



# SAML V2.0 Errata

## Approved Errata Working Draft 51 16 October 2009

### Specification URIs:

#### This Version:

TBD

#### 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:

TBD

#### 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.

TC members should send comments on this specification to the TC's email list.

Others should send comments to the TC by using the "Send A Comment" button on the TC's web page at <http://www.oasis-open.org/committees/security>.

38 For information on whether any patents have been disclosed that may be essential to  
39 implementing this specification, and any offers of patent licensing terms, please refer to the IPR  
40 section of the TC web page (<http://www.oasis-open.org/committees/security/ipr.php>.  
41 The non-normative errata page for this specification is located at [http://www.oasis-  
open.org/committees/security](http://www.oasis-<br/>42 open.org/committees/security).

---

## 43 Notices

44 Copyright © OASIS® 1993–2009. All Rights Reserved. OASIS trademark, IPR and other policies apply.  
45 All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual  
46 Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

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

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

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

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

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

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

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

## Table of Contents

88	1 Introduction.....	6
89	1.1 Normative References.....	6
90	2 Approved Errata.....	8
91	E0: Incorrect Section Reference.....	8
92	E1: Relay State for HTTP Redirect.....	8
93	E2: Metadata Clarifications for HTTP Artifact Binding.....	8
94	E4: No Role for SAML V1.1 Artifacts in SAML V2.0.....	8
95	E6: Clarify Constraints on Encrypted NameID.....	9
96	E7: Metadata for Agreeing to Sign Authentication Requests.....	9
97	E8: SLO and NameID Termination .....	10
98	E10: Logout Request Reason Mismatch with Schema .....	10
99	E11: Improperly Labeled Feature.....	10
100	E12: Clarification on ManageNameIDRequest.....	10
101	E13: Inaccurate Description of Authorization Decision .....	11
102	E14: AllowCreate.....	11
103	E15: NameID Policy Adherence.....	13
104	E17: Authentication Response IssuerName vs. Assertion IssuerName.....	13
105	E18: Reference to Identity Provider Discovery Service in ECP Profile.....	14
106	E19: Clarification on Error Processing.....	14
107	E20: ECP SSO Profile and Metadata.....	14
108	E21: PAOS Version.....	15
109	E22: Error in Profile/ECP.....	15
110	E24: HTTPS in URI Binding.....	15
111	E25: Metadata Feature in Conformance.....	15
112	E26: Ambiguities Around Multiple Assertions and Statements in the SSO Profile.....	16
113	E27: Incorrect Step Number in ECP Profile.....	19
114	E28: Profile Labeling in Conformance.....	19
115	E29: Incomplete Listing of Features in Conformance.....	19
116	E30: Key Replacement.....	19
117	E31: Various Minor Errors in Binding.....	19
118	E32: Missing Required Information in Profiles.....	20
119	E33: References to Assertion Request Protocol.....	20
120	E34: RequestedAttribute Section Heading.....	20
121	E35: Response Consumer URL Rules and Example.....	20
122	E36: Clarification on Action Element.....	21
123	E37: Clarification in Metadata on Indexed Endpoints.....	21
124	E38: Clarification Regarding Index on <LogoutRequest>.....	21
125	E39: Error in SAML Profile Example.....	22
126	E40: Holder of Key.....	22
127	E41: EndpointType ResponseLocation Clarification in Metadata.....	23
128	E42: Match Authorities to Queries in Conformance.....	23

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

---

# 1 Introduction

167

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

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

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

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

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

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

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

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

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

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

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

## 1.1 Normative References

197

198 In general, the latest revisions of all errata-related documents will be linked from the TC home page at  
199 [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=security](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security).

200 **[SAMLAuthCtx]** J. Kemp et al. *Authentication Context for the OASIS Security Assertion Markup*  
201 *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)  
202 [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).

203 **[SAMLBind]** S. Cantor et al. *Bindings for the OASIS Security Assertion Markup Language*  
204 *(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)  
205 [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).

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

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

212       **[SAMLErrWork]**    S Cantor. *Errata Working Document for SAML V2.0*. OASIS SSTC, October  
213                           2009. Revision 51 corresponds to this Working Draft; see [http://www.oasis-  
215       \*\*\[SAMLMeta\]\*\*        S. Cantor et al. \*Metadata for the OASIS Security Assertion Markup Language\*  
216                           \(SAML\) V2.0. OASIS SSTC, March 2005. See \[http://docs.oasis-  
218       \\*\\*\\[SAMLProf\\]\\*\\*        J. Hughes et al. \\*Profiles for the OASIS Security Assertion Markup Language\\*  
219                           \\(SAML\\) V2.0. OASIS SSTC, March 2005. See \\[http://docs.oasis-  
221       \\\*\\\*\\\[SAMLSec\\\]\\\*\\\*         F. Hirsch et al. \\\*Security Considerations for the OASIS Security Assertion Markup\\\*  
222                           \\\*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-<br/>223                           open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf\\\).\\]\\(http://docs.oasis-<br/>220                           open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf\\)\]\(http://docs.oasis-<br/>217                           open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf\)](http://www.oasis-<br/>214                           open.org/committees/download.php/34737/sstc-saml-errata-2.0-draft-51.pdf)

---

## 224 2 Approved Errata

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

### 226 E0: Incorrect Section Reference

227 Change [SAMLCore] at line 2660 to refer to section 3.7.3 rather than 3.6.3 for Reason codes. This was a  
228 typographical error.

### 229 E1: Relay State for HTTP Redirect

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

233 Original:

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

239 New:

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

### 243 E2: Metadata Clarifications for HTTP Artifact Binding

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

246 Original:

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

251 New:

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

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

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

259 New:

260 The following describes the single artifact type defined by SAML V2.0. **Although the general artifact  
261 structure resembles that used in prior versions of SAML and the type code of the single format  
262 described below does not conflict with previously defined formats, there is explicitly no  
263 correspondence between SAML V2.0 artifacts and those found in any previous specifications, and**

264  
265

artifact formats not defined specifically for use with SAML V2.0 MUST NOT be used with this binding.

266

## E6: Clarify Constraints on Encrypted NameID

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

269 New:

270  
271  
272  
273  
274  
275  
276

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

277

## E7: Metadata for Agreeing to Sign Authentication Requests

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

281 New at line 710:

282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295

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

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

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

296 New at lines 741-742:

297  
298  
299  
300  
301  
302

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

303 New at lines 744-747:

304  
305  
306  
307  
308

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

## E8: SLO and NameID Termination

309

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

312 Original:

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

316 New:

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

## E10: Logout Request Reason Mismatch with Schema

326

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

331 New:

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

## E11: Improperly Labeled Feature

335

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

337 Original labels:

338 Name Identifier Management, HTTP Redirect (IdP-initiated)  
339 Name Identifier Management, SOAP (IdP-initiated)  
340 Name Identifier Management, HTTP Redirect  
341 Name Identifier Management, SOAP

342 New labels:

343 **Name Identifier Management (IdP-Initiated), HTTP Redirect**  
344 **Name Identifier Management (IdP-Initiated), SOAP**  
345 **Name Identifier Management (SP-Initiated), HTTP Redirect**  
346 **Name Identifier Management (SP-Initiated), SOAP**

## E12: Clarification on ManageNameIDRequest

347

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

351 New [SAMLCore] at lines 2412-2413:

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

356 New [SAMLCore] at line 2438:

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

361 New [SAMLProf] at lines 1320-23121:

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

## 364 **E13: Inaccurate Description of Authorization Decision**

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

367 New:

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

## 370 **E14: AllowCreate**

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

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

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

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

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

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

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

391  
392 **When "false", the requester tries to constrain the identity provider to issue an assertion only if such**  
393 **state has already been established or is not deemed applicable by the identity provider to the use of**  
394 **an identifier. Thus, this does not prevent the identity provider from assuming such information**  
395 **exists outside the context of this specific request (for example, establishing it in advance for a large**  
396 **number of principals).**

397  
398 **A value of "true" permits the identity provider to take any related actions it wishes to fulfill the**

399 request, subject to any other constraints imposed by the request and policy (the `IsPassive`  
400 attribute, for example).  
401  
402 Generally, requesters cannot assume specific behavior from identity providers regarding the initial  
403 creation or association of identifiers on their behalf, as these are details left to implementations or  
404 deployments. Absent specific profiles governing the use of this attribute, it might be used as a hint  
405 to identity providers about the requester's intention to store the identifier or link it to a local value.  
406  
407 A value of "false" might be used to indicate that the requester is not prepared or able to do so and  
408 save the identity provider wasted effort.  
409  
410 Requesters that do not make specific use of this attribute SHOULD generally set it to "true" to  
411 maximize interoperability.  
412  
413 The use of the `AllowCreate` attribute MUST NOT be used and SHOULD be ignored in conjunction  
414 with requests for or assertions issued with name identifiers with a `Format` of  
415 `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` (they preclude any such state in  
416 and of themselves).

