MUST**
1172 **be set to** `xs:base64Binary`. The profile-specific `Encoding` XML attribute is provided, with a value of
1173 "LDAP".

E49: Clarification on Attribute Name Format

1174

1175 Change [SAMLCore] Section 2.7.3.1 at line 1217 to clarify the relationship between an attribute's
1176 `NameFormat` setting and its syntax.

1177 New (add text to the end of the definition of <AttributeValue>):

1178 <AttributeValue> [Any Number]

1179 Contains a value of the attribute. If an attribute contains more than one discrete value, it is
1180 RECOMMENDED that each value appear in its own <AttributeValue> element. If more than one
1181 <AttributeValue> element is supplied for an attribute, and any of the elements have a datatype
1182 assigned through `xsi:type`, then all of the <AttributeValue> elements must have the identical
1183 datatype assigned.

1184 **Attributes are identified/named by the combination of the `NameFormat` and `Name` XML attributes**
1185 **described above. Neither one in isolation can be assumed to be unique, but taken together, they**
1186 **ought to be unambiguous within a given deployment.**

1187 **The SAML profiles specification [SAMLProf] includes a number of attribute profiles designed to**
1188 **improve the interoperability of attribute usage in some identified scenarios. Such profiles typically**
1189 **include constraints on attribute naming and value syntax. There is no explicit indicator when an**
1190 **attribute profile is in use, and it is assumed that deployments can establish this out of band, based**
1191 **on the combination of `NameFormat` and `Name`.**

E50: Clarification on SSL Ciphersuites

1192

1193 Change [SAMLConf] Section 4 at line 235 and Section 5 at line 257 to clarify that the named ciphersuites
1194 are not the only ones that can be supported.

1195 New at Section 4, line 235:

1196 SAML V2.0 uses XML Signature [XMLSig] to implement XML signing and encryption functionality for
1197 integrity, and source authentication. SAML V2.0 uses XML Encryption [XMLEnc] to implement
1198 confidentiality, including encrypted identifiers, encrypted assertions, and encrypted attributes. **The**
1199 **algorithms listed below as being required for SAML V2.0 conformance are based on the mandated**
1200 **algorithms in the W3C recommendations for XML Signature and for XML Encryption, but modified by**
1201 **the SSTC to ensure interoperability of conformant SAML implementations. While the SAML-defined**
1202 **set of algorithms is a minimal set for conformance, additional algorithms supported by XML**
1203 **Signature and XML Encryption MAY be used. Note, however, that the use of non-mandated**
1204 **algorithms may introduce interoperability issues if those algorithms are not widely implemented. As**
1205 **additional algorithms become mandated for use in XML Signature and XML Encryption, the set**
1206 **required for SAML conformance may be extended.**

1207 New at Section 5, line 257:

1208 In any SAML V2.0 use of SSL 3.0 [SSL3] or TLS 1.0 [RFC 2246], servers MUST authenticate to clients
1209 using a X.509 v3 certificate. The client MUST establish server identity based on contents of the certificate
1210 (typically through examination of the certificate's subject DN field). **The set of algorithms required for**
1211 **SAML V2.0 conformance is equivalent to that defined in SAML V1.0 and SAML V1.1. These mandated**
1212 **algorithms were chosen by the SSTC because of their wide implementation support in the industry.**
1213 **While the algorithms defined below are the minimal set for SAML conformance, additional**
1214 **algorithms supported by SSL 3.0 and TLS 1.0 MAY be used.**

1215 **E51: Schema Type of Contents of <AttributeValue>**

1216 Change [SAMLProf] Section 8.1.4 at line 1670 to change the reference from **Section 3.3** to **Section 3**, in
1217 order to fix a typographical error that would have improperly restricted the valid types for attribute values
1218 to derived types, rather than the larger category of built-in types.

1219 **E52: Clarification on NotOnOrAfter Attribute for Subject** 1220 **Confirmation**

1221 Change [SAMLProf] Section 4.1.4.2 at line 557 to correctly reflect the type of validity period that applies to
1222 subject confirmation.

