



# SAML V2.0 Approved Errata

1 December 2009

## Specification URIs:

### This Version:

<http://docs.oasis-open.org/security/saml/v2.0/ssstc-saml-approved-errata-2.0-02.html>  
<http://docs.oasis-open.org/security/saml/v2.0/ssstc-saml-approved-errata-2.0-02.odt>  
<http://docs.oasis-open.org/security/saml/v2.0/ssstc-saml-approved-errata-2.0-02.pdf> (Authoritative)

### Previous Version:

<http://docs.oasis-open.org/security/saml/v2.0/ssstc-saml-errata-2.0-cd-04.html>  
<http://docs.oasis-open.org/security/saml/v2.0/ssstc-saml-errata-2.0-cd-04.odt>  
<http://docs.oasis-open.org/security/saml/v2.0/ssstc-saml-errata-2.0-cd-04.pdf> (Authoritative)

### Latest Version:

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

## 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.

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

---

# Notices

46

47 Copyright © OASIS® 1993–2009. All Rights Reserved.

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

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

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

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

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

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

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

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

## Table of Contents

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

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

---

# 1 Introduction

170

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

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

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

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

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

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

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

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

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

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

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

## 1.1 Normative References

200

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

203 **[SAMLAuthCtx]** J. Kemp et al. *Authentication Context for the OASIS Security Assertion Markup*  
204 *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)  
205 [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).

206 **[SAMLBind]** S. Cantor et al. *Bindings for the OASIS Security Assertion Markup Language*  
207 *(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)  
208 [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).

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

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

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

---

## 2 Approved Errata

227

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

229

### E0: Incorrect Section Reference

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

232

### E1: Relay State for HTTP Redirect

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

236 Original:

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

242 New:

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

246

### E2: Metadata Clarifications for HTTP Artifact Binding

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

249 Original:

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

254 New:

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

259

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

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

262 New:

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

267  
268

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

269

## E6: Clarify Constraints on Encrypted NameID

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

272 New:

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

280

## E7: Metadata for Agreeing to Sign Authentication Requests

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

284 New at line 710:

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

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

295  
296  
297 The following schema fragment defines the `<IDPSSODescriptor>` element and its  
298 `IDPSSODescriptorType` complex type:

299 New at lines 741-742:

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

306 New at lines 744-747:

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

## E8: SLO and NameID Termination

312

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

315 Original:

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

319 New:

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

## E10: Logout Request Reason Mismatch with Schema

329

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

334 New:

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

## E11: Improperly Labeled Feature

338

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

340 Original labels:

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

345 New labels:

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

## E12: Clarification on ManageNameIDRequest

350

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

354 New [SAMLCore] at lines 2412-2413:

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

359 New [SAMLCore] at line 2438:

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

364 New [SAMLProf] at lines 1320-23121:

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

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

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

370 New:

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

## 373 **E14: AllowCreate**

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

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

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

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

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

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

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

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

400  
401 **A value of "true" permits the identity provider to take any related actions it wishes to fulfill the**

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

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

421 A service provider also uses this message to register or change the `SPProvidedID` value to be included  
422 when the underlying name identifier is used to communicate with it, or to terminate the use of a name  
423 identifier between itself and the identity provider.  
424

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

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

428 A service provider also uses this message to register or change the `SPProvidedID` value to be included  
429 when the underlying name identifier is used to communicate with it, or to terminate the use of a name  
430 identifier between itself and the identity provider.  
431

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

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

436 If the `<Terminate>` element is included in the request, the requesting provider is indicating that (in the case  
437 of a service provider) it will no longer accept assertions from the identity provider or (in the case of an  
438 identity provider) it will no longer issue assertions to the service provider about the principal. The receiving  
439 provider can perform any maintenance with the knowledge that the relationship represented by the name  
440 identifier has been terminated. It can choose to invalidate the active session(s) of a principal for whom a  
441 relationship has been terminated.  
442

443 **If the receiving provider is maintaining state associated with the name identifier, such as the value of**  
444 **the identifier itself (in the case of a pair-wise identifier), an `SPProvidedID` value, the sender's**  
445 **consent to the identifier's creation/use, etc., then the receiver can perform any maintenance with the**  
446 **knowledge that the relationship represented by the name identifier has been terminated.**  
447

448 **Any subsequent operations performed by the receiver on behalf of the sender regarding the**  
449 **principal (for example, a subsequent `<AuthnRequest>`) SHOULD be carried out in a manner**  
450 **consistent with the absence of any previous state.**  
451

452 **Termination is potentially the cleanup step for any state management behavior triggered by the use**  
453 **of the `AllowCreate` attribute in the Authentication Request protocol (see Section 3.4). Deployments**  
454 **that do not make use of that attribute are likely to avoid the use of the `<Terminate>` element or**  
455 **would treat it as a purely advisory matter.**  
456

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

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

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

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

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

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

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

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

## 477 **E15: NameID Policy Adherence**

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

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

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

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

490 ● the Format value of the <NameID> within the <Subject> of any <Assertion> MUST be identical  
491 to the Format value supplied in the <NameIDPolicy>, and  
492

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

## 497 **E17: Authentication Response IssuerName vs. Assertion 498 IssuerName**

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

501 Original:

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

505 New:

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

## 510 **E18: Reference to Identity Provider Discovery Service in ECP** 511 **Profile**

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

514 New:

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

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

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

521 Original at Section 3.2.2.1, lines 310-317:

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

529 New at Section 3.2.2.1, lines 310-317:

530 The SAML responder **SHOULD** return a **SOAP message containing either a SAML response element in**  
531 **the body or a SOAP fault**. The SAML responder **MUST NOT** include more than one SAML response per  
532 SOAP message or include any additional XML elements in the SOAP body. SOAP fault codes **SHOULD**  
533 **NOT** be sent for errors within the SAML problem domain, for example, inability to find an extension schema  
534 or as a signal that the subject is not authorized to access a resource in an authorization query. **See Section**  
535 **3.2.3.3 for more information about error handling**. (SOAP 1.1 faults and fault codes are discussed in  
536 [SOAP11] Section 4.1.)

537 Original at Section 3.2.3.3, line 378:

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

540 New at Section 3.2.3.3, line 378:

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

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

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

546 New (small portion of previous subsection shown):

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

550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560

## 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 a service provider may send requests to an identity provider using this profile.