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

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

421  
422 **Note that this protocol is typically not used with "transient" name identifiers, since their value is not**  
423 **intended to be managed on a long-term basis.**

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

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

428  
429 **This protocol MUST NOT be used in conjunction with the**  
430 **`urn:oasis:names:tc:SAML:2.0:nameidformat:transient` <NameID> Format.**

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

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

439  
440 **If the receiving provider is maintaining state associated with the name identifier, such as the value of**  
441 **the identifier itself (in the case of a pair-wise identifier), an `SPProvidedID` value, the sender's**  
442 **consent to the identifier's creation/use, etc., then the receiver can perform any maintenance with the**  
443 **knowledge that the relationship represented by the name identifier has been terminated.**

444  
445 **Any subsequent operations performed by the receiver on behalf of the sender regarding the**  
446 **principal (for example, a subsequent <AuthnRequest>) SHOULD be carried out in a manner**  
447 **consistent with the absence of any previous state.**

448  
449 **Termination is potentially the cleanup step for any state management behavior triggered by the use**  
450 **of the `AllowCreate` attribute in the Authentication Request protocol (see Section 3.4). Deployments**  
451 **that do not make use of that attribute are likely to avoid the use of the <Terminate> element or**  
452 **would treat it as a purely advisory matter.**

453  
454 **Note that in most cases (a notable exception being the rules surrounding the `SPProvidedID`**

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

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

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

463  
464 If the service provider wishes to permit the identity provider to establish a new identifier for the  
465 principal if none exists, it MUST include a <NameIDPolicy> element with the AllowCreate attribute  
466 set to "true". Otherwise, only a principal for whom the identity provider has previously established  
467 an identifier usable by the service provider can be authenticated successfully.

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

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

471  
472 This profile does not provide any guidelines for the use of AllowCreate; see [SAMLCore] for  
473 normative rules on using AllowCreate.

## 474 E15: NameID Policy Adherence

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

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

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

482  
483 When a Format defined in Section Error: Reference source not found 8.3 other than  
484 urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified or  
485 urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted is used, then if the identity provider  
486 returns any assertions:

487  
488 ● the Format value of the <NameID> within the <Subject> of any <Assertion> MUST be identical  
489 to the Format value supplied in the <NameIDPolicy>, and

490  
491 ● if SPNameQualifier is not omitted in <NameIDPolicy>, the SPNameQualifier value of the  
492 <NameID> within the <Subject> of any <Assertion> MUST be identical to the SPNameQualifier  
493 value supplied in the <NameIDPolicy>.

## 494 E17: Authentication Response IssuerName vs. Assertion 495 IssuerName

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

498 Original:

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

502 New:

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

## 507 **E18: Reference to Identity Provider Discovery Service in ECP** 508 **Profile**

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

511 New:

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

## 515 **E19: Clarification on Error Processing**

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

518 Original at Section 3.2.2.1, lines 310-317:

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

526 New at Section 3.2.2.1, lines 310-317:

527 The SAML responder **SHOULD** return a **SOAP message containing either a SAML response element in**  
528 **the body or a SOAP fault**. The SAML responder **MUST NOT** include more than one SAML response per  
529 SOAP message or include any additional XML elements in the SOAP body. SOAP fault codes **SHOULD**  
530 **NOT** be sent for errors within the SAML problem domain, for example, inability to find an extension schema  
531 or as a signal that the subject is not authorized to access a resource in an authorization query. **See Section**  
532 **3.2.3.3 for more information about error handling**. (SOAP 1.1 faults and fault codes are discussed in  
533 [SOAP11] Section 4.1.)

534 Original at Section 3.2.3.3, line 378:

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

537 New at Section 3.2.3.3, line 378:

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

## 540 **E20: ECP SSO Profile and Metadata**

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

543 New (small portion of previous subsection shown):

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

547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557

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

558

## E21: PAOS Version

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

560  
561

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

562

## E22: Error in Profile/ECP

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

565

## E24: HTTPS in URI Binding

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

568 Original:

569  
570  
571

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

572 New:

573  
574

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

575

## E25: Metadata Feature in Conformance

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

578 New in Table 2:

579  
580  
581

Feature	IdP	IdP Lite	SP	SP Lite	ECP
<b>Metadata Structures</b>	<b>OPT</b>	<b>OPT</b>	<b>OPT</b>	<b>OPT</b>	<b>N/A</b>
<b>Metadata Interoperation</b>	<b>OPT</b>	<b>OPT</b>	<b>OPT</b>	<b>OPT</b>	<b>N/A</b>

582 New in Table 4:

583  
584  
585

Feature	Authn	Attrib	Authz	Requester
<b>Metadata Structures</b>	<b>OPT</b>	<b>OPT</b>	<b>OPT</b>	<b>OPT</b>
<b>Metadata Interoperation</b>	<b>OPT</b>	<b>OPT</b>	<b>OPT</b>	<b>OPT</b>

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

587  
588

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

### 3.6 Metadata Structures

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

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

- Implementing SAML metadata according to the extensible SAML V2.0 Metadata format in all cases where an interoperating peer has the option, as stated in SAML V2.0 specifications, of depending on the existence of SAML V2.0 Metadata. Electing the Metadata Structures option has the effect of requiring that such metadata be available to the interoperating peer. The Metadata Interoperation feature, described below, provides a means of satisfying this requirement.
- Referencing, consuming, and adhering to the SAML metadata, according to [SAMLMeta], of an interoperating peer when the known metadata relevant to that peer and the particular operation, and the current exchange, has expired or is no longer valid in cache, provided the metadata is available and is not prohibited by policy or the particular operation and that specific exchange.

### 3.7 Metadata Interoperation

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

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

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

Original at Section 4.1.4.2, lines 541-572:

- The <Issuer> element MAY be omitted, but if present it MUST contain the unique identifier of the issuing identity provider; the Format attribute MUST be omitted or have a value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- It MUST contain at least one <Assertion>. Each assertion's <Issuer> element MUST contain the unique identifier of the **issuing** identity provider; the Format attribute MUST be omitted or have a value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- **The set of one or more assertions MUST contain at least one <AuthnStatement> that reflects the authentication of the principal to the identity provider.**
- **At least one assertion containing an <AuthnStatement> MUST contain a <Subject> element with at least one <SubjectConfirmation> element containing a Method of urn:oasis:names:tc:SAML:2.0:cm:bearer. If the identity provider supports the Single Logout profile, defined in Section 4.4, any such authentication statements MUST include a SessionIndex attribute to enable per-session logout requests by the service provider.**
- **The bearer <SubjectConfirmation> element described above MUST contain a <SubjectConfirmationData> element that contains a Recipient attribute containing the service provider's assertion consumer service URL and a NotOnOrAfter attribute that limits the window during which the assertion can be delivered. It MAY contain an Address attribute limiting the client address from which the assertion can be delivered. It MUST NOT contain a NotBefore attribute. If the containing message is in response to an <AuthnRequest>, then the InResponseTo attribute MUST match the request's ID.**
- Other statements and confirmation methods MAY be included in the assertion(s) at the discretion of the identity provider. In particular, <AttributeStatement> elements MAY be included. The

641 <AuthnRequest> MAY contain an AttributeConsumingServiceIndex XML attribute referencing  
642 information about desired or required attributes in [SAMLMeta]. The identity provider MAY ignore this, or  
643 send other attributes at its discretion.