1223 Original:

1224 The bearer <SubjectConfirmation> element described above MUST contain a
1225 <SubjectConfirmationData> element that contains a Recipient attribute containing the service
1226 provider's assertion consumer service URL and a NotOnOrAfter attribute that limits the window during
1227 which the assertion can be **delivered**. It MAY contain an Address attribute limiting the client address from
1228 which the assertion can be delivered.

1229 New (note that E26 specifies additional changes to the original text shown here):

1230 The bearer <SubjectConfirmation> element described above MUST contain a
1231 <SubjectConfirmationData> element that contains a Recipient attribute containing the service
1232 provider's assertion consumer service URL and a NotOnOrAfter attribute that limits the window during
1233 which the assertion can be **confirmed by the relying party**. It MAY contain an Address attribute limiting
1234 the client address from which the assertion can be delivered.

1235 **E53: Correction to LDAP/X.500 Profile Attribute**

1236 Deprecate [SAMLProf] Section 8.2 at lines 1677-1799 by adding a notice after line 1677.

1237 New:

1238 **8.2 X.500/LDAP Attribute Profile – Deprecated**
1239 **NOTE: This attribute profile is deprecated because of a flaw that makes it schema-invalid. The SSTC**
1240 **has replaced it with a separately published SAML V2.0 X.500/LDAP Attribute Profile specification**
1241 **that removes this flaw.**
1242 Directories based on the ITU-T X.500 specifications [X.500] and the related IETF Lightweight Directory
1243 Access Protocol specifications [LDAP] are widely deployed....

1244 **E54: Corrections to ECP URN**

1245 Change [SAMLProf] Section 4.2.3.1 at lines 757 and 763-764 to correct the usage of quotation marks in
1246 HTTP headers.

1247 New at line 757 (add double quotation marks around the URN):

1248 Furthermore, support for this profile **MUST** be specified in the HTTP PAOS Header field as a service value,
1249 with the value "urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp".

1250 Original at lines 763-764 (single quotation marks are problematic):

```
1251 GET /index HTTP/1.1
1252 Host: identity-service.example.com
1253 Accept: text/html; application/vnd.paos+xml
1254 PAOS: ver='urn:liberty:paos:2003-08' ;
1255 'urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp'
```

1256 New at lines 763-764 (double quotation marks used instead):

```
1257 GET /index HTTP/1.1
1258 Host: identity-service.example.com
1259 Accept: text/html; application/vnd.paos+xml
1260 PAOS: ver="urn:liberty:paos:2003-08" ;
1261 "urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
```

1262 E55: Language Cleanup Around Name Identifier Management

1263 Change [SAMLCore] Section 3.6.3 at lines 2477, 2483, and 2486-2487, and Section 8.3.7 at lines 3337-
1264 3339, and change [SAMLProf] Section 4.5 at lines 1319 and 1323 to clear up ambiguities around name
1265 identifier management and its application to various name identifier formats and differing identities for a
1266 principal.

1267 Original at [SAMLCore] Section 3.6.3, lines 2477, 2483, and 2486-2487:

1268 If the <Terminate> element is included in the request, the requesting provider is indicating that (in the case
1269 of a service provider) it will no longer accept assertions from the identity provider or (in the case of an
1270 identity provider) it will no longer issue assertions to the service provider **about the principal**. The receiving
1271 provider can perform any maintenance with the knowledge that the relationship represented by the name
1272 identifier has been terminated.

1273 If the service provider requests that its identifier for the principal be changed by including a <NewID> (or
1274 <NewEncryptedID>) element, the identity provider **MUST** include the element's content as the
1275 SPProvidedID when subsequently communicating to the service provider **regarding this principal**.

1276 If the identity provider requests that its identifier for the principal be changed by including a <NewID> (or
1277 <NewEncryptedID>) element, the service provider **MUST** use the element's content as the
1278 <saml:NameID> element content when subsequently communicating with the identity provider **regarding**
1279 **this principal**.