561

## E21: PAOS Version

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

563  
564

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

565

## E22: Error in Profile/ECP

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

568

## E24: HTTPS in URI Binding

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

571 Original:

572  
573  
574

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)**.

575 New:

576  
577

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**.

578

## E25: Metadata Feature in Conformance

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

581 New in Table 2:

582  
583  
584

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>

585 New in Table 4:

586  
587  
588

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>

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

590  
591

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

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

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

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

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

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

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

705 Original at Section 4.1.4.3, lines 576-591:

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

723 New at Section 4.1.4.3, lines 576-591:

- 724 • Verify that the Recipient attribute in **the** bearer <SubjectConfirmationData> matches the assertion  
 725 consumer service URL to which the <Response> or artifact was delivered
- 726
- 727 • Verify that the NotOnOrAfter attribute in **the** bearer <SubjectConfirmationData> has not passed,  
 728 subject to allowable clock skew between the providers
- 729
- 730 • Verify that the InResponseTo attribute in the bearer <SubjectConfirmationData> equals the ID of  
 731 its original <AuthnRequest> message, unless the response is unsolicited (see Section 4.1.5 ), in which  
 732 case the attribute MUST NOT be present
- 733 • Verify that any assertions relied upon are valid in other respects. **Note that while multiple bearer**  
 734 **<SubjectConfirmation> elements may be present, the successful evaluation of a single such**  
 735 **element in accordance with this profile is sufficient to confirm an assertion. However, each**  
 736 **assertion, if more than one is present, MUST be evaluated independently.**
- 737 • If **any the** bearer <SubjectConfirmationData> includes an Address attribute, the service provider  
 738 MAY check the user agent's client address against it.
- 739 • Any assertion which is not valid, or whose subject confirmation requirements cannot be met SHOULD be  
 740 discarded and SHOULD NOT be used to establish a security context for the principal.
- 741 • If an <AuthnStatement> used to establish a security context for the principal contains a  
 742 SessionNotOnOrAfter attribute, the security context SHOULD be discarded once this time is reached,  
 743 unless the service provider reestablishes the principal's identity by repeating the use of this profile. **Note**

744 that if multiple <AuthnStatement> elements are present, the SessionNotOnOrAfter value closest  
745 to the present time SHOULD be honored.

746 Original at Section 4.1.4.5, lines 600-601:

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

748 New at Section 4.1.4.5, lines 600-601:

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

## 752 E27: Incorrect Step Number in ECP Profile

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

## 755 E28: Profile Labeling in Conformance

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

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

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

## 762 E29: Incomplete Listing of Features in Conformance

763 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

## 767 E30: Key Replacement

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

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

771 New:

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

## 774 E31: Various Minor Errors in Binding

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

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

778 Original at Section 3.5.3, line 785:

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

781 New at Section 3.5.3, line 785:

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

784 Original at Section 3.6.5, line 1136:

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

788 New at Section 3.6.5, line 1136:

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

792 Original at Section 3.6.5, line 1397:

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

794 New at Section 3.6.5, line 1397:

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

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

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

### 799 **4.3.1 Required Information**

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

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

802 **Description:** Given below.

803 **Updates:** None.

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

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

## 808 **E34: RequestedAttribute Section Heading**

809 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  
810 consistency in reflecting element nesting in the document outline.

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

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

814 Original at Section 4.2.4.1, lines 906-908:

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

819 New at lines Section 4.2.4.1, 906-908:

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

824 Original at Section 4.2.4.3, line 964:

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

827 New at Section 4.2.4.3, line 964:

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

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

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

835 Original:

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

840 New:

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

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

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

845 Original:

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

848 New:

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

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

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

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

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

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

```
859 <SessionIndex> [Optional]
```

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

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

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

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

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

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

875 Original:

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

885 New:

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

## 895 **E40: Holder of Key**

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

898 Original:

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

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

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

## E41: EndpointType ResponseLocation Clarification in Metadata

906

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

909 New:

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

## E42: Match Authorities to Queries in Conformance

917

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

921 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

926 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

931

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

935 Original:

936 **6.2 Combining Signatures and Encryption**

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

- 940 • **When a signed <Assertion> element is encrypted, the signature MUST first be calculated and**  
941 **placed within the <Assertion> element before the element is encrypted.**
- 942 • **When a <BaseID>, <NameID>, or <Attribute> element is encrypted, the encryption MUST be**  
943 **performed first and then the signature calculated over the assertion or message containing the**  
944 **encrypted element.**

945 New:

946 **6.2 Key and Data Referencing Guidelines**

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

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

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

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

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

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

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

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

### 975 6.3 Examples

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

```

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

```

1005 In the following <EncryptedAttribute> example, the <xenc:EncryptedKey> element is contained  
1006 within the <xenc:EncryptedData> element, so there is no explicit referencing:

```
1007 <saml:EncryptedAttribute  
1008   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">  
1009   <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"  
1010     Id="Encrypted_DATA_ID"  
1011     Type="http://www.w3.org/2001/04/xmlenc#Element">  
1012     <xenc:EncryptionMethod  
1013       Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>  
1014     <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">  
1015       <xenc:EncryptedKey Id="Encrypted_KEY_ID">  
1016         <xenc:EncryptionMethod  
1017           Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>  
1018         <xenc:CipherData>  
1019           <xenc:CipherValue>SDFSDF... </xenc:CipherValue>  
1020         </xenc:CipherData>  
1021       </xenc:EncryptedKey>  
1022     </ds:KeyInfo>  
1023     <xenc:CipherData>  
1024       <xenc:CipherValue>Nk4W4mx...</xenc:CipherValue>  
1025     </xenc:CipherData>  
1026   </xenc:EncryptedData>  
1027 </saml:EncryptedAttribute>
```