- 644 • **The assertion(s) containing a bearer subject confirmation** MUST contain an  
645 <AudienceRestriction> including the service provider's unique identifier as an <Audience>.
- 646 • Other conditions (and other <Audience> elements) MAY be included as requested by the service  
647 provider or at the discretion of the identity provider. (Of course, all such conditions MUST be understood  
648 by and accepted by the service provider in order for the assertion to be considered valid.) The identity  
649 provider is NOT obligated to honor the requested set of <Conditions> in the <AuthnRequest>, if  
650 any.
- 651 • The identity provider is NOT obligated to honor the requested set of <Conditions> in the  
652 <AuthnRequest>, if any.

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

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

691 required attributes in [SAMLMeta]. The identity provider MAY ignore this, or send other attributes at its  
692 discretion.

- 693 • **Each bearer** assertion MUST contain an <AudienceRestriction> including the service provider's  
694 unique identifier as an <Audience>.
- 695 • Other conditions (and other <Audience> elements) MAY be included as requested by the service  
696 provider or at the discretion of the identity provider. (Of course, all such conditions MUST be understood  
697 by and accepted by the service provider in order for the assertion to be considered valid.) The identity  
698 provider is NOT obligated to honor the requested set of <Conditions> in the <AuthnRequest>, if  
699 any.
- 700 • The identity provider is NOT obligated to honor the requested set of <Conditions> in the  
701 <AuthnRequest>, if any.

702 Original at Section 4.1.4.3, lines 576-591:

- 703 • Verify that the Recipient attribute in any bearer <SubjectConfirmationData> matches the assertion  
704 consumer service URL to which the <Response> or artifact was delivered
- 705
- 706 • Verify that the NotOnOrAfter attribute in any bearer <SubjectConfirmationData> has not passed,  
707 subject to allowable clock skew between the providers
- 708
- 709 • Verify that the InResponseTo attribute in the bearer <SubjectConfirmationData> equals the ID of  
710 its original <AuthnRequest> message, unless the response is unsolicited (see Section 4.1.5 ), in which  
711 case the attribute MUST NOT be present
- 712 • Verify that any assertions relied upon are valid in other respects.
- 713 • If any bearer <SubjectConfirmationData> includes an Address attribute, the service provider MAY  
714 check the user agent's client address against it.
- 715 • Any assertion which is not valid, or whose subject confirmation requirements cannot be met SHOULD be  
716 discarded and SHOULD NOT be used to establish a security context for the principal.
- 717 • If an <AuthnStatement> used to establish a security context for the principal contains a  
718 SessionNotOnOrAfter attribute, the security context SHOULD be discarded once this time is reached,  
719 unless the service provider reestablishes the principal's identity by repeating the use of this profile.

720 New at Section 4.1.4.3, lines 576-591:

- 721 • Verify that the Recipient attribute in **the** bearer <SubjectConfirmationData> matches the assertion  
722 consumer service URL to which the <Response> or artifact was delivered
- 723
- 724 • Verify that the NotOnOrAfter attribute in **the** bearer <SubjectConfirmationData> has not passed,  
725 subject to allowable clock skew between the providers
- 726
- 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.

745 New at Section 4.1.4.5, lines 600-601:

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

## 749 E27: Incorrect Step Number in ECP Profile

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

## 752 E28: Profile Labeling in Conformance

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

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

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

## 759 E29: Incomplete Listing of Features in Conformance

760 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

## 764 E30: Key Replacement

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

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

768 New:

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

## 771 E31: Various Minor Errors in Binding

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

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

775 Original at Section 3.5.3, line 785:

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

778 New at Section 3.5.3, line 785:

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

781 Original at Section 3.6.5, line 1136:

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

785 New 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 **synchronous**  
788 SAML binding, as in step 3.

789 Original at Section 3.6.5, line 1397:

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

791 New at Section 3.6.5, line 1397:

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

## 793 **E32: Missing Required Information in Profiles**

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

### 796 **4.3.1 Required Information**

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

798 **Contact information:** [security-services-comment@lists.oasis-open.org](mailto:security-services-comment@lists.oasis-open.org)

799 **Description:** Given below.

800 **Updates:** None.

## 801 **E33: References to Assertion Request Protocol**

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

## 805 **E34: RequestedAttribute Section Heading**

806 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  
807 consistency in reflecting element nesting in the document outline.

## 808 **E35: Response Consumer URL Rules and Example**

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

811 Original at Section 4.2.4.1, lines 906-908:

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

816 New at lines Section 4.2.4.1, 906-908:

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

821 Original at Section 4.2.4.3, line 964:

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

824 New at Section 4.2.4.3, line 964:

```
825 <paos:Request xmlns:paos="urn:liberty:paos:2003-08"  
826   responseConsumerURL="  
827   https://ServiceProvider.example.com/ecp_assertion_consumer"
```

## 828 **E36: Clarification on Action Element**

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

832 Original:

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

837 New:

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

## 840 **E37: Clarification in Metadata on Indexed Endpoints**

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

842 Original:

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

845 New:

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

## 849 **E38: Clarification Regarding Index on <LogoutRequest>**

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

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

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

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

```
856 <SessionIndex> [Optional]
```

857 **The index of the session between the principal identified by the <saml:BaseID>, <saml:NameID>, 858 or <saml:EncryptedID> element, and the session authority. This must correlate to the 859 SessionIndex attribute, if any, in the <saml:AuthnStatement> of the assertion used to establish 860 the session that is being terminated.**

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

862 If the requester is a session participant, it MUST include at least one <SessionIndex> element in the 863 request. (Note that the session participant always receives a SessionIndex attribute in the 864 <saml:AuthnStatement> elements that it receives to initiate the session, per Section 4.1.4.2 of 865 the Web Browser SSO Profile.) If the requester is a session authority (or acting on its behalf), then it MAY 866 omit any such elements to indicate the termination of all of the principal's applicable sessions.

## 867 **E39: Error in SAML Profile Example**

868 **Note:** E39 corrects text in a section that is affected by E53, which deprecates the entire 869 section. Please see E53 for details.

870 Change [SAMLProf] Section 8.5.6 at lines 2095-2098 to move the ldaprof:Encoding attribute to the 871 correct location.

872 Original:

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

882 New:

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

## 892 **E40: Holder of Key**

893 Change [SAMLProf] Section 3.1 at lines 335-337 to align the description of Holder of Key in the profiles 894 specification with the language in the core specification.

895 Original:

896 As described in [XMLSig], each <ds:KeyInfo> element holds a key or information that enables an 897 application to obtain a key. The holder of a specified key is considered to be **the subject of** the assertion by 898 the asserting party.

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

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 **an acceptable attesting entity 902 for** the assertion by the asserting party.

## E41: EndpointType ResponseLocation Clarification in Metadata

903

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

906 New:

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

## E42: Match Authorities to Queries in Conformance

914

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

918 Original:

Feature	Authn	Attrib	Authz	Requester
Authentication Query, SOAP	MUST	OPT	OPT	OPT
Attribute Query, SOAP	OPT	MUST	OPT	OPT
Authorization Decision Query, SOAP	OPT	OPT	MUST	OPT

923 New:

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

## E43: Key Location in saml:EncryptedData

928

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

932 Original:

### 6.2 Combining Signatures and Encryption

933

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

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

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

942 New:

### 6.2 Key and Data Referencing Guidelines

943

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

946 Implementations of SAML MAY implicitly associate keys with the corresponding data they are used  
947 to encrypt, through the positioning of `<xenc:EncryptedKey>` elements next to the associated

948 <xenc:EncryptedData> element, within the enclosing SAML parent element. However, the  
 949 following set of explicit referencing guidelines are suggested to facilitate interoperability.

950 If the encrypted key is included in the XML instance, then it SHOULD be referenced within the  
 951 associated <xenc:EncryptedData> element, or alternatively embedded within the  
 952 <xenc:EncryptedData> element. When an <xenc:EncryptedKey> element is used, the  
 953 <ds:KeyInfo> element within <xenc:EncryptedData> SHOULD reference the  
 954 <xenc:EncryptedKey> element using a <ds:RetrievalMethod> element of Type  
 955 http://www.w3.org/2001/04/xmlenc#EncryptedKey.