1280 New at [SAMLCore] Section 3.6.3, lines 2477, 2483, and 2486-2487 (note that E8 specifies additional
1281 changes to the original text shown here):

1282 If the <Terminate> element is included in the request, the requesting provider is indicating that (in the case
1283 of a service provider) it will no longer accept assertions from the identity provider or (in the case of an
1284 identity provider) it will no longer issue assertions to the service provider **using that identifier**. The receiving
1285 provider can perform any maintenance with the knowledge that the relationship represented by the name
1286 identifier has been terminated.

1287 If the service provider requests that its identifier for the principal be changed by including a <NewID> (or
1288 <NewEncryptedID>) element, the identity provider **MUST** include the element's content as the
1289 SPProvidedID when subsequently communicating to the service provider **using the primary identifier**.

1290 If the identity provider requests that its identifier for the principal be changed by including a <NewID> (or
1291 <NewEncryptedID>) element, the service provider **MUST** use the element's content as the
1292 <saml:NameID> element content when subsequently communicating with the identity provider **in any case**
1293 **where the identifier being changed would have been used**.

1294 New at [SAMLCore] Section 8.4.7, lines 3337-3339:

1295 The element's `SPNameQualifier` attribute, if present, MUST contain the unique identifier of the service
1296 provider or affiliation of providers for whom the identifier was generated (see Section 8.3.6). It MAY be
1297 omitted if the element is contained in a message intended only for consumption directly by the service
1298 provider, and the value would be the unique identifier of that service provider.

1299 ~~The element's `sSPProvidedID` attribute MUST contain the alternative identifier of the principal most~~
1300 ~~recently set by the service provider or affiliation, if any (see Section 3.6). If no such identifier has~~
1301 ~~been established, then the attribute MUST be omitted.~~

1302 Original at [SAMLProf] Section 4.5, lines 1319 and 1323:

1303 In the scenario supported by the Name Identifier Management profile, an identity provider has exchanged
1304 some form of **persistent** identifier for a principal with a service provider, allowing them to share a common
1305 identifier for some length of time. Subsequently, the identity provider may wish to notify the service provider
1306 of a change in the format and/or value that it will use to identify the same principal in the future. Alternatively
1307 the service provider may wish to attach its own "alias" for the principal in order to ensure that the identity
1308 provider will include it when communicating with it in the future **about the principal**. Finally, one of the
1309 providers may wish to inform the other that it will no longer issue or accept messages using a particular
1310 identifier. To implement these scenarios, a profile of the SAML Name Identifier Management protocol is
1311 used.

1312 New at [SAMLProf] Section 4.5, lines 1319 and 1323 (note that E12 specifies additional changes to the
1313 original text shown here):

1314 In the scenario supported by the Name Identifier Management profile, an identity provider has exchanged
1315 some form of **long-term** identifier (**including but not limited to identifiers with a Format of**
1316 **`urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`**) for a principal with a service
1317 provider, allowing them to share a common identifier for some length of time. Subsequently, the identity
1318 provider may wish to notify the service provider of a change in the format and/or value that it will use to
1319 identify the same principal in the future. Alternatively the service provider may wish to attach its own "alias"
1320 for the principal in order to ensure that the identity provider will include it when communicating with it in the
1321 future **using that identifier**. Finally, one of the providers may wish to inform the other that it will no longer
1322 issue or accept messages using a particular identifier. To implement these scenarios, a profile of the SAML
1323 Name Identifier Management protocol is used.

1324 **E56: Confirmation Method Typo**

1325 Change [SAMLProf] Section 3 at line 326 to change the reference from `<ConfirmationMethod>` (an
1326 element that no longer exists) to `Method` (an attribute, used instead of the element beginning in V2.0 of
1327 SAML).

1328 **E57: SAMLmime Reference**

1329 Change [SAMLBind] Section 4 at lines 1468-1469 to replace a reference to an expired IETF I-D for the
1330 SAMLmime definition to a persistent reference for the same definition.