1028 The final example shows an assertion encrypted for multiple recipients, using the  
1029 <xenc:CarriedKeyName> approach:

```
1030 <saml:EncryptedAssertion  
1031   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">  
1032   <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"  
1033     Id="Encrypted_DATA_ID"  
1034     Type="http://www.w3.org/2001/04/xmlenc#Element">  
1035     <xenc:EncryptionMethod  
1036       Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>  
1037     <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">  
1038       <ds:KeyName>MULTICAST_KEY_NAME</ds:KeyName>  
1039     </ds:KeyInfo>  
1040     <xenc:CipherData>  
1041       <xenc:CipherValue>Nk4W4mx...</xenc:CipherValue>  
1042     </xenc:CipherData>  
1043   </xenc:EncryptedData>  
1044  
1045   <xenc:EncryptedKey xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"  
1046     Id="Encrypted_KEY_ID_1" Recipient="https://sp1.org">  
1047     <xenc:EncryptionMethod  
1048       Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>  
1049     <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">  
1050       <ds:KeyName>KEY_NAME_1</ds:KeyName>  
1051     </ds:KeyInfo>  
1052     <xenc:CipherData>  
1053       <xenc:CipherValue>xyzABC...</xenc:CipherValue>  
1054     </xenc:CipherData>  
1055     <xenc:ReferenceList>  
1056       <xenc:DataReference URI="#Encrypted_DATA_ID"/>  
1057     </xenc:ReferenceList>  
1058  
1059     <xenc:CarriedKeyName>MULTICAST_KEY_NAME</xenc:CarriedKeyName>  
1060   </xenc:EncryptedKey>  
1061  
1062   <xenc:EncryptedKey xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"  
1063     Id="Encrypted_KEY_ID_2" Recipient="https://sp2.org">  
1064     <xenc:EncryptionMethod  
1065       Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
```

```

1066 <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
1067   <ds:KeyName>KEY_NAME_2</ds:KeyName>
1068 </ds:KeyInfo>
1069 <xenc:CipherData>
1070   <xenc:CipherValue>abcXYZ...</xenc:CipherValue>
1071 </xenc:CipherData>
1072 <xenc:ReferenceList>
1073   <xenc:DataReference URI="#Encrypted_DATA_ID"/>
1074 </xenc:ReferenceList>
1075
1076   <xenc:CarriedKeyName>MULTICAST_KEY_NAME</xenc:CarriedKeyName>
1077 </xenc:EncryptedKey>
1078 </saml:EncryptedAssertion>

```

## E45: AuthnContext Comparison Order

1079  
1080 Change [SAMLCore] Section 3.3.2.2.1 at lines 1815-1819 and 1826 to clarify the lack of orderedness in  
1081 the comparison of a set of authentication contexts.

1082 Original at Section 3.3.2.2.1, lines 1815-1819:

1083 Either a set of class references or a set of declaration references can be used. The set of supplied  
1084 references MUST be evaluated as an ordered set, where the first element is the most preferred  
1085 authentication context class or declaration. If none of the specified classes or declarations can be satisfied in  
1086 accordance with the rules below, then the responder MUST return a <Response> message with a second-  
1087 level <StatusCode> of urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext.

1088 New at Section 3.3.2.2.1, lines 1815-1819:

1089 Either a set of class references or a set of declaration references can be used. **If ordering is relevant to**  
1090 **the evaluation of the request, then** the set of supplied references MUST be evaluated as an ordered set,  
1091 where the first element is the most preferred authentication context class or declaration. If none of the  
1092 specified classes or declarations can be satisfied in accordance with the rules below, then the responder  
1093 MUST return a <Response> message with a second-level <StatusCode> of  
1094 urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext. **For example, ordering is significant**  
1095 **when using this element in an <AuthnRequest> message but not in an <AuthnQuery> message.**

1096 Original 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 **any** of the authentication contexts specified.

1099 New at Section 3.3.2.2.1, line 1826:

1100 If Comparison is set to "better", then the resulting authentication context in the authentication statement  
1101 MUST be stronger (as deemed by the responder) than **one** of the authentication contexts specified.

## E46: AudienceRestriction Clarifications

1103 Change [SAMLCore] Section 2.5.1.4 at lines 924-925 to clarify the logical sense with respect to individual  
1104 audience elements within an audience-restriction condition grouping.

1105 Original:

1106 Note that multiple <AudienceRestriction> elements MAY be included in a single assertion, and each  
1107 MUST be evaluated independently. The effect of this requirement and the preceding definition is that within  
1108 a given **condition**, the **audiences** form a disjunction (an "OR") while multiple **conditions** form a conjunction  
1109 (an "AND").

1110 New:

1111 Note that multiple <AudienceRestriction> elements MAY be included in a single assertion, and each  
1112 MUST be evaluated independently. The effect of this requirement and the preceding definition is that within

1113 a given <AudienceRestrictions>, the <Audience> elements form a disjunction (an "OR") while  
1114 multiple <AudienceRestrictions> elements form a conjunction (an "AND").

## 1115 **E47: Clarification on SubjectConfirmation**

1116 Change [SAMLCore] Section 2.4.1.1 at line 698, and change [SAMLProf] Section 3.1 at lines 336 and 341  
1117 and Section 3.3 at lines 361-363, in order to clarify behavior around the subject confirmation element and  
1118 the intent of the embedded secondary identifier.

1119 New at [SAMLCore] Section 2.4.1.1, line 698 (add text just before the schema listing introduction):

1120 **If the <SubjectConfirmation> element in an assertion subject contains an identifier the issuer**  
1121 **authorizes the attesting entity to wield the assertion on behalf of that subject. A relying party MAY**  
1122 **apply additional constraints on the use of such an assertion at its discretion, based upon the**  
1123 **identities of both the subject and the attesting entity.**

1124 **If an assertion is issued for use by an entity other than the subject, then that entity SHOULD be**  
1125 **identified in the <SubjectConfirmation> element.**

1126 The following schema fragment defines the <SubjectConfirmation> element and its  
1127 SubjectConfirmationType complex type:

1128 Original at [SAMLProf] Section 3.1, line 336:

1129 As described in [XMLSig], each <ds:KeyInfo> element holds a key or information that enables an  
1130 application to obtain a key. The holder of a **specified key** is considered to be the subject of the assertion by  
1131 the asserting party.

1132 New at [SAMLProf] Section 3.1, line 336 (note that E40 specified additional changes to the original text  
1133 shown here):

1134 As described in [XMLSig], each <ds:KeyInfo> element holds a key or information that enables an  
1135 application to obtain a key. The holder of **one or more of the specified keys** is considered to be the subject  
1136 of the assertion by the asserting party.

