SAML Version 2.0 Errata 05

OASIS Approved Errata

01 May 2012

Specification URIs

This version:

http://docs.oasis-open.org/security/saml/v2.0/errata05/os/saml-v2.0-errata05-os.odt (Authoritative)
http://docs.oasis-open.org/security/saml/v2.0/errata05/os/saml-v2.0-errata05-os.html
http://docs.oasis-open.org/security/saml/v2.0/errata05/os/saml-v2.0-errata05-os.pdf

Previous version:

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

Latest version:

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

Technical Committee:

OASIS Security Services (SAML) TC

Chairs:

Thomas Hardjono (hardjono@mit.edu), M.I.T.
Nate Klingenstein (ndk@internet2.edu), Internet2

Editor:

Scott Cantor (cantor.2@osu.edu), Internet2

Related work:

This specification is related to the OASIS Standard *OASIS Security Assertion Markup Language (SAML) V2.0*, comprised of the following documents:

- *Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0* http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf
- *Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0* http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf
- *Conformance Requirements for the OASIS Security Assertion Mark Markup Language (SAML) V2.0* http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf
- *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0* http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf
- *Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0* http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf
- *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0* http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf
- *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0* http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf
- *Security Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0* 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 OASIS Security Services (SAML) TC on the above date. The level of approval is also listed above. Check the “Latest version” location noted above for possible later revisions of this document.

Technical Committee members should send comments on this Work Product to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee’s web page at http://www.oasis-open.org/committees/security/.

For information on whether any patents have been disclosed that may be essential to implementing this Work Product, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (http://www.oasis-open.org/committees/security/ipr.php).

Citation format:

When referencing this Work Product the following citation format should be used:
[SAML v2.0 Errata 05]
SAML Version 2.0 Errata 05. 01 May 2012. OASIS Approved Errata. http://docs.oasis-open.org/security/saml/v2.0/errata05/os/saml-v2.0-errata05-os.html.

Notices

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

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

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

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

Table of Contents

1 Introduction

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

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

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

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

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

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

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

> This is yet another original specification sentence which contains a~~**n inappropriately**~~ long description.

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

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

1.1 Normative References

[SAMLAuthCtx] OASIS Standard, *Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0*, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf

[SAMLBind] OASIS Standard, *Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0*, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf

[SAMLConf] OASIS Standard, *Conformance Requirements for the OASIS Security Assertion Mark Markup Language (SAML) V2.0*, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf

[SAMLCore] OASIS Standard, *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[SAMLMeta] OASIS Standard, *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

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

[SAMLSec] OASIS Standard, *Security Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0*, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf

1.2 Non-Normative References

[Sec2011] *From Multiple Credentials to Browser-based Single Sign-On: Are We More Secure?*, in the Proceedings of the 26th IFIP TC-11 International Information Security Conference (SEC 2010), Luzern, Switzerland, June 7-9, 2011. http://www.ai-lab.it/armando/pub/sec2011.pdf

[Enc2011] T. Jager, J. Somorovsky. *How to Break XML Encryption*. October 2011. http://www.nds.rub.de/media/nds/veroeffentlichungen/2011/10/22/HowToBreakXMLenc.pdf

[RFC3218] E. Rescorla. *Preventing the Million Message Attack on Cryptographic Message Syntax*. IETF RFC 3218, January 2002. http://www.ietf.org/rfc/rfc3218.txt

[800-38D] M. Dworkin. *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*. November 2007. http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf

2 Approved Errata

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

E0: Incorrect Section Reference

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

E1: Relay State for HTTP Redirect

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

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

New:

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

E2: Metadata Clarifications for HTTP Artifact Binding

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

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

New:

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

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

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

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

E6: Clarify Constraints on Encrypted NameID

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

New:

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

E7: Metadata for Agreeing to Sign Authentication Requests

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

New at line 710:

> **The `WantAuthnRequestsSigned` attribute is intended to indicate to service providers whether or not they can expect an unsigned `<AuthnRequest>` message to be accepted by the identity provider. The identity provider is not obligated to reject unsigned requests nor is a service provider obligated to sign its requests, although it might reasonably expect an unsigned request will be rejected. In some cases, a service provider may not even know which identity provider will ultimately receive and respond to its requests, so the use of this attribute in such a case cannot be strictly defined.**
>
> **Furthermore, note that the specific method of signing that would be expected is binding dependent. The HTTP Redirect binding (see [SAMLBind]) requires that the signature be applied to the URL-encoded value rather than placed within the XML message, while other bindings generally permit the signature to be within the message in the usual fashion.**
>
> The following schema fragment defines the `<IDPSSODescriptor>` element and its IDPSSODescriptorType complex type:

New at lines 741-742:

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

New at lines 744-747:

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

E8: SLO and NameID Termination

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

Original:

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

New:

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

E10: Logout Request Reason Mismatch with Schema

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

New:

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

E11: Improperly Labeled Feature

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

Original labels:

> Name Identifier Management, HTTP Redirect (IdP-initiated)
> Name Identifier Management, SOAP (IdP-initiated)
> Name Identifier Management, HTTP Redirect
> Name Identifier Management, SOAP

New labels:

> **Name Identifier Management (IdP-Initiated), HTTP Redirect**
> **Name Identifier Management (IdP-Initiated), SOAP**
> **Name Identifier Management (SP-Initiated), HTTP Redirect**
> **Name Identifier Management (SP-Initiated), SOAP**

E12: Clarification on ManageNameIDRequest

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

New [SAMLCore] at lines 2412-2413:

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

New [SAMLCore] at line 2438:

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

New [SAMLProf] at lines 1320-23121:

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

E13: Inaccurate Description of Authorization Decision

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

New:

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

E14: AllowCreate

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

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

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

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

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

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

> **The `AllowCreate` attribute may be used by some deployments to influence the creation of state maintained by the identity provider pertaining to the use of a name identifier (or any other persistent, uniquely identifying attributes) by a particular relying party, for purposes such as dynamic identifier or attribute creation, tracking of consent, subsequent use of the Name Identifier Management protocol (see Section 3.6), or other related purposes.**
>
> **When “false”, the requester tries to constrain the identity provider to issue an assertion only if such state has already been established or is not deemed applicable by the identity provider to the use of an identifier. Thus, this does not prevent the identity provider from assuming such information exists outside the context of this specific request (for example, establishing it in advance for a large number of principals).**
>
> **A value of “true” permits the identity provider to take any related actions it wishes to fulfill the request, subject to any other constraints imposed by the request and policy (the `IsPassive` attribute, for example).**
>
> **Generally, requesters cannot assume specific behavior from identity providers regarding the initial creation or association of identifiers on their behalf, as these are details left to implementations or deployments. Absent specific profiles governing the use of this attribute, it might be used as a hint to identity providers about the requester’s intention to store the identifier or link it to a local value.**
>
> **A value of “false” might be used to indicate that the requester is not prepared or able to do so and save the identity provider wasted effort.**
>
> **Requesters that do not make specific use of this attribute SHOULD generally set it to “true” to maximize interoperability.**
>
> **The use of the AllowCreate attribute MUST NOT be used and SHOULD be ignored in conjunction with requests for or assertions issued with name identifiers with a `Format` of `urn:oasis:names:tc:SAML:2.0:nameid-format:transient` (they preclude any such state in and of themselves).**

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

> A service provider also uses this message to register or change the `SPProvidedID` value to be included when the underlying name identifier is used to communicate with it, or to terminate the use of a name identifier between itself and the identity provider.
>
> **Note that this protocol is typically not used with “transient” name identifiers, since their value is not intended to be managed on a long-term basis.**

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

> A service provider also uses this message to register or change the `SPProvidedID` value to be included when the underlying name identifier is used to communicate with it, or to terminate the use of a name identifier between itself and the identity provider.
>
> **This protocol MUST NOT be used in conjunction with the `urn:oasis:names:tc:SAML:2.0:nameidformat:transient` `<NameID>` Format.**