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

959 In scenarios where the encrypted element is being “multicast” to multiple recipients, and the key  
 960 used to encrypt the message must be in turn encrypted individually and independently for each of  
 961 the multiple recipients, the <xenc:CarriedKeyName> element SHOULD be used to assign a  
 962 common name to each of the <xenc:EncryptedKey> elements so that a <ds:KeyName> can be  
 963 used from within the <xenc:EncryptedData> element’s <ds:KeyInfo> element.

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

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

972 **6.3 Examples**

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

```

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

1002 In the following <EncryptedAttribute> example, the <xenc:EncryptedKey> element is contained  
1003 within the <xenc:EncryptedData> element, so there is no explicit referencing:

```
1004 <saml:EncryptedAttribute  
1005   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">  
1006   <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"  
1007     Id="Encrypted_DATA_ID"  
1008     Type="http://www.w3.org/2001/04/xmlenc#Element">  
1009     <xenc:EncryptionMethod  
1010       Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>  
1011     <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">  
1012       <xenc:EncryptedKey Id="Encrypted_KEY_ID">  
1013         <xenc:EncryptionMethod  
1014           Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>  
1015         <xenc:CipherData>  
1016           <xenc:CipherValue>SDFSDF... </xenc:CipherValue>  
1017         </xenc:CipherData>  
1018       </xenc:EncryptedKey>  
1019     </ds:KeyInfo>  
1020     <xenc:CipherData>  
1021       <xenc:CipherValue>Nk4W4mx...</xenc:CipherValue>  
1022     </xenc:CipherData>  
1023   </xenc:EncryptedData>  
1024 </saml:EncryptedAttribute>
```

1025 The final example shows an assertion encrypted for multiple recipients, using the  
1026 <xenc:CarriedKeyName> approach:

```
1027 <saml:EncryptedAssertion  
1028   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">  
1029   <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"  
1030     Id="Encrypted_DATA_ID"  
1031     Type="http://www.w3.org/2001/04/xmlenc#Element">  
1032     <xenc:EncryptionMethod  
1033       Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>  
1034     <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">  
1035       <ds:KeyName>MULTICAST_KEY_NAME</ds:KeyName>  
1036     </ds:KeyInfo>  
1037     <xenc:CipherData>  
1038       <xenc:CipherValue>Nk4W4mx...</xenc:CipherValue>  
1039     </xenc:CipherData>  
1040   </xenc:EncryptedData>  
1041  
1042   <xenc:EncryptedKey xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"  
1043     Id="Encrypted_KEY_ID_1" Recipient="https://sp1.org">  
1044     <xenc:EncryptionMethod  
1045       Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>  
1046     <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">  
1047       <ds:KeyName>KEY_NAME_1</ds:KeyName>  
1048     </ds:KeyInfo>  
1049     <xenc:CipherData>  
1050       <xenc:CipherValue>xyzABC...</xenc:CipherValue>  
1051     </xenc:CipherData>  
1052     <xenc:ReferenceList>  
1053       <xenc:DataReference URI="#Encrypted_DATA_ID"/>  
1054     </xenc:ReferenceList>  
1055  
1056     <xenc:CarriedKeyName>MULTICAST_KEY_NAME</xenc:CarriedKeyName>  
1057   </xenc:EncryptedKey>  
1058  
1059   <xenc:EncryptedKey xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"  
1060     Id="Encrypted_KEY_ID_2" Recipient="https://sp2.org">  
1061     <xenc:EncryptionMethod  
1062       Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
```