1137 New at [SAMLProf] Section 3.1, line 341 (add text just before the example):

1138 **If the <SubjectConfirmation> element in an assertion subject contains an identifier the issuer**  
1139 **authorizes the attesting entity to wield the assertion on behalf of that subject. A relying party MAY**  
1140 **apply additional constraints on the use of such an assertion at its discretion, based upon the**  
1141 **identities of both the subject and the attesting entity.**

1142 **If an assertion is issued for use by an entity other than the subject, then that entity SHOULD be**  
1143 **identified in the <SubjectConfirmation> element.**

1144 Example: The holder of the key named "By-Tor" or the holder of the key named "Snow Dog" can confirm  
1145 itself as the subject.

1146 Original at [SAMLProf] Section 3.3, lines 361-363:

1147 The subject of the assertion is **the bearer of the assertion**, subject to optional constraints on confirmation  
1148 using the attributes that MAY be present in the <SubjectConfirmationData> element, as defined by  
1149 [SAMLCore].

1150 New at [SAMLProf] Section 3.3, lines 361-363:

1151 The subject of the assertion is **considered to be an acceptable attesting entity for the assertion by the**  
1152 **asserting party**, subject to optional constraints on confirmation using the attributes that MAY be present in  
1153 the <SubjectConfirmationData> element, as defined by [SAMLCore].

1154 **If the intended bearer is known by the asserting party to be an entity other than the subject, then the**  
1155 **asserting party SHOULD identify that entity to the relying party by including a SAML identifier**  
1156 **representing it in the enclosing <SubjectConfirmation> element.**

1157 **If multiple attesting entities are to be permitted to use the assertion based on bearer semantics, then**  
1158 **multiple <SubjectConfirmation> elements SHOULD be included.**

1159

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

1160

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

1161

1162 Change [SAMLProf] at line 1762. Original:

1163

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".

1164

1165

1166

1167 New:

1168

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".

1169

1170

1171

1172

1173

## E49: Clarification on Attribute Name Format

1174

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

1175

1176 New (add text to the end of the definition of `<AttributeValue>`):

1177

`<AttributeValue>` [Any Number]

1178

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.

1179

1180

1181

1182

1183

**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.**

1184

1185

1186

**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`.**

1187

1188

1189

1190

1191

## E50: Clarification on SSL Ciphersuites

1192

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.

1193

1194 New at Section 4, line 235:

1195

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.**

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206 New at Section 5, line 257:

1207 In any SAML V2.0 use of SSL 3.0 [SSL3] or TLS 1.0 [RFC 2246], servers MUST authenticate to clients  
1208 using a X.509 v3 certificate. The client MUST establish server identity based on contents of the certificate  
1209 (typically through examination of the certificate's subject DN field). **The set of algorithms required for**  
1210 **SAML V2.0 conformance is equivalent to that defined in SAML V1.0 and SAML V1.1. These mandated**  
1211 **algorithms were chosen by the SSTC because of their wide implementation support in the industry.**  
1212 **While the algorithms defined below are the minimal set for SAML conformance, additional**  
1213 **algorithms supported by SSL 3.0 and TLS 1.0 MAY be used.**

## 1214 **E51: Schema Type of Contents of <AttributeValue>**

1215 Change [SAMLProf] Section 8.1.4 at line 1670 to change the reference from **Section 3.3** to **Section 3**, in  
1216 order to fix a typographical error that would have improperly restricted the valid types for attribute values  
1217 to derived types, rather than the larger category of built-in types.

## 1218 **E52: Clarification on NotOnOrAfter Attribute for Subject** 1219 **Confirmation**

1220 Change [SAMLProf] Section 4.1.4.2 at line 557 to correctly reflect the type of validity period that applies to  
1221 subject confirmation.

1222 Original:

1223 The bearer <SubjectConfirmation> element described above MUST contain a  
1224 <SubjectConfirmationData> element that contains a Recipient attribute containing the service  
1225 provider's assertion consumer service URL and a NotOnOrAfter attribute that limits the window during  
1226 which the assertion can be **delivered**. It MAY contain an Address attribute limiting the client address from  
1227 which the assertion can be delivered.

1228 New (note that E26 specifies additional changes to the original text shown here):

1229 The bearer <SubjectConfirmation> element described above MUST contain a  
1230 <SubjectConfirmationData> element that contains a Recipient attribute containing the service  
1231 provider's assertion consumer service URL and a NotOnOrAfter attribute that limits the window during  
1232 which the assertion can be **confirmed by the relying party**. It MAY contain an Address attribute limiting  
1233 the client address from which the assertion can be delivered.

## 1234 **E53: Correction to LDAP/X.500 Profile Attribute**

1235 Deprecate [SAMLProf] Section 8.2 at lines 1677-1799 by adding a notice after line 1677.

1236 New:

1237 **8.2 X.500/LDAP Attribute Profile – Deprecated**  
1238 **NOTE: This attribute profile is deprecated because of a flaw that makes it schema-invalid. The SSTC**  
1239 **has replaced it with a separately published SAML V2.0 X.500/LDAP Attribute Profile specification**  
1240 **that removes this flaw.**  
1241 Directories based on the ITU-T X.500 specifications [X.500] and the related IETF Lightweight Directory  
1242 Access Protocol specifications [LDAP] are widely deployed....

## 1243 **E54: Corrections to ECP URN**

1244 Change [SAMLProf] Section 4.2.3.1 at lines 757 and 763-764 to correct the usage of quotation marks in  
1245 HTTP headers.

1246 New at line 757 (add double quotation marks around the URN):

1247 Furthermore, support for this profile **MUST** be specified in the HTTP PAOS Header field as a service value,  
1248 with the value "urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp".

1249 Original at lines 763-764 (single quotation marks are problematic):

```
1250 GET /index HTTP/1.1  
1251 Host: identity-service.example.com  
1252 Accept: text/html; application/vnd.paos+xml  
1253 PAOS: ver='urn:liberty:paos:2003-08' ;  
1254 'urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp'
```

1255 New at lines 763-764 (double quotation marks used instead):

```
1256 GET /index HTTP/1.1  
1257 Host: identity-service.example.com  
1258 Accept: text/html; application/vnd.paos+xml  
1259 PAOS: ver="urn:liberty:paos:2003-08" ;  
1260 "urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
```

## 1261 E55: Language Cleanup Around Name Identifier Management

1262 Change [SAMLCore] Section 3.6.3 at lines 2477, 2483, and 2486-2487, and Section 8.3.7 at lines 3337-  
1263 3339, and change [SAMLProf] Section 4.5 at lines 1319 and 1323 to clear up ambiguities around name  
1264 identifier management and its application to various name identifier formats and differing identities for a  
1265 principal.