1331 Original:

1332 [SAMLmime] **application/saml+xml Media Type Registration, IETF Internet-Draft,**
1333 **<http://www.ietf.org/internet-drafts/draft-hodges-saml-mediatype-01.txt>.**

1334 New:

1335 [SAMLmime] **OASIS Security Services Technical Committee (SSTC),**
1336 **"application/samlassertion+xml MIME Media Type Registration", IANA**
1337 **MIME Media Types Registry application/samlassertion+xml, December**
1338 **2004. See [http://www.iana.org/assignments/media-](http://www.iana.org/assignments/media-types/application/samlassertion+xml)**
1339 **types/application/samlassertion+xml.**

1340

E58: KeyDescriptor Typos in Profiles

1341 Change [SAMLProf] Section 4.1.6 at lines 626 and 627 to expand the keyword **sign** to **signing** and to
1342 expand the keyword **encrypt** to **encryption**. These were typographical errors.

1343 Original:

1344 The providers MAY document the key(s) used to sign requests, responses, and assertions with
1345 `<md:KeyDescriptor>` elements with a `use` attribute of **sign**. When encrypting SAML elements,
1346 `<md:KeyDescriptor>` elements with a `use` attribute of **encrypt** MAY be used to document supported
1347 encryption algorithms and settings, and public keys used to receive bulk encryption keys.

1348 New:

1349 The providers MAY document the key(s) used to sign requests, responses, and assertions with
1350 `<md:KeyDescriptor>` elements with a `use` attribute of **signing**. When encrypting SAML elements,
1351 `<md:KeyDescriptor>` elements with a `use` attribute of **encryption** MAY be used to document
1352 supported encryption algorithms and settings, and public keys used to receive bulk encryption keys.

1353

E59: SSO Response When Using HTTP-Artifact

1354 Change [SAMLBind] Section 3.6.5.2 at line 1173 to observe for clarity's sake that particular message
1355 delivery mechanisms are not mandated for the "nested" message exchange that takes place as part of
1356 the HTTP-Artifact binding.

1357 New:

1358 Note also that there is no mechanism defined to protect the integrity of the relationship between the artifact
1359 and the "RelayState" value, if any. That is, an attacker can potentially recombine a pair of valid HTTP
1360 responses by switching the "RelayState" values associated with each artifact. As a result, the
1361 producer/consumer of "RelayState" information MUST take care not to associate sensitive state information
1362 with the "RelayState" value without taking additional precautions (such as based on the information in the
1363 SAML protocol message retrieved via artifact).

1364 **Finally, note that the use of the `Destination` attribute in the root SAML element of the protocol
1365 message is unspecified by this binding, because of the message indirection involved.**

1366

E60: Incorrect URI for Unspecified NameID Format

1367 Change [SAMLCore] Section 2.2.2 at line 460 to change the name identifier format from
1368 `urn:oasis:names:tc:SAML:1.0:nameid-format:unspecified` to
1369 `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`. This was a typographical error.

1370

E61: Reference to Non-Existent Element

1371 Change [SAMLCore] Section 7.1.2 at lines 3160.

1372 Original:

1373 The following SAML protocol **elements** are intended specifically for use as extension points in an extension
1374 schema; **their types** are set to abstract, and are thus usable only as the base of a derived type:

- 1375 • **<Request>** and RequestAbstractType
- 1376 • **<SubjectQuery>** and SubjectQueryAbstractType

1377 New:

1378 The following SAML protocol **constructs** are intended specifically for use as extension points in an
1379 extension schema; **the types listed** are set to abstract, and are thus usable only as the base of a derived
1380 type:

- 1381 • RequestAbstractType

1382 • <SubjectQuery> and SubjectQueryAbstractType

1383 **E62: TLS Keys in KeyDescriptor**

1384 Change [SAMLMeta] Section 2.4.1.1 at line 624 to specify more clearly how to interpret the
1385 KeyDescriptor element's use attribute.