```

1063 <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
1064   <ds:KeyName>KEY_NAME_2</ds:KeyName>
1065 </ds:KeyInfo>
1066 <xenc:CipherData>
1067   <xenc:CipherValue>abcXYZ...</xenc:CipherValue>
1068 </xenc:CipherData>
1069 <xenc:ReferenceList>
1070   <xenc:DataReference URI="#Encrypted_DATA_ID"/>
1071 </xenc:ReferenceList>
1072
1073   <xenc:CarriedKeyName>MULTICAST_KEY_NAME</xenc:CarriedKeyName>
1074 </xenc:EncryptedKey>
1075 </saml:EncryptedAssertion>

```

## E45: AuthnContext Comparison Order

1076

1077 Change [SAMLCore] Section 3.3.2.2.1 at lines 1815-1819 and 1826 to clarify the lack of orderedness in  
 1078 the comparison of a set of authentication contexts.

1079 Original at Section 3.3.2.2.1, lines 1815-1819:

1080 Either a set of class references or a set of declaration references can be used. The set of supplied  
 1081 references MUST be evaluated as an ordered set, where the first element is the most preferred  
 1082 authentication context class or declaration. If none of the specified classes or declarations can be satisfied in  
 1083 accordance with the rules below, then the responder MUST return a <Response> message with a second-  
 1084 level <StatusCode> of urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext.

1085 New at Section 3.3.2.2.1, lines 1815-1819:

1086 Either a set of class references or a set of declaration references can be used. **If ordering is relevant to**  
 1087 **the evaluation of the request, then** the set of supplied references MUST be evaluated as an ordered set,  
 1088 where the first element is the most preferred authentication context class or declaration. If none of the  
 1089 specified classes or declarations can be satisfied in accordance with the rules below, then the responder  
 1090 MUST return a <Response> message with a second-level <StatusCode> of  
 1091 urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext. **For example, ordering is significant**  
 1092 **when using this element in an <AuthnRequest> message but not in an <AuthnQuery> message.**

1093 Original at Section 3.3.2.2.1, line 1826:

1094 If Comparison is set to "better", then the resulting authentication context in the authentication statement  
 1095 MUST be stronger (as deemed by the responder) than **any** of the authentication contexts specified.

1096 New at Section 3.3.2.2.1, line 1826:

1097 If Comparison is set to "better", then the resulting authentication context in the authentication statement  
 1098 MUST be stronger (as deemed by the responder) than **one** of the authentication contexts specified.

## E46: AudienceRestriction Clarifications

1099

1100 Change [SAMLCore] Section 2.5.1.4 at lines 924-925 to clarify the logical sense with respect to individual  
 1101 audience elements within an audience-restriction condition grouping.

1102 Original:

1103 Note that multiple <AudienceRestriction> elements MAY be included in a single assertion, and each  
 1104 MUST be evaluated independently. The effect of this requirement and the preceding definition is that within  
 1105 a given **condition**, the **audiences** form a disjunction (an "OR") while multiple **conditions** form a conjunction  
 1106 (an "AND").

1107 New:

1108 Note that multiple <AudienceRestriction> elements MAY be included in a single assertion, and each  
 1109 MUST be evaluated independently. The effect of this requirement and the preceding definition is that within

1110 a given <AudienceRestrictions>, the <Audience> elements form a disjunction (an "OR") while  
1111 multiple <AudienceRestrictions> elements form a conjunction (an "AND").

## 1112 **E47: Clarification on SubjectConfirmation**

1113 Change [SAMLCore] Section 2.4.1.1 at line 698, and change [SAMLProf] Section 3.1 at lines 336 and 341  
1114 and Section 3.3 at lines 361-363, in order to clarify behavior around the subject confirmation element and  
1115 the intent of the embedded secondary identifier.

1116 New at [SAMLCore] Section 2.4.1.1, line 698 (add text just before the schema listing introduction):

1117 **If the <SubjectConfirmation> element in an assertion subject contains an identifier the issuer**  
1118 **authorizes the attesting entity to wield the assertion on behalf of that subject. A relying party MAY**  
1119 **apply additional constraints on the use of such an assertion at its discretion, based upon the**  
1120 **identities of both the subject and the attesting entity.**

1121 **If an assertion is issued for use by an entity other than the subject, then that entity SHOULD be**  
1122 **identified in the <SubjectConfirmation> element.**

1123 The following schema fragment defines the <SubjectConfirmation> element and its  
1124 SubjectConfirmationType complex type:

1125 Original at [SAMLProf] Section 3.1, line 336:

1126 As described in [XMLSig], each <ds:KeyInfo> element holds a key or information that enables an  
1127 application to obtain a key. The holder of a **specified key** is considered to be the subject of the assertion by  
1128 the asserting party.

1129 New at [SAMLProf] Section 3.1, line 336 (note that E40 specified additional changes to the original text  
1130 shown here):

1131 As described in [XMLSig], each <ds:KeyInfo> element holds a key or information that enables an  
1132 application to obtain a key. The holder of **one or more of the specified keys** is considered to be the subject  
1133 of the assertion by the asserting party.

1134 New at [SAMLProf] Section 3.1, line 341 (add text just before the example):

1135 **If the <SubjectConfirmation> element in an assertion subject contains an identifier the issuer**  
1136 **authorizes the attesting entity to wield the assertion on behalf of that subject. A relying party MAY**  
1137 **apply additional constraints on the use of such an assertion at its discretion, based upon the**  
1138 **identities of both the subject and the attesting entity.**

1139 **If an assertion is issued for use by an entity other than the subject, then that entity SHOULD be**  
1140 **identified in the <SubjectConfirmation> element.**

1141 Example: The holder of the key named "By-Tor" or the holder of the key named "Snow Dog" can confirm  
1142 itself as the subject.

1143 Original at [SAMLProf] Section 3.3, lines 361-363:

1144 The subject of the assertion is **the bearer of the assertion**, subject to optional constraints on confirmation  
1145 using the attributes that MAY be present in the <SubjectConfirmationData> element, as defined by  
1146 [SAMLCore].

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

1148 The subject of the assertion is **considered to be an acceptable attesting entity for the assertion by the**  
1149 **asserting party**, subject to optional constraints on confirmation using the attributes that MAY be present in  
1150 the <SubjectConfirmationData> element, as defined by [SAMLCore].

1151 **If the intended bearer is known by the asserting party to be an entity other than the subject, then the**  
1152 **asserting party SHOULD identify that entity to the relying party by including a SAML identifier**  
1153 **representing it in the enclosing <SubjectConfirmation> element.**

1154 **If multiple attesting entities are to be permitted to use the assertion based on bearer semantics, then**  
1155 **multiple <SubjectConfirmation> elements SHOULD be included.**

1156

## E48: Clarification on Encoding for Binary Values in LDAP Profile

1157

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

1158

1159 Change [SAMLProf] at line 1762. Original:

1160

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

1161

1162

1163

1164 New:

1165

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

1166

1167

1168

1169

1170

## E49: Clarification on Attribute Name Format

1171

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

1172

1173 New (add text to the end of the definition of <AttributeValue>):

1174

<AttributeValue> [Any Number]

1175

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

1176

1177

1178

1179

1180

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

1181

1182

1183

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

1184

1185

1186

1187

1188

## E50: Clarification on SSL Ciphersuites

1189

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

1190

1191 New at Section 4, line 235:

1192

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

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203 New at Section 5, line 257:

1204 In any SAML V2.0 use of SSL 3.0 [SSL3] or TLS 1.0 [RFC 2246], servers MUST authenticate to clients  
1205 using a X.509 v3 certificate. The client MUST establish server identity based on contents of the certificate  
1206 (typically through examination of the certificate's subject DN field). **The set of algorithms required for**  
1207 **SAML V2.0 conformance is equivalent to that defined in SAML V1.0 and SAML V1.1. These mandated**  
1208 **algorithms were chosen by the SSTC because of their wide implementation support in the industry.**  
1209 **While the algorithms defined below are the minimal set for SAML conformance, additional**  
1210 **algorithms supported by SSL 3.0 and TLS 1.0 MAY be used.**

## 1211 **E51: Schema Type of Contents of <AttributeValue>**

1212 Change [SAMLProf] Section 8.1.4 at line 1670 to change the reference from **Section 3.3** to **Section 3**, in  
1213 order to fix a typographical error that would have improperly restricted the valid types for attribute values  
1214 to derived types, rather than the larger category of built-in types.

## 1215 **E52: Clarification on NotOnOrAfter Attribute for Subject** 1216 **Confirmation**

1217 Change [SAMLProf] Section 4.1.4.2 at line 557 to correctly reflect the type of validity period that applies to  
1218 subject confirmation.

1219 Original:

1220 The bearer <SubjectConfirmation> element described above MUST contain a  
1221 <SubjectConfirmationData> element that contains a Recipient attribute containing the service  
1222 provider's assertion consumer service URL and a NotOnOrAfter attribute that limits the window during  
1223 which the assertion can be **delivered**. It MAY contain an Address attribute limiting the client address from  
1224 which the assertion can be delivered.

1225 New (note that E26 specifies additional changes to the original text shown here):

1226 The bearer <SubjectConfirmation> element described above MUST contain a  
1227 <SubjectConfirmationData> element that contains a Recipient attribute containing the service  
1228 provider's assertion consumer service URL and a NotOnOrAfter attribute that limits the window during  
1229 which the assertion can be **confirmed by the relying party**. It MAY contain an Address attribute limiting  
1230 the client address from which the assertion can be delivered.

## 1231 **E53: Correction to LDAP/X.500 Profile Attribute**

1232 Deprecate [SAMLProf] Section 8.2 at lines 1677-1799 by adding a notice after line 1677.

1233 New:

1234 **8.2 X.500/LDAP Attribute Profile – Deprecated**  
1235 **NOTE: This attribute profile is deprecated because of a flaw that makes it schema-invalid. The SSTC**  
1236 **has replaced it with a separately published SAML V2.0 X.500/LDAP Attribute Profile specification**  
1237 **that removes this flaw.**  
1238 Directories based on the ITU-T X.500 specifications [X.500] and the related IETF Lightweight Directory  
1239 Access Protocol specifications [LDAP] are widely deployed....

## 1240 **E54: Corrections to ECP URN**

1241 Change [SAMLProf] Section 4.2.3.1 at lines 757 and 763-764 to correct the usage of quotation marks in  
1242 HTTP headers.

1243 New at line 757 (add double quotation marks around the URN):

1244 Furthermore, support for this profile **MUST** be specified in the HTTP PAOS Header field as a service value,  
1245 with the value "urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp".

1246 Original at lines 763-764 (single quotation marks are problematic):

```
1247 GET /index HTTP/1.1  
1248 Host: identity-service.example.com  
1249 Accept: text/html; application/vnd.paos+xml  
1250 PAOS: ver='urn:liberty:paos:2003-08' ;  
1251 'urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp'
```

1252 New at lines 763-764 (double quotation marks used instead):