1266 Original at [SAMLCore] Section 3.6.3, lines 2477, 2483, and 2486-2487:

1267 If the <Terminate> element is included in the request, the requesting provider is indicating that (in the case  
1268 of a service provider) it will no longer accept assertions from the identity provider or (in the case of an  
1269 identity provider) it will no longer issue assertions to the service provider **about the principal**. The receiving  
1270 provider can perform any maintenance with the knowledge that the relationship represented by the name  
1271 identifier has been terminated.

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

1275 If the identity provider requests that its identifier for the principal be changed by including a <NewID> (or  
1276 <NewEncryptedID>) element, the service provider **MUST** use the element's content as the  
1277 <saml:NameID> element content when subsequently communicating with the identity provider **regarding**  
1278 **this principal**.

1279 New at [SAMLCore] Section 3.6.3, lines 2477, 2483, and 2486-2487 (note that E8 specifies additional  
1280 changes to the original text shown here):

1281 If the <Terminate> element is included in the request, the requesting provider is indicating that (in the case  
1282 of a service provider) it will no longer accept assertions from the identity provider or (in the case of an  
1283 identity provider) it will no longer issue assertions to the service provider **using that identifier**. The receiving  
1284 provider can perform any maintenance with the knowledge that the relationship represented by the name  
1285 identifier has been terminated.

1286 If the service provider requests that its identifier for the principal be changed by including a <NewID> (or  
1287 <NewEncryptedID>) element, the identity provider **MUST** include the element's content as the  
1288 SPProvidedID when subsequently communicating to the service provider **using the primary identifier**.

1289 If the identity provider requests that its identifier for the principal be changed by including a <NewID> (or  
1290 <NewEncryptedID>) element, the service provider **MUST** use the element's content as the  
1291 <saml:NameID> element content when subsequently communicating with the identity provider **in any case**  
1292 **where the identifier being changed would have been used**.

1293 New at [SAMLCore] Section 8.4.7, lines 3337-3339:

1294 The element's `SPNameQualifier` attribute, if present, MUST contain the unique identifier of the service  
1295 provider or affiliation of providers for whom the identifier was generated (see Section 8.3.6). It MAY be  
1296 omitted if the element is contained in a message intended only for consumption directly by the service  
1297 provider, and the value would be the unique identifier of that service provider.

1298 ~~The element's `sSPProvidedID` attribute MUST contain the alternative identifier of the principal most~~  
1299 ~~recently set by the service provider or affiliation, if any (see Section 3.6). If no such identifier has~~  
1300 ~~been established, then the attribute MUST be omitted.~~

1301 Original at [SAMLProf] Section 4.5, lines 1319 and 1323:

1302 In the scenario supported by the Name Identifier Management profile, an identity provider has exchanged  
1303 some form of **persistent** identifier for a principal with a service provider, allowing them to share a common  
1304 identifier for some length of time. Subsequently, the identity provider may wish to notify the service provider  
1305 of a change in the format and/or value that it will use to identify the same principal in the future. Alternatively  
1306 the service provider may wish to attach its own "alias" for the principal in order to ensure that the identity  
1307 provider will include it when communicating with it in the future **about the principal**. Finally, one of the  
1308 providers may wish to inform the other that it will no longer issue or accept messages using a particular  
1309 identifier. To implement these scenarios, a profile of the SAML Name Identifier Management protocol is  
1310 used.

1311 New at [SAMLProf] Section 4.5, lines 1319 and 1323 (note that E12 specifies additional changes to the  
1312 original text shown here):

1313 In the scenario supported by the Name Identifier Management profile, an identity provider has exchanged  
1314 some form of **long-term** identifier (**including but not limited to identifiers with a Format of**  
1315 **`urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`**) for a principal with a service  
1316 provider, allowing them to share a common identifier for some length of time. Subsequently, the identity  
1317 provider may wish to notify the service provider of a change in the format and/or value that it will use to  
1318 identify the same principal in the future. Alternatively the service provider may wish to attach its own "alias"  
1319 for the principal in order to ensure that the identity provider will include it when communicating with it in the  
1320 future **using that identifier**. Finally, one of the providers may wish to inform the other that it will no longer  
1321 issue or accept messages using a particular identifier. To implement these scenarios, a profile of the SAML  
1322 Name Identifier Management protocol is used.

## 1323 **E56: Confirmation Method Typo**

1324 Change [SAMLProf] Section 3 at line 326 to change the reference from `<ConfirmationMethod>` (an  
1325 element that no longer exists) to `Method` (an attribute, used instead of the element beginning in V2.0 of  
1326 SAML).

## 1327 **E57: SAMLmime Reference**

1328 Change [SAMLBind] Section 4 at lines 1468-1469 to replace a reference to an expired IETF I-D for the  
1329 SAMLmime definition to a persistent reference for the same definition.

1330 Original:

1331 [SAMLmime] **application/saml+xml Media Type Registration, IETF Internet-Draft,**  
1332 **<http://www.ietf.org/internet-drafts/draft-hodges-saml-mediatype-01.txt>.**

1333 New:

1334 [SAMLmime] **OASIS Security Services Technical Committee (SSTC),**  
1335 **"application/samlassertion+xml MIME Media Type Registration", IANA**  
1336 **MIME Media Types Registry application/samlassertion+xml, December**  
1337 **2004. See [http://www.iana.org/assignments/media-](http://www.iana.org/assignments/media-types/application/samlassertion+xml)**  
1338 **types/application/samlassertion+xml.**

1339

## E58: KeyDescriptor Typos in Profiles

1340 Change [SAMLProf] Section 4.1.6 at lines 626 and 627 to expand the keyword **sign** to **signing** and to  
1341 expand the keyword **encrypt** to **encryption**. These were typographical errors.

1342 Original:

1343 The providers MAY document the key(s) used to sign requests, responses, and assertions with  
1344 `<md:KeyDescriptor>` elements with a `use` attribute of **sign**. When encrypting SAML elements,  
1345 `<md:KeyDescriptor>` elements with a `use` attribute of **encrypt** MAY be used to document supported  
1346 encryption algorithms and settings, and public keys used to receive bulk encryption keys.