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

> If the `<Terminate>` element is included in the request, the requesting provider is indicating that (in the case of a service provider) it will no longer accept assertions from the identity provider or (in the case of an identity provider) it will no longer issue assertions to the service provider about the principal. The receiving provider can perform any maintenance with the knowledge that the relationship represented by the name identifier has been terminated. It can choose to invalidate the active session(s) of a principal for whom a relationship has been terminated.
>
> **If the receiving provider is maintaining state associated with the name identifier, such as the value of the identifier itself (in the case of a pair-wise identifier), an `SPProvidedID` value, the sender’s consent to the identifier’s creation/use, etc., then the receiver can perform any maintenance with the knowledge that the relationship represented by the name identifier has been terminated.**
>
> **Any subsequent operations performed by the receiver on behalf of the sender regarding the principal (for example, a subsequent `<AuthnRequest>`) SHOULD be carried out in a manner consistent with the absence of any previous state.**
>
> **Termination is potentially the cleanup step for any state management behavior triggered by the use of the `AllowCreate` attribute in the Authentication Request protocol (see Section 3.4). Deployments that do not make use of that attribute are likely to avoid the use of the `<Terminate>` element or would treat it as a purely advisory matter.**
>
> **Note that in most cases (a notable exception being the rules surrounding the `SPProvidedID` attribute), there are no requirements on either identity providers or service providers regarding the creation or use of persistent state. Therefore, no explicit behavior is mandated when the `<Terminate>` element is received. However, if persistent state is present pertaining to the use of an identifier (such as if an `SPProvidedID` attribute was attached), the `<Terminate>` element provides a clear indication that this state SHOULD be deleted (or marked as obsolete in some fashion).**

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

> If the identity provider cannot or will not satisfy the request, it MUST respond with a `<Response>` message containing an appropriate error status code or codes.
>
> **If the service provider wishes to permit the identity provider to establish a new identifier for the principal if none exists, it MUST include a `<NameIDPolicy>` element with the `AllowCreate` attribute set to "true". Otherwise, only a principal for whom the identity provider has previously established an identifier usable by the service provider can be authenticated successfully.**

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

> If the identity provider cannot or will not satisfy the request, it MUST respond with a `<Response>` message containing an appropriate error status code or codes.
>
> **This profile does not provide any guidelines for the use of `AllowCreate`; see [SAMLCore] for normative rules on using `AllowCreate`.**

E15: NameID Policy Adherence

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

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

> The special Format value urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted indicates that the resulting assertion(s) MUST contain <EncryptedID> elements instead of plaintext. The underlying name identifier's unencrypted form can be of any type supported by the identity provider for the requested subject.
>
> **When a Format defined in Section Error: Reference source not found8.3 other than urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified or urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted is used, then if the identity provider returns any assertions:**
>
> - **the Format value of the <NameID> within the <Subject> of any <Assertion> MUST be identical to the Format value supplied in the <NameIDPolicy>, and**
>
> - **if SPNameQualifier is not omitted in <NameIDPolicy>, the SPNameQualifier value of the <NameID> within the <Subject> of any <Assertion> MUST be identical to the SPNameQualifier value supplied in the <NameIDPolicy>.**

E17: Authentication Response IssuerName vs. Assertion IssuerName

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

Original:

> **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.

New:

> **If the <Response> message is signed or if an enclosed assertion is encrypted, then the <Issuer> element MUST be present. Otherwise it MAY be omitted. 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.

E18: Reference to Identity Provider Discovery Service in ECP Profile

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

New:

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

E19: Clarification on Error Processing

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

Original at Section 3.2.2.1, lines 310-317:

> The SAML responder **MUST** return **either a SAML response element within the body of another SOAP message or generate a SOAP fault**. The SAML responder MUST NOT include more than one SAML response per SOAP message or include any additional XML elements in the SOAP body. **If a SAML responder cannot, for some reason, process a SAML request, it MUST generate a SOAP fault.** SOAP

> fault codes **MUST** NOT be sent for errors within the SAML problem domain, for example, inability to find an extension schema or as a signal that the subject is not authorized to access a resource in an authorization query. (SOAP 1.1 faults and fault codes are discussed in [SOAP11] Section 4.1.)