```
1253 GET /index HTTP/1.1  
1254 Host: identity-service.example.com  
1255 Accept: text/html; application/vnd.paos+xml  
1256 PAOS: ver="urn:liberty:paos:2003-08" ;  
1257 "urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
```

## 1258 E55: Language Cleanup Around Name Identifier Management

1259 Change [SAMLCore] Section 3.6.3 at lines 2477, 2483, and 2486-2487, and Section 8.3.7 at lines 3337-  
1260 3339, and change [SAMLProf] Section 4.5 at lines 1319 and 1323 to clear up ambiguities around name  
1261 identifier management and its application to various name identifier formats and differing identities for a  
1262 principal.

1263 Original at [SAMLCore] Section 3.6.3, lines 2477, 2483, and 2486-2487:

1264 If the <Terminate> element is included in the request, the requesting provider is indicating that (in the case  
1265 of a service provider) it will no longer accept assertions from the identity provider or (in the case of an  
1266 identity provider) it will no longer issue assertions to the service provider **about the principal**. The receiving  
1267 provider can perform any maintenance with the knowledge that the relationship represented by the name  
1268 identifier has been terminated.

1269 If the service provider requests that its identifier for the principal be changed by including a <NewID> (or  
1270 <NewEncryptedID>) element, the identity provider **MUST** include the element's content as the  
1271 SPProvidedID when subsequently communicating to the service provider **regarding this principal**.

1272 If the identity provider requests that its identifier for the principal be changed by including a <NewID> (or  
1273 <NewEncryptedID>) element, the service provider **MUST** use the element's content as the  
1274 <saml:NameID> element content when subsequently communicating with the identity provider **regarding  
1275 this principal**.

1276 New at [SAMLCore] Section 3.6.3, lines 2477, 2483, and 2486-2487 (note that E8 specifies additional  
1277 changes to the original text shown here):

1278 If the <Terminate> element is included in the request, the requesting provider is indicating that (in the case  
1279 of a service provider) it will no longer accept assertions from the identity provider or (in the case of an  
1280 identity provider) it will no longer issue assertions to the service provider **using that identifier**. The receiving  
1281 provider can perform any maintenance with the knowledge that the relationship represented by the name  
1282 identifier has been terminated.

1283 If the service provider requests that its identifier for the principal be changed by including a <NewID> (or  
1284 <NewEncryptedID>) element, the identity provider **MUST** include the element's content as the  
1285 SPProvidedID when subsequently communicating to the service provider **using the primary identifier**.

1286 If the identity provider requests that its identifier for the principal be changed by including a <NewID> (or  
1287 <NewEncryptedID>) element, the service provider **MUST** use the element's content as the  
1288 <saml:NameID> element content when subsequently communicating with the identity provider **in any case  
1289 where the identifier being changed would have been used**.

1290 New at [SAMLCore] Section 8.4.7, lines 3337-3339:

1291 The element's `SPNameQualifier` attribute, if present, MUST contain the unique identifier of the service  
1292 provider or affiliation of providers for whom the identifier was generated (see Section 8.3.6). It MAY be  
1293 omitted if the element is contained in a message intended only for consumption directly by the service  
1294 provider, and the value would be the unique identifier of that service provider.

1295 ~~The element's `sSPProvidedID` attribute MUST contain the alternative identifier of the principal most~~  
1296 ~~recently set by the service provider or affiliation, if any (see Section 3.6). If no such identifier has~~  
1297 ~~been established, then the attribute MUST be omitted.~~

1298 Original at [SAMLProf] Section 4.5, lines 1319 and 1323:

1299 In the scenario supported by the Name Identifier Management profile, an identity provider has exchanged  
1300 some form of **persistent** identifier for a principal with a service provider, allowing them to share a common  
1301 identifier for some length of time. Subsequently, the identity provider may wish to notify the service provider  
1302 of a change in the format and/or value that it will use to identify the same principal in the future. Alternatively  
1303 the service provider may wish to attach its own "alias" for the principal in order to ensure that the identity  
1304 provider will include it when communicating with it in the future **about the principal**. Finally, one of the  
1305 providers may wish to inform the other that it will no longer issue or accept messages using a particular  
1306 identifier. To implement these scenarios, a profile of the SAML Name Identifier Management protocol is  
1307 used.

1308 New at [SAMLProf] Section 4.5, lines 1319 and 1323 (note that E12 specifies additional changes to the  
1309 original text shown here):

1310 In the scenario supported by the Name Identifier Management profile, an identity provider has exchanged  
1311 some form of **long-term** identifier (**including but not limited to identifiers with a Format of**  
1312 **`urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`**) for a principal with a service  
1313 provider, allowing them to share a common identifier for some length of time. Subsequently, the identity  
1314 provider may wish to notify the service provider of a change in the format and/or value that it will use to  
1315 identify the same principal in the future. Alternatively the service provider may wish to attach its own "alias"  
1316 for the principal in order to ensure that the identity provider will include it when communicating with it in the  
1317 future **using that identifier**. Finally, one of the providers may wish to inform the other that it will no longer  
1318 issue or accept messages using a particular identifier. To implement these scenarios, a profile of the SAML  
1319 Name Identifier Management protocol is used.

## 1320 **E56: Confirmation Method Typo**

1321 Change [SAMLProf] Section 3 at line 326 to change the reference from `<ConfirmationMethod>` (an  
1322 element that no longer exists) to `Method` (an attribute, used instead of the element beginning in V2.0 of  
1323 SAML).

## 1324 **E57: SAMLmime Reference**

1325 Change [SAMLBind] Section 4 at lines 1468-1469 to replace a reference to an expired IETF I-D for the  
1326 SAMLmime definition to a persistent reference for the same definition.

1327 Original:

1328 [SAMLmime] **application/saml+xml Media Type Registration, IETF Internet-Draft,**  
1329 **<http://www.ietf.org/internet-drafts/draft-hodges-saml-mediatype-01.txt>.**

1330 New:

1331 [SAMLmime] **OASIS Security Services Technical Committee (SSTC),**  
1332 **"application/samlassertion+xml MIME Media Type Registration", IANA**  
1333 **MIME Media Types Registry application/samlassertion+xml, December**  
1334 **2004. See [http://www.iana.org/assignments/media-](http://www.iana.org/assignments/media-types/application/samlassertion+xml)**  
1335 **types/application/samlassertion+xml.**

1336

## E58: KeyDescriptor Typos in Profiles

1337 Change [SAMLProf] Section 4.1.6 at lines 626 and 627 to expand the keyword **sign** to **signing** and to  
1338 expand the keyword **encrypt** to **encryption**. These were typographical errors.

1339 Original:

1340 The providers MAY document the key(s) used to sign requests, responses, and assertions with  
1341 `<md:KeyDescriptor>` elements with a `use` attribute of **sign**. When encrypting SAML elements,  
1342 `<md:KeyDescriptor>` elements with a `use` attribute of **encrypt** MAY be used to document supported  
1343 encryption algorithms and settings, and public keys used to receive bulk encryption keys.

1344 New:

1345 The providers MAY document the key(s) used to sign requests, responses, and assertions with  
1346 `<md:KeyDescriptor>` elements with a `use` attribute of **signing**. When encrypting SAML elements,  
1347 `<md:KeyDescriptor>` elements with a `use` attribute of **encryption** MAY be used to document  
1348 supported encryption algorithms and settings, and public keys used to receive bulk encryption keys.

## 1349 E59: SSO Response When Using HTTP-Artifact

1350 Change [SAMLBind] Section 3.6.5.2 at line 1173 to observe for clarity's sake that particular message  
1351 delivery mechanisms are not mandated for the "nested" message exchange that takes place as part of  
1352 the HTTP-Artifact binding.

1353 New:

1354 Note also that there is no mechanism defined to protect the integrity of the relationship between the artifact  
1355 and the "RelayState" value, if any. That is, an attacker can potentially recombine a pair of valid HTTP  
1356 responses by switching the "RelayState" values associated with each artifact. As a result, the  