1347 New:

1348 The providers MAY document the key(s) used to sign requests, responses, and assertions with  
1349 `<md:KeyDescriptor>` elements with a `use` attribute of **signing**. When encrypting SAML elements,  
1350 `<md:KeyDescriptor>` elements with a `use` attribute of **encryption** MAY be used to document  
1351 supported encryption algorithms and settings, and public keys used to receive bulk encryption keys.

1352

## E59: SSO Response When Using HTTP-Artifact

1353 Change [SAMLBind] Section 3.6.5.2 at line 1173 to observe for clarity's sake that particular message  
1354 delivery mechanisms are not mandated for the "nested" message exchange that takes place as part of  
1355 the HTTP-Artifact binding.

1356 New:

1357 Note also that there is no mechanism defined to protect the integrity of the relationship between the artifact  
1358 and the "RelayState" value, if any. That is, an attacker can potentially recombine a pair of valid HTTP  
1359 responses by switching the "RelayState" values associated with each artifact. As a result, the  
1360 producer/consumer of "RelayState" information MUST take care not to associate sensitive state information  
1361 with the "RelayState" value without taking additional precautions (such as based on the information in the  
1362 SAML protocol message retrieved via artifact).

1363 **Finally, note that the use of the `Destination` attribute in the root SAML element of the protocol  
1364 message is unspecified by this binding, because of the message indirection involved.**

1365

## E60: Incorrect URI for Unspecified NameID Format

1366 Change [SAMLCore] Section 2.2.2 at line 460 to change the name identifier format from  
1367 `urn:oasis:names:tc:SAML:1.0:nameid-format:unspecified` to  
1368 `urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified`. This was a typographical error.

1369

## E61: Reference to Non-Existent Element

1370 Change [SAMLCore] Section 7.1.2 at lines 3160.

1371 Original:

1372 The following SAML protocol **elements** are intended specifically for use as extension points in an extension  
1373 schema; **their types** are set to abstract, and are thus usable only as the base of a derived type:

- 1374 • **<Request>** and RequestAbstractType
- 1375 • **<SubjectQuery>** and SubjectQueryAbstractType

1376 New:

1377 The following SAML protocol **constructs** are intended specifically for use as extension points in an  
1378 extension schema; **the types listed** are set to abstract, and are thus usable only as the base of a derived  
1379 type:

- 1380 • RequestAbstractType

1381 • <SubjectQuery> and SubjectQueryAbstractType

## 1382 **E62: TLS Keys in KeyDescriptor**

1383 Change [SAMLMeta] Section 2.4.1.1 at line 624 to specify more clearly how to interpret the  
1384 KeyDescriptor element's use attribute.

1385 New (just after the conclusion of the definition list for KeyDescriptorType):

1386 **A use value of "signing" means that the contained key information is applicable to both signing**  
1387 **and TLS/SSL operations performed by the entity when acting in the enclosing role.**

1388 **A use value of "encryption" means that the contained key information is suitable for use in**  
1389 **wrapping encryption keys for use by the entity when acting in the enclosing role.**

1390 **If the use attribute is omitted, then the contained key information is applicable to both of the above**  
1391 **uses.**

1392 The following schema fragment defines the <KeyDescriptor> element and its KeyDescriptorType  
1393 complex type:

## 1394 **E63: IdP Discovery Cookie Interpretation**

1395 Change [SAMLProf] Section 4.3.1 at line 1105 to clear up confusion over interpretation of the contents of  
1396 an IdP Discovery cookie. (Note that E32 specifies changes to Section 4 that result in a new Section 4.3.1  
1397 being inserted before the original one; E63 applies to the original Section 4.3.1.)

1398 New:

1399 Cookie syntax should be in accordance with IETF RFC 2965 [RFC2965] or [NSCookie]. The cookie MAY be  
1400 either session-only or persistent. This choice may be made within a deployment, but should apply uniformly  
1401 to all identity providers in the deployment. **Note that while a session-only cookie can be used, the intent**  
1402 **of this profile is not to provide a means of determining whether a user actually has an active session**  
1403 **with one or more of the identity providers stored in the cookie. The cookie merely identifies identity**  
1404 **providers known to have been used in the past. Service providers MAY instead rely on the**  
1405 **IsPassive attribute in their <samlp:AuthnRequest> message to probe for active sessions.**

## 1406 **E64: Liberty Moniker Used Inappropriately**

1407 Change [SAMLSec] Section 7.1.1.9, Impersonation without Reauthentication to replace an accidental use  
1408 of the moniker "Liberty" in place of "SAML V2.0".

1409 New:

1410 Cookies posted by identity providers MAY be used to support this validation process, though **LibertySAML**  
1411 **V2.0** does not mandate a cookie-based approach.

## 1412 **E65: Second-level StatusCode**

1413 Change various sections as follows in [SAMLCore] to constrain the optional second-level <StatusCode>  
1414 element used, and clarify that use of second-level codes is optional.

1415 Change section 3.3.2.2.1, lines 1817-1819.

1416 New:

1417 **If none of the specified classes or declarations can be satisfied in accordance with the rules below, then the**  
1418 **responder MUST return a <Response> message with a top-level <StatusCode> value of**  
1419 **urn:oasis:names:tc:SAML:2.0:status:Responder and MAY return a second-level**  
1420 **<StatusCode> of urn:oasis:names:tc:SAML:2.0:status:NoAuthnContext.**

1421 Change section 3.4.1.2, lines 2172-2173.

1422 New:

1423 In profiles specifying an active intermediary, the intermediary MAY examine the list and return a  
1424 <Response> message with an error <Status> and **optionally** a second-level <StatusCode> of

1425 Change section 3.4.1.5.1, lines 2282-2285.

1426 Original:

1427 An identity provider MUST NOT proxy a request where <ProxyCount> is set to zero. The identity  
1428 provider MUST return an error <Status> containing a second-level <StatusCode> value of  
1429 urn:oasis:names:tc:SAML:2.0:status:ProxyCountExceeded, unless it can directly  
1430 authenticate the presenter.