1386 New (just after the conclusion of the definition list for KeyDescriptorType):

1387 **A use value of "signing" means that the contained key information is applicable to both signing**
1388 **and TLS/SSL operations performed by the entity when acting in the enclosing role.**

1389 **A use value of "encryption" means that the contained key information is suitable for use in**
1390 **wrapping encryption keys for use by the entity when acting in the enclosing role.**

1391 **If the use attribute is omitted, then the contained key information is applicable to both of the above**
1392 **uses.**

1393 The following schema fragment defines the <KeyDescriptor> element and its KeyDescriptorType
1394 complex type:

1395 **E63: IdP Discovery Cookie Interpretation**

1396 Change [SAMLProf] Section 4.3.1 at line 1105 to clear up confusion over interpretation of the contents of
1397 an IdP Discovery cookie. (Note that E32 specifies changes to Section 4 that result in a new Section 4.3.1
1398 being inserted before the original one; E63 applies to the original Section 4.3.1.)

1399 New:

1400 Cookie syntax should be in accordance with IETF RFC 2965 [RFC2965] or [NSCookie]. The cookie MAY be
1401 either session-only or persistent. This choice may be made within a deployment, but should apply uniformly
1402 to all identity providers in the deployment. **Note that while a session-only cookie can be used, the intent**
1403 **of this profile is not to provide a means of determining whether a user actually has an active session**
1404 **with one or more of the identity providers stored in the cookie. The cookie merely identifies identity**
1405 **providers known to have been used in the past. Service providers MAY instead rely on the**
1406 **IsPassive attribute in their <samlp:AuthnRequest> message to probe for active sessions.**

1407 **E64: Liberty Moniker Used Inappropriately**

1408 Change [SAMLSec] Section 7.1.1.9, Impersonation without Reauthentication to replace an accidental use
1409 of the moniker "Liberty" in place of "SAML V2.0".

1410 New:

1411 Cookies posted by identity providers MAY be used to support this validation process, though **LibertySAML**
1412 **V2.0** does not mandate a cookie-based approach.

1413 **E65: Second-level StatusCode**

1414 Change various sections as follows in [SAMLCore] to constrain the optional second-level <StatusCode>
1415 element used, and clarify that use of second-level codes is optional.

1416 Change section 3.3.2.2.1, lines 1817-1819.

1417 New:

1418 **If none of the specified classes or declarations can be satisfied in accordance with the rules below, then the**
1419 **responder MUST return a <Response> message with a top-level <StatusCode> value of**
1420 **urn:oasis:names:tc:SAML:2.0:status:Responder and MAY return a second-level**
1421 **<StatusCode> of urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext.**

1422 Change section 3.4.1.2, lines 2172-2173.

1423 New:

1424 In profiles specifying an active intermediary, the intermediary MAY examine the list and return a
1425 <Response> message with an error <Status> and **optionally** a second-level <StatusCode> of

1426 Change section 3.4.1.5.1, lines 2282-2285.

1427 Original:

1428 An identity provider MUST NOT proxy a request where <ProxyCount> is set to zero. The identity
1429 provider MUST return an error <Status> containing a second-level <StatusCode> value of
1430 urn:oasis:names:tc:SAML:2.0:status:ProxyCountExceeded, unless it can directly
1431 authenticate the presenter.

1432 New:

1433 **Unless the identity provider can directly authenticate the presenter, it MUST return a <Response>**
1434 **message with a top-level <StatusCode> value of**
1435 **urn:oasis:names:tc:SAML:2.0:status:Responder and MAY return a second-level**
1436 **<StatusCode> value of urn:oasis:names:tc:SAML:2.0:status:ProxyCountExceeded.**

1437 Change section 3.8.3, lines 2729-2731.

1438 New:

1439 If the responder does not recognize the principal identified in the request, it MAY respond with an error
1440 <Status>, **optionally** containing a second-level <StatusCode> of
1441 urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal.