1357 producer/consumer of "RelayState" information MUST take care not to associate sensitive state information  
1358 with the "RelayState" value without taking additional precautions (such as based on the information in the  
1359 SAML protocol message retrieved via artifact).

1360 **Finally, note that the use of the `Destination` attribute in the root SAML element of the protocol  
1361 message is unspecified by this binding, because of the message indirection involved.**

## 1362 E60: Incorrect URI for Unspecified NameID Format

1363 Change [SAMLCore] Section 2.2.2 at line 460 to change the name identifier format from  
1364 `urn:oasis:names:tc:SAML:1.0:nameid-format:unspecified` to  
1365 `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`. This was a typographical error.

## 1366 E61: Reference to Non-Existent Element

1367 Change [SAMLCore] Section 7.1.2 at lines 3160.

1368 Original:

1369 The following SAML protocol **elements** are intended specifically for use as extension points in an extension  
1370 schema; **their types** are set to abstract, and are thus usable only as the base of a derived type:

- 1371 • **<Request>** and RequestAbstractType
- 1372 • **<SubjectQuery>** and SubjectQueryAbstractType

1373 New:

1374 The following SAML protocol **constructs** are intended specifically for use as extension points in an  
1375 extension schema; **the types listed** are set to abstract, and are thus usable only as the base of a derived  
1376 type:

- 1377 • RequestAbstractType

1378 • <SubjectQuery> and SubjectQueryAbstractType

## 1379 **E62: TLS Keys in KeyDescriptor**

1380 Change [SAMLMeta] Section 2.4.1.1 at line 624 to specify more clearly how to interpret the  
1381 KeyDescriptor element's use attribute.

1382 New (just after the conclusion of the definition list for KeyDescriptorType):

1383 **A use value of "signing" means that the contained key information is applicable to both signing**  
1384 **and TLS/SSL operations performed by the entity when acting in the enclosing role.**

1385 **A use value of "encryption" means that the contained key information is suitable for use in**  
1386 **wrapping encryption keys for use by the entity when acting in the enclosing role.**

1387 **If the use attribute is omitted, then the contained key information is applicable to both of the above**  
1388 **uses.**

1389 The following schema fragment defines the <KeyDescriptor> element and its KeyDescriptorType  
1390 complex type:

## 1391 **E63: IdP Discovery Cookie Interpretation**

1392 Change [SAMLProf] Section 4.3.1 at line 1105 to clear up confusion over interpretation of the contents of  
1393 an IdP Discovery cookie. (Note that E32 specifies changes to Section 4 that result in a new Section 4.3.1  
1394 being inserted before the original one; E63 applies to the original Section 4.3.1.)

1395 New:

1396 Cookie syntax should be in accordance with IETF RFC 2965 [RFC2965] or [NSCookie]. The cookie MAY be  
1397 either session-only or persistent. This choice may be made within a deployment, but should apply uniformly  
1398 to all identity providers in the deployment. **Note that while a session-only cookie can be used, the intent**  
1399 **of this profile is not to provide a means of determining whether a user actually has an active session**  
1400 **with one or more of the identity providers stored in the cookie. The cookie merely identifies identity**  
1401 **providers known to have been used in the past. Service providers MAY instead rely on the**  
1402 **IsPassive attribute in their <samlp:AuthnRequest> message to probe for active sessions.**

## 1403 **E64: Liberty Moniker Used Inappropriately**

1404 Change [SAMLSec] Section 7.1.1.9, Impersonation without Reauthentication to replace an accidental use  
1405 of the moniker "Liberty" in place of "SAML V2.0".

1406 New:

1407 Cookies posted by identity providers MAY be used to support this validation process, though **LibertySAML**  
1408 **V2.0** does not mandate a cookie-based approach.

## 1409 **E65: Second-level StatusCode**

1410 Change various sections as follows in [SAMLCore] to constrain the optional second-level <StatusCode>  
1411 element used, and clarify that use of second-level codes is optional.

1412 Change section 3.3.2.2.1, lines 1817-1819.

1413 New:

1414 **If none of the specified classes or declarations can be satisfied in accordance with the rules below, then the**  
1415 **responder MUST return a <Response> message with a top-level <StatusCode> value of**  
1416 **urn:oasis:names:tc:SAML:2.0:status:Responder and MAY return a second-level**  
1417 **<StatusCode> of urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext.**

1418 Change section 3.4.1.2, lines 2172-2173.

1419 New:

1420 In profiles specifying an active intermediary, the intermediary MAY examine the list and return a  
1421 <Response> message with an error <Status> and **optionally** a second-level <StatusCode> of

1422 Change section 3.4.1.5.1, lines 2282-2285.

1423 Original:

1424 An identity provider MUST NOT proxy a request where <ProxyCount> is set to zero. The identity  
1425 provider MUST return an error <Status> containing a second-level <StatusCode> value of  
1426 urn:oasis:names:tc:SAML:2.0:status:ProxyCountExceeded, unless it can directly  
1427 authenticate the presenter.

1428 New:

1429 **Unless the identity provider can directly authenticate the presenter, it MUST return a <Response>**  
1430 **message with a top-level <StatusCode> value of**  
1431 **urn:oasis:names:tc:SAML:2.0:status:Responder and MAY return a second-level**  
1432 **<StatusCode> value of urn:oasis:names:tc:SAML:2.0:status:ProxyCountExceeded.**

1433 Change section 3.8.3, lines 2729-2731.

1434 New:

1435 If the responder does not recognize the principal identified in the request, it MAY respond with an error  
1436 <Status>, **optionally** containing a second-level <StatusCode> of  
1437 urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal.

## 1438 **E66: Metadata and DNSSEC**

1439 Change [SAMLMeta] to update the DNSSEC reference from RFC 2535 to RFC 4035.

1440 Updated line 1253:

1441 It is RECOMMENDED that entities publish their resource records in signed zone files using ~~[RFC2535]~~  
1442 **[RFC4035]**

1443 Original at lines 1447-1448:

1444 [RFC2535] D. Eastlake. *Domain Name System Security Extensions*. IETF RFC 2535, March 1999. See  
1445 <http://www.ietf.org/rfc/rfc2535.txt>.

1446 New at lines 1447-1448:

1447 **[RFC4035] R. Arends et al. *Protocol Modifications for the DNS Security Extensions*. IETF RFC 4035,**  
1448 **March 2005. See <http://www.ietf.org/rfc/rfc4035.txt>.**

## 1449 **E68: Use of Multiple <KeyDescriptor> Elements**

1450 Add text to section 2.4.1.1 of [SAMLMeta] to clarify the meaning of identically-purposed  
1451 <KeyDescriptor> elements within a role.

1452 New at line 625:

1453 **The inclusion of multiple <KeyDescriptor> elements with the same use attribute (or no such**  
1454 **attribute) indicates that any of the included keys may be used by the containing role or affiliation. A**  
1455 **relying party SHOULD allow for the use of any of the included keys. When possible the signing or**  
1456 **encrypting party SHOULD indicate as specifically as possible which key it used to enable more**  
1457 **efficient processing.**

1458 The following schema fragment defines the <KeyDescriptor> element and its KeyDescriptorType  
1459 complex type:

## 1460 **E69: Semantics of <ds:KeyInfo> in <KeyDescriptor>**

1461 Add text to section 2.4.1.1 of [SAMLMeta] to clarify the limitations of the specification regarding the  
1462 semantics of various kinds of common key representations.

1463 New at line 625 (this change should appear after E68 above):

1464 **The <ds:KeyInfo> element is a highly generic and extensible means of communicating key**  
1465 **material. This specification takes no position on the allowable or suggested content of this element,**  
1466 **nor on its meaning to a relying party. As a concrete example, no implications of including an X.509**  
1467 **certificate by value or reference are to be assumed. Its validity period, extensions, revocation status,**  
1468 **and other relevant content may or may not be enforced, at the discretion of the relying party. The**  
1469 **details of such processing, and their security implications, are out of scope; they may, however, be**  
1470 **addressed by other SAML profiles.**

1471 The following schema fragment defines the <KeyDescriptor> element and its KeyDescriptorType  
1472 complex type:

## 1473 **E70: Obsolete reference to UUID URN namespace**

1474 Change [SAMLProf] to update the Internet Draft reference for the UUID URN namespace to RFC 4122.

1475 Updated Section 8.3.3.1, line 1836:

1476 values are equal in the sense of [~~http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-05.txt~~][RFC4122].  
1477 The

1478 Updated Section 8.4.3.1, line 1885:

1479 values are equal in the sense of [~~http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-05.txt~~][RFC4122].  
1480 The

1481 Original at lines 2111-2112:

1482 [Mealling] P Leach et al. *A UUID URN Namespace*. IETF Internet-Draft, December 2004. See  
1483 <http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-05.txt>.

1484 New at lines 2111-2112:

1485 [RFC4122] P. Leach et al. *A Universally Unique Identifier (UUID) URN Namespace*. IETF RFC 4122,  
1486 July 2005. See <http://www.ietf.org/rfc/rfc4122.txt>.

## 1487 **E71: Missing namespace definition in Profiles**

1488 Change [SAMLProf] to add the "xs" namespace prefix to the table in Section 1.

1489 New row of table in Section 1, between lines 267-268:

1490 **xs :**

1491 **<http://www.w3.org/2001/XMLSchema>**

1492 **This namespace is defined in the W3C XML Schema specification [Schema1]. In schema listings, this**  
1493 **is the default namespace and no prefix is shown. For clarity, the prefix is generally shown in**  
1494 **specification text when XML Schema-related constructs are mentioned.**

## 1495 **E74: Update XML Signature Reference**

1496 Update the XML Signature specification reference in [SAMLCore], [SAMLBind], [SAMLProf], [SAMLMeta],  
1497 [SAMLAuthCtx], [SAMLConf], [SAMLSec] to the "Second Edition". Also remove a stale non-normative  
1498 reference in [SAMLCore].

1499 Strike [SAMLCore], lines 3439-3440:

1500 [RFC 3075] D. Eastlake, J. Reagle, D. Solo. *XML Signature Syntax and Processing*. IETF RFC 3075,  
1501 March 2001. See <http://www.ietf.org/rfc/rfc3075.txt>.

1502 Original at [SAMLCore] lines 3415-3416, [SAMLBind] lines 1489-1491, [SAMLProf] lines 2205-2206,  
1503 [SAMLMeta] lines 1490-1491, [SAMLAuthCtx] lines 3926-3928, [SAMLConf] lines 410-412, [SAMLSec] lines  
1504 1078-1079:

1505 [XMLSig] D. Eastlake et al. *XML Signature Syntax and Processing*. World Wide Web Consortium,  
1506 February 2002. See <http://www.w3.org/TR/xmldsig-core/>. Note that this specification normatively  
1507 references [XMLSig-XSD], listed below.

1508 New at [SAMLCore] lines 3415-3416, [SAMLBind] lines 1489-1491, [SAMLProf] lines 2205-2206,  
1509 [SAMLMeta] lines 1490-1491, [SAMLAuthCtx] lines 3926-3928, [SAMLConf] lines 410-412, [SAMLSec]  
1510 lines 1078-1079:

1511 [XMLSig] D. Eastlake et al. *XML Signature Syntax and Processing, Second Edition*. World  
1512 Wide Web Consortium, June 2008. See <http://www.w3.org/TR/xmldsig-core/>.

## 1513 **E75: Clarify Handling of SubjectConfirmation in AuthnRequest**

1514 Change [SAMLCore] Section 3.4.1.4 to clarify an identity provider's obligation to return an error if can't  
1515 honor the requirements of a <SubjectConfirmation> element in an <AuthnRequest> message.

1516 New at line 2247:

1517 In such a case, the identifier's physical content MAY be different, but it MUST refer to the same principal. **If**  
1518 **the identity provider cannot or will not produce assertions with a strongly matching subject, then it**  
1519 **MUST return a <Response> with an error <Status>, and MAY return a second-level <StatusCode>**  
1520 **that reflects the reason for the failure.**

## 1521 **E76: Clarify nested validUntil/cacheDuration**

1522 Add text to [SAMLMeta] to clarify the processing of nested `validUntil` or `cacheDuration` attributes.

1523 New in Sections 2.3.1 and 2.3.2, before lines 336 and 409:

1524 When not used as the root element of a metadata instance, a `validUntil` or `cacheDuration` attribute  
1525 MAY be used to impose a shorter expiration or cache duration than that of the parent or root element, but  
1526 never a longer one; the smaller value takes precedence.

1527 New in Sections 2.4.1 and 2.5, before lines 589 and 972:

1528 A `validUntil` or `cacheDuration` attribute MAY be used to impose a shorter expiration or cache duration  
1529 than that of the parent or root element, but never a longer one; the smaller value takes precedence.

## 1530 **E77: Generalize scope of Metadata specification**

1531 Change [SAMLMeta] to address inadvertent language appearing to restrict use of SAML metadata to only  
1532 SAML profiles.

1533 New in Section 1, before line 137:

1534 A variety of extension points are also included to allow for the use of SAML metadata in non-SAML  
1535 specifications, profiles, and deployments, and such use is encouraged.

1536 Updated Section 2, lines 153-154:

1537 SAML metadata is organized around an extensible collection of roles representing common combinations of  
1538 SAML (and potentially non-SAML) protocols and profiles supported by system entities.

1539 Remove the word "SAML" from lines 226, 230, 311, 323, 332, 360, 372, 397, 403, 444, 478, 531, and  
1540 940.

1541

## E78: Reassignment of persistent identifiers

1542 Add text to [SAMLCore] Section 8.3.7, at line 3325, to clarify that non-reassignment to different principals  
1543 is a required property of "persistent" name identifiers.

1544 New:

1545 **Persistent name identifier values MUST NOT exceed a length of 256 characters. A given value, once**  
1546 **associated with a principal, MUST NOT be assigned to a different principal at any time in the future.**

1547

## E79: Clarification of SessionNotOnOrAfter

1548 Change [SAMLCore] Section 2.7.2, lines 1062-1065 to loosen wording around the  
1549 `SessionNotOnOrAfter` attribute and defer more explicitly to profiles.

1550 Original:

1551 Specifies a time instant at which the session between the principal identified by the subject and the SAML  
1552 authority issuing this statement MUST be considered ended. The time value is encoded in UTC, as  
1553 described in Section 1.3.3. There is no required relationship between this attribute and a `NotOnOrAfter`  
1554 condition attribute that may be present in the assertion.

1555 New:

1556 **Indicates an upper bound on sessions with the subject derived from the enclosing assertion. The**  
1557 **time value is encoded in UTC, as described in Section 1.3.3. There is no required relationship between this**  
1558 **attribute and a `NotOnOrAfter` condition attribute that may be present in the assertion. It's left to profiles**  
1559 **to provide specific processing rules for relying parties based on this attribute.**

1560

## E81: Algorithm statement in XML Signature profile

1561 Change [SAMLCore] Section 5.4.1, lines 2926-2927, and [SAMLMeta] Section 3.1.1, lines 1182-1183, to  
1562 relax the implication that RSA with SHA1 is the only supported algorithm.

1563 Original:

1564 SAML processors SHOULD support the use of RSA signing and verification for public key operations in  
1565 accordance with the algorithm identified by <http://www.w3.org/2000/09/xmldsig#rsa-sha1>.

1566 New:

1567 **Any algorithm defined for use with the XML Signature specification MAY be used.**

1568

## E82: Empty <ContactPerson> element

1569 Add text to [SAMLMeta] Section 2.3.2.2, before line 500, to clarify that child elements should be included.

1570 New:

1571 **At least one child element SHOULD be present in a <ContactPerson> element.**

1572

## E83: Weaken claim made about Exclusive C14N

1573 Change [SAMLCore] Section 5.4.3, lines 2939-2940, and [SAMLMeta] Section 3.1.3, lines 1196-1197, to  
1574 better explain the purpose of using exclusive canonicalization.

1575 Original:

1576 Use of Exclusive Canonicalization ensures that signatures created over SAML messages embedded in an  
1577 XML context can be verified independent of that context.

1578 New:

1579 Use of Exclusive Canonicalization facilitates the verification of signatures created over SAML messages  
1580 when placed into a different XML context than present during signing.

1581 Note that use of this algorithm alone does not guarantee that a particular signed object can be moved from  
1582 one context to another safely, nor is that a requirement of signed SAML objects in general, though it MAY be  
1583 required by particular profiles.

1584

---

## 3 Acknowledgments

1585

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

1586

1587

- Rob Philpott, EMC Corporation

1588

- John Bradley, Individual

1589

- Jeff Hodges, Individual

1590

- Scott Cantor, Internet2

1591

- Nate Klingenstein, Internet2

1592

- Bob Morgan, Internet2

1593

- Joni Brennan, Liberty Alliance Project

1594

- Thomas Hardjono, M.I.T.

1595

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

1596

- Peter Davis, NeuStar, Inc.

1597

- Frederick Hirsch, Nokia Corporation

1598

- Paul Madsen, NTT Corporation

1599

- Ari Kermaier, Oracle Corporation

1600

- Hal Lockhart, Oracle Corporation

1601

- Anil Saldhana, Red Hat

1602

- Kent Spaulding, Skyworth TTG Holdings Limited

1603

- Eve Maler, Sun Microsystems

1604

- Emily Xu, Sun Microsystems

1605

- David Staggs, Veterans Health Administration

1606

The editors also would like to gratefully acknowledge **Jahan Moreh** of Sigaba, who during his tenure on

1607

the TC was the primary editor of the errata working document and who made major substantive

1608

contributions to all of the errata materials.