1431 New:

1432 **Unless the identity provider can directly authenticate the presenter, it MUST return a <Response>**  
1433 **message with a top-level <StatusCode> value of**  
1434 **urn:oasis:names:tc:SAML:2.0:status:Responder and MAY return a second-level**  
1435 **<StatusCode> value of urn:oasis:names:tc:SAML:2.0:status:ProxyCountExceeded.**

1436 Change section 3.8.3, lines 2729-2731.

1437 New:

1438 If the responder does not recognize the principal identified in the request, it MAY respond with an error  
1439 <Status>, **optionally** containing a second-level <StatusCode> of  
1440 urn:oasis:names:tc:SAML:2.0:status:UnknownPrincipal.

## 1441 **E66: Metadata and DNSSEC**

1442 Change [SAMLMeta] to update the DNSSEC reference from RFC 2535 to RFC 4035.

1443 Updated line 1253:

1444 It is RECOMMENDED that entities publish their resource records in signed zone files using ~~[RFC2535]~~  
1445 **[RFC4035]**

1446 Original at lines 1447-1448:

1447 [RFC2535] D. Eastlake. *Domain Name System Security Extensions*. IETF RFC 2535, March 1999. See  
1448 <http://www.ietf.org/rfc/rfc2535.txt>.

1449 New at lines 1447-1448:

1450 **[RFC4035] R. Arends et al. *Protocol Modifications for the DNS Security Extensions*. IETF RFC 4035,**  
1451 **March 2005. See <http://www.ietf.org/rfc/rfc4035.txt>.**

## 1452 **E68: Use of Multiple <KeyDescriptor> Elements**

1453 Add text to section 2.4.1.1 of [SAMLMeta] to clarify the meaning of identically-purposed  
1454 <KeyDescriptor> elements within a role.

1455 New at line 625:

1456 **The inclusion of multiple <KeyDescriptor> elements with the same use attribute (or no such**  
1457 **attribute) indicates that any of the included keys may be used by the containing role or affiliation. A**  
1458 **relying party SHOULD allow for the use of any of the included keys. When possible the signing or**  
1459 **encrypting party SHOULD indicate as specifically as possible which key it used to enable more**  
1460 **efficient processing.**

1461 The following schema fragment defines the <KeyDescriptor> element and its KeyDescriptorType  
1462 complex type:

## 1463 **E69: Semantics of <ds:KeyInfo> in <KeyDescriptor>**

1464 Add text to section 2.4.1.1 of [SAMLMeta] to clarify the limitations of the specification regarding the  
1465 semantics of various kinds of common key representations.

1466 New at line 625 (this change should appear after E68 above):

1467 **The <ds:KeyInfo> element is a highly generic and extensible means of communicating key**  
1468 **material. This specification takes no position on the allowable or suggested content of this element,**  
1469 **nor on its meaning to a relying party. As a concrete example, no implications of including an X.509**  
1470 **certificate by value or reference are to be assumed. Its validity period, extensions, revocation status,**  
1471 **and other relevant content may or may not be enforced, at the discretion of the relying party. The**  
1472 **details of such processing, and their security implications, are out of scope; they may, however, be**  
1473 **addressed by other SAML profiles.**

1474 The following schema fragment defines the <KeyDescriptor> element and its KeyDescriptorType  
1475 complex type:

## 1476 **E70: Obsolete reference to UUID URN namespace**

1477 Change [SAMLProf] to update the Internet Draft reference for the UUID URN namespace to RFC 4122.

1478 Updated Section 8.3.3.1, line 1836:

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

1481 Updated Section 8.4.3.1, line 1885:

1482 values are equal in the sense of [<http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-05.txt>][RFC4122].  
1483 The

1484 Original at lines 2111-2112:

1485 [Mealling] P Leach et al. *A UUID URN Namespace*. IETF Internet-Draft, December 2004. See  
1486 <http://www.ietf.org/internet-drafts/draft-mealling-uuid-urn-05.txt>.

1487 New at lines 2111-2112:

1488 [RFC4122] P. Leach et al. *A Universally Unique Identifier (UUID) URN Namespace*. IETF RFC 4122,  
1489 July 2005. See <http://www.ietf.org/rfc/rfc4122.txt>.

## 1490 **E71: Missing namespace definition in Profiles**

1491 Change [SAMLProf] to add the "xs" namespace prefix to the table in Section 1.

1492 New row of table in Section 1, between lines 267-268:

1493 **xs :**

1494 **<http://www.w3.org/2001/XMLSchema>**

1495 **This namespace is defined in the W3C XML Schema specification [Schema1]. In schema listings, this**  
1496 **is the default namespace and no prefix is shown. For clarity, the prefix is generally shown in**  
1497 **specification text when XML Schema-related constructs are mentioned.**

## 1498 **E74: Update XML Signature Reference**

1499 Update the XML Signature specification reference in [SAMLCore], [SAMLBind], [SAMLProf], [SAMLMeta],  
1500 [SAMLAuthCtx], [SAMLConf], [SAMLSec] to the "Second Edition". Also remove a stale non-normative  
1501 reference in [SAMLCore].

1502 Strike [SAMLCore], lines 3439-3440:

1503 [RFC 3075] D. Eastlake, J. Reagle, D. Solo. *XML Signature Syntax and Processing*. IETF RFC 3075,  
1504 March 2001. See <http://www.ietf.org/rfc/rfc3075.txt>.

1505 Original at [SAMLCore] lines 3415-3416, [SAMLBind] lines 1489-1491, [SAMLProf] lines 2205-2206,  
1506 [SAMLMeta] lines 1490-1491, [SAMLAuthCtx] lines 3926-3928, [SAMLConf] lines 410-412, [SAMLSec] lines  
1507 1078-1079:

1508 [XMLSig] D. Eastlake et al. *XML Signature Syntax and Processing*. World Wide Web Consortium,  
1509 February 2002. See <http://www.w3.org/TR/xmldsig-core/>. Note that this specification normatively  
1510 references [XMLSig-XSD], listed below.

1511 New at [SAMLCore] lines 3415-3416, [SAMLBind] lines 1489-1491, [SAMLProf] lines 2205-2206,  
1512 [SAMLMeta] lines 1490-1491, [SAMLAuthCtx] lines 3926-3928, [SAMLConf] lines 410-412, [SAMLSec]  
1513 lines 1078-1079:

1514 [XMLSig] D. Eastlake et al. *XML Signature Syntax and Processing, Second Edition*. World  
1515 Wide Web Consortium, June 2008. See <http://www.w3.org/TR/xmldsig-core/>.