1442 **E66: Metadata and DNSSEC**

1443 Change [SAMLMeta] to update the DNSSEC reference from RFC 2535 to RFC 4035.

1444 Updated line 1253:

1445 It is RECOMMENDED that entities publish their resource records in signed zone files using ~~[RFC2535]~~
1446 **[RFC4035]**

1447 Original at lines 1447-1448:

1448 [RFC2535] D. Eastlake. *Domain Name System Security Extensions*. IETF RFC 2535, March 1999. See
1449 <http://www.ietf.org/rfc/rfc2535.txt>.

1450 New at lines 1447-1448:

1451 **[RFC4035] R. Arends et al. *Protocol Modifications for the DNS Security Extensions*. IETF RFC 4035,**
1452 **March 2005. See <http://www.ietf.org/rfc/rfc4035.txt>.**

1453 **E68: Use of Multiple <KeyDescriptor> Elements**

1454 Add text to section 2.4.1.1 of [SAMLMeta] to clarify the meaning of identically-purposed
1455 <KeyDescriptor> elements within a role.

1456 New at line 625:

1457 **The inclusion of multiple <KeyDescriptor> elements with the same use attribute (or no such**
1458 **attribute) indicates that any of the included keys may be used by the containing role or affiliation. A**
1459 **relying party SHOULD allow for the use of any of the included keys. When possible the signing or**
1460 **encrypting party SHOULD indicate as specifically as possible which key it used to enable more**
1461 **efficient processing.**

1462 The following schema fragment defines the <KeyDescriptor> element and its KeyDescriptorType
1463 complex type:

1464 **E69: Semantics of <ds:KeyInfo> in <KeyDescriptor>**

1465 Add text to section 2.4.1.1 of [SAMLMeta] to clarify the limitations of the specification regarding the
1466 semantics of various kinds of common key representations.

1467 New at line 625 (this change should appear after E68 above):

1468 **The <ds:KeyInfo> element is a highly generic and extensible means of communicating key**
1469 **material. This specification takes no position on the allowable or suggested content of this element,**
1470 **nor on its meaning to a relying party. As a concrete example, no implications of including an X.509**
1471 **certificate by value or reference are to be assumed. Its validity period, extensions, revocation status,**
1472 **and other relevant content may or may not be enforced, at the discretion of the relying party. The**
1473 **details of such processing, and their security implications, are out of scope; they may, however, be**
1474 **addressed by other SAML profiles.**

1475 The following schema fragment defines the <KeyDescriptor> element and its KeyDescriptorType
1476 complex type:

1477 **E70: Obsolete reference to UUID URN namespace**

1478 Change [SAMLProf] to update the Internet Draft reference for the UUID URN namespace to RFC 4122.

1479 Updated Section 8.3.3.1, line 1836:

1480 values are equal in the sense of [~~http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-05.txt~~][**RFC4122**].
1481 The

1482 Updated Section 8.4.3.1, line 1885:

1483 values are equal in the sense of [~~http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-05.txt~~][**RFC4122**].
1484 The

1485 Original at lines 2111-2112:

1486 [Mealling] P Leach et al. *A UUID URN Namespace*. IETF Internet-Draft, December 2004. See
1487 <http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-05.txt>.

1488 New at lines 2111-2112:

1489 [**RFC4122**] P. Leach et al. *A Universally Unique Identifier (UUID) URN Namespace*. IETF RFC 4122,
1490 July 2005. See <http://www.ietf.org/rfc/rfc4122.txt>.

1491 **E71: Missing namespace definition in Profiles**

1492 Change [SAMLProf] to add the "xs" namespace prefix to the table in Section 1.

1493 New row of table in Section 1, between lines 267-268:

1494 **xs :**

1495 **<http://www.w3.org/2001/XMLSchema>**

1496 **This namespace is defined in the W3C XML Schema specification [Schema1]. In schema listings, this**
1497 **is the default namespace and no prefix is shown. For clarity, the prefix is generally shown in**
1498 **specification text when XML Schema-related constructs are mentioned.**

1499 **E74: Update XML Signature Reference**

1500 Update the XML Signature specification reference in [SAMLCore], [SAMLBind], [SAMLProf], [SAMLMeta],
1501 [SAMLAuthCtx], [SAMLConf], [SAMLSec] to the "Second Edition". Also remove a stale non-normative
1502 reference in [SAMLCore].

1503 Strike [SAMLCore], lines 3439-3440:

1504 [RFC 3075] D. Eastlake, J. Reagle, D. Solo. *XML Signature Syntax and Processing*. IETF RFC 3075,
1505 March 2001. See <http://www.ietf.org/rfc/rfc3075.txt>.

1506 Original at [SAMLCore] lines 3415-3416, [SAMLBind] lines 1489-1491, [SAMLProf] lines 2205-2206,
1507 [SAMLMeta] lines 1490-1491, [SAMLAuthCtx] lines 3926-3928, [SAMLConf] lines 410-412, [SAMLSec] lines
1508 1078-1079:

1509 [XMLSig] D. Eastlake et al. *XML Signature Syntax and Processing*. World Wide Web Consortium,
1510 February 2002. See <http://www.w3.org/TR/xmldsig-core/>. Note that this specification normatively
1511 references [XMLSig-XSD], listed below.

1512 New at [SAMLCore] lines 3415-3416, [SAMLBind] lines 1489-1491, [SAMLProf] lines 2205-2206,
1513 [SAMLMeta] lines 1490-1491, [SAMLAuthCtx] lines 3926-3928, [SAMLConf] lines 410-412, [SAMLSec]
1514 lines 1078-1079:

1515 [XMLSig] D. Eastlake et al. *XML Signature Syntax and Processing, Second Edition*. World
1516 Wide Web Consortium, June 2008. See <http://www.w3.org/TR/xmldsig-core/>.

1517 **E75: Clarify Handling of SubjectConfirmation in AuthnRequest**

1518 Change [SAMLCore] Section 3.4.1.4 to clarify an identity provider's obligation to return an error if can't
1519 honor the requirements of a <SubjectConfirmation> element in an <AuthnRequest> message.

1520 New at line 2247:

1521 In such a case, the identifier's physical content MAY be different, but it MUST refer to the same principal. **If**
1522 **the identity provider cannot or will not produce assertions with a strongly matching subject, then it**
1523 **MUST return a <Response> with an error <Status>, and MAY return a second-level <StatusCode>**
1524 **that reflects the reason for the failure.**

1525 **E76: Clarify nested validUntil/cacheDuration**

1526 Add text to [SAMLMeta] to clarify the processing of nested `validUntil` or `cacheDuration` attributes.

1527 New in Sections 2.3.1 and 2.3.2, before lines 336 and 409:

1528 When not used as the root element of a metadata instance, a `validUntil` or `cacheDuration` attribute
1529 MAY be used to impose a shorter expiration or cache duration than that of the parent or root element, but
1530 never a longer one; the smaller value takes precedence.

1531 New in Sections 2.4.1 and 2.5, before lines 589 and 972:

1532 A `validUntil` or `cacheDuration` attribute MAY be used to impose a shorter expiration or cache duration
1533 than that of the parent or root element, but never a longer one; the smaller value takes precedence.

1534 **E77: Generalize scope of Metadata specification**

1535 Change [SAMLMeta] to address inadvertent language appearing to restrict use of SAML metadata to only
1536 SAML profiles.

1537 New in Section 1, before line 137:

1538 A variety of extension points are also included to allow for the use of SAML metadata in non-SAML
1539 specifications, profiles, and deployments, and such use is encouraged.

1540 Updated Section 2, lines 153-154:

1541 SAML metadata is organized around an extensible collection of roles representing common combinations of
1542 SAML (and potentially non-SAML) protocols and profiles supported by system entities.

1543 Remove the word "SAML" from lines 226, 230, 311, 323, 332, 360, 372, 397, 403, 444, 478, 531, and
1544 940.

1545

E78: Reassignment of persistent identifiers

1546 Add text to [SAMLCore] Section 8.3.7, at line 3325, to clarify that non-reassignment to different principals
1547 is a required property of "persistent" name identifiers.