## 1516 **E75: Clarify Handling of SubjectConfirmation in AuthnRequest**

1517 Change [SAMLCore] Section 3.4.1.4 to clarify an identity provider's obligation to return an error if can't  
1518 honor the requirements of a <SubjectConfirmation> element in an <AuthnRequest> message.

1519 New at line 2247:

1520 In such a case, the identifier's physical content MAY be different, but it MUST refer to the same principal. **If**  
1521 **the identity provider cannot or will not produce assertions with a strongly matching subject, then it**  
1522 **MUST return a <Response> with an error <Status>, and MAY return a second-level <StatusCode>**  
1523 **that reflects the reason for the failure.**

## 1524 **E76: Clarify nested validUntil/cacheDuration**

1525 Add text to [SAMLMeta] to clarify the processing of nested `validUntil` or `cacheDuration` attributes.

1526 New in Sections 2.3.1 and 2.3.2, before lines 336 and 409:

1527 When not used as the root element of a metadata instance, a `validUntil` or `cacheDuration` attribute  
1528 MAY be used to impose a shorter expiration or cache duration than that of the parent or root element, but  
1529 never a longer one; the smaller value takes precedence.

1530 New in Sections 2.4.1 and 2.5, before lines 589 and 972:

1531 A `validUntil` or `cacheDuration` attribute MAY be used to impose a shorter expiration or cache duration  
1532 than that of the parent or root element, but never a longer one; the smaller value takes precedence.

## 1533 **E77: Generalize scope of Metadata specification**

1534 Change [SAMLMeta] to address inadvertent language appearing to restrict use of SAML metadata to only  
1535 SAML profiles.

1536 New in Section 1, before line 137:

1537 A variety of extension points are also included to allow for the use of SAML metadata in non-SAML  
1538 specifications, profiles, and deployments, and such use is encouraged.

1539 Updated Section 2, lines 153-154:

1540 SAML metadata is organized around an extensible collection of roles representing common combinations of  
1541 SAML (and potentially non-SAML) protocols and profiles supported by system entities.

1542 Remove the word "SAML" from lines 226, 230, 311, 323, 332, 360, 372, 397, 403, 444, 478, 531, and  
1543 940.

1544

## E78: Reassignment of persistent identifiers

1545 Add text to [SAMLCore] Section 8.3.7, at line 3325, to clarify that non-reassignment to different principals  
1546 is a required property of "persistent" name identifiers.

1547 New:

1548 **Persistent name identifier values MUST NOT exceed a length of 256 characters. A given value, once**  
1549 **associated with a principal, MUST NOT be assigned to a different principal at any time in the future.**

1550

## E79: Clarification of SessionNotOnOrAfter

1551 Change [SAMLCore] Section 2.7.2, lines 1062-1065 to loosen wording around the  
1552 `SessionNotOnOrAfter` attribute and defer more explicitly to profiles.

1553 Original:

1554 Specifies a time instant at which the session between the principal identified by the subject and the SAML  
1555 authority issuing this statement MUST be considered ended. The time value is encoded in UTC, as  
1556 described in Section 1.3.3. There is no required relationship between this attribute and a `NotOnOrAfter`  
1557 condition attribute that may be present in the assertion.

1558 New:

1559 **Indicates an upper bound on sessions with the subject derived from the enclosing assertion. The**  
1560 **time value is encoded in UTC, as described in Section 1.3.3. There is no required relationship between this**  
1561 **attribute and a `NotOnOrAfter` condition attribute that may be present in the assertion. It's left to profiles**  
1562 **to provide specific processing rules for relying parties based on this attribute.**

1563

## E81: Algorithm statement in XML Signature profile

1564 Change [SAMLCore] Section 5.4.1, lines 2926-2927, and [SAMLMeta] Section 3.1.1, lines 1182-1183, to  
1565 relax the implication that RSA with SHA1 is the only supported algorithm.

1566 Original:

1567 SAML processors SHOULD support the use of RSA signing and verification for public key operations in  
1568 accordance with the algorithm identified by <http://www.w3.org/2000/09/xmldsig#rsa-sha1>.

1569 New:

1570 Any algorithm defined for use with the XML Signature specification MAY be used.

1571

## E82: Empty <ContactPerson> element

1572 Add text to [SAMLMeta] Section 2.3.2.2, before line 500, to clarify that child elements should be included.

1573 New:

1574 At least one child element SHOULD be present in a `<ContactPerson>` element.

1575

## E83: Weaken claim made about Exclusive C14N

1576 Change [SAMLCore] Section 5.4.3, lines 2939-2940, and [SAMLMeta] Section 3.1.3, lines 1196-1197, to  
1577 better explain the purpose of using exclusive canonicalization.

1578 Original:

1579 Use of Exclusive Canonicalization ensures that signatures created over SAML messages embedded in an  
1580 XML context can be verified independent of that context.

1581 New:

1582 Use of Exclusive Canonicalization facilitates the verification of signatures created over SAML messages  
1583 when placed into a different XML context than present during signing.

1584 Note that use of this algorithm alone does not guarantee that a particular signed object can be moved from  
1585 one context to another safely, nor is that a requirement of signed SAML objects in general, though it MAY be  
1586 required by particular profiles.

1587

---

## 3 Acknowledgments

1588

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

1589

1590

- Rob Philpott, EMC Corporation

1591

- Richard Franck, IBM

1592

- John Bradley, Individual

1593

- Scott Cantor, Internet2

1594

- Nate Klingenstein, Internet2

1595

- Bob Morgan, Internet2

1596

- Thomas Hardjono, M.I.T.

1597

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

1598

- Frederick Hirsch, Nokia Corporation

1599

- Paul Madsen, NTT Corporation

1600

- Ari Kermaier, Oracle Corporation

1601

- Hal Lockhart, Oracle Corporation

1602

- Anil Saldhana, Red Hat

1603

- Kent Spaulding, Skyworth TTG Holdings Limited

1604

- Duane DeCouteau, Veterans Health Administration

1605

- David Staggs, Veterans Health Administration

1606

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

1607

1608