1548 New:

1549 **Persistent name identifier values MUST NOT exceed a length of 256 characters. A given value, once**
1550 **associated with a principal, MUST NOT be assigned to a different principal at any time in the future.**

1551

E79: Clarification of SessionNotOnOrAfter

1552 Change [SAMLCore] Section 2.7.2, lines 1062-1065 to loosen wording around the
1553 `SessionNotOnOrAfter` attribute and defer more explicitly to profiles.

1554 Original:

1555 Specifies a time instant at which the session between the principal identified by the subject and the SAML
1556 authority issuing this statement MUST be considered ended. The time value is encoded in UTC, as
1557 described in Section 1.3.3. There is no required relationship between this attribute and a `NotOnOrAfter`
1558 condition attribute that may be present in the assertion.

1559 New:

1560 **Indicates an upper bound on sessions with the subject derived from the enclosing assertion. The**
1561 **time value is encoded in UTC, as described in Section 1.3.3. There is no required relationship between this**
1562 **attribute and a `NotOnOrAfter` condition attribute that may be present in the assertion. It's left to profiles**
1563 **to provide specific processing rules for relying parties based on this attribute.**

1564

E81: Algorithm statement in XML Signature profile

1565 Change [SAMLCore] Section 5.4.1, lines 2926-2927, and [SAMLMeta] Section 3.1.1, lines 1182-1183, to
1566 relax the implication that RSA with SHA1 is the only supported algorithm.

1567 Original:

1568 SAML processors SHOULD support the use of RSA signing and verification for public key operations in
1569 accordance with the algorithm identified by <http://www.w3.org/2000/09/xmldsig#rsa-sha1>.

1570 New:

1571 **Any algorithm defined for use with the XML Signature specification MAY be used.**

1572

E82: Empty <ContactPerson> element

1573 Add text to [SAMLMeta] Section 2.3.2.2, before line 500, to clarify that child elements should be included.

1574 New:

1575 **At least one child element SHOULD be present in a <ContactPerson> element.**

1576

E83: Weaken claim made about Exclusive C14N

1577 Change [SAMLCore] Section 5.4.3, lines 2939-2940, and [SAMLMeta] Section 3.1.3, lines 1196-1197, to
1578 better explain the purpose of using exclusive canonicalization.

1579 Original:

1580 Use of Exclusive Canonicalization ensures that signatures created over SAML messages embedded in an
1581 XML context can be verified independent of that context.

1582 New:

1583 Use of Exclusive Canonicalization facilitates the verification of signatures created over SAML messages
1584 when placed into a different XML context than present during signing.

1585 Note that use of this algorithm alone does not guarantee that a particular signed object can be moved from
1586 one context to another safely, nor is that a requirement of signed SAML objects in general, though it MAY be
1587 required by particular profiles.

1588

3 Acknowledgments

1589

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

1590

1591

- Rob Philpott, EMC Corporation

1592

- Richard Franck, IBM

1593

- John Bradley, Individual

1594

- Scott Cantor, Internet2

1595

- Nate Klingenstein, Internet2

1596

- Bob Morgan, Internet2

1597

- Thomas Hardjono, M.I.T.

1598

- Tom Scavo, National Center for Supercomputing Applications (NCSA)

1599

- Frederick Hirsch, Nokia Corporation

1600

- Paul Madsen, NTT Corporation

1601

- Ari Kermaier, Oracle Corporation

1602

- Hal Lockhart, Oracle Corporation

1603

- Anil Saldhana, Red Hat

1604

- Kent Spaulding, Skyworth TTG Holdings Limited

1605

- Duane DeCouteau, Veterans Health Administration

1606

- David Staggs, Veterans Health Administration

1607

The editors also would like to gratefully acknowledge **Jahan Moreh** of Sigaba and **Eve Maler** of PayPal, who during their tenures on the TC were editors of the errata working document and made major substantive contributions to all of the errata materials.

1608

1609