New at Section 3.2.2.1, lines 310-317:

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

Original at Section 3.2.3.3, line 378:

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

New at Section 3.2.3.3, line 378:

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

E20: ECP SSO Profile and Metadata

Change [SAMLProf] at line 1081 to add a new subsection, Section 4.2.6, in order to add metadata considerations to the ECP profile.
New (small portion of previous subsection shown):

> The ECP SHOULD be authenticated to the identity provider, such as by maintaining an authenticated session. Any HTTP exchanges subsequent to the delivery of the `<AuthnRequest>` message and before the identity provider returns a `<Response>` MUST be securely associated with the original request.
>
> **4.2.6 Use of Metadata**
>
> **The rules specified in the browser SSO profile in Section 4.1.6 apply here as well. Specifically, the indexed endpoint element `<md:AssertionConsumerService>` with a binding of `urn:oasis:names:tc:SAML:2.0:bindings:PAOS` MAY be used to describe the supported binding and location(s) to which an identity provider may send responses to a service provider using this profile. IN addition, the endpoint `<md:SingleSignOnService>` with a binding of `urn:oasis:names:tc:SAML:2.0:bindings:SOAP` MAY be used to describe the supported binding and location(s) to which an service provider may send requests to an identity provider using this profile.**

E21: PAOS Version

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

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

E22: Error in Profile/ECP

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

E24: HTTPS in URI Binding

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

> 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)**.

New:

> 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**.

E25: Metadata Feature in Conformance

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

New in Table 2:

Feature	IdP	IdP Lite	SP	SP Lite	ECP
Metadata Structures	**OPT**	**OPT**	**OPT**	**OPT**	**N/A**
Metadata Interoperation	**OPT**	**OPT**	**OPT**	**OPT**	**N/A**

New in Table 4:

Feature	Authn	Attrib	Authz	Requester
Metadata Structures	**OPT**	**OPT**	**OPT**	**OPT**
Metadata Interoperation	**OPT**	**OPT**	**OPT**	**OPT**

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

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

- **The** assertion**(s) containing a bearer subject confirmation** MUST contain an `<AudienceRestriction>` including the service provider's unique identifier as an `<Audience>`.

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

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

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

- 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 **responding** identity provider; the `Format` attribute MUST be omitted or have a value of `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`. **Note that this profile assumes a single responding identity provider, and all assertions in a response MUST be issued by the same entity.**

- **If multiple assertions are included, then each assertion's `<Subject>` element MUST refer to the same principal. It is allowable for the content of the `<Subject>` elements to differ (e.g. using different `<NameID>` or alternative `<SubjectConfirmation>` elements).**

- **Any assertion issued for consumption using this profile MUST contain a `<Subject>` element with at least one `<SubjectConfirmation>` element containing a `Method` of `urn:oasis:names:tc:SAML:2.0:cm:bearer`. Such an assertion is termed a bearer assertion. Bearer assertions MAY contain additional `<SubjectConfirmation>` elements.**

- **Assertions without a bearer `<SubjectConfirmation>` MAY also be included; processing of additional assertions or `<SubjectConfirmation>` elements is outside the scope of this profile.**

- **At lease one bearer `<SubjectConfirmation>` element MUST contain a `<SubjectConfirmationData>` element that itself MUST contain 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 [PE52]confirmed by the relying party. It MAY also 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.**

- **The set of one or more bearer assertions MUST contain at least one `<AuthnStatement>` that reflects the authentication of the principal to the identity provider. Multiple `<AuthnStatement>` elements MAY be included, but the semantics of multiple statements is not defined by this profile.**

- **If the identity provider supports the Single Logout profile, defined in Section Error: Reference source not found, any authentication statements MUST include a `SessionIndex` attribute to enable per-session logout requests by the service provider.**

- Other statements MAY be included in the **bearer** assertion(s) at the discretion of the identity provider. In particular, `<AttributeStatement>` elements MAY be included. The `<AuthnRequest>` MAY contain an `AttributeConsumingServiceIndex` XML attribute referencing information about desired or required attributes in [SAMLMeta]. The identity provider MAY ignore this, or send other attributes at its discretion.

- **Each bearer** assertion MUST contain an `<AudienceRestriction>` including the service provider's unique identifier as an `<Audience>`.

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

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

Original at Section 4.1.4.3, lines 576-591:

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

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

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

- Verify that any assertions relied upon are valid in other respects**.**

> • If any bearer `<SubjectConfirmationData>` includes an `Address` attribute, the service provider MAY check the user agent's client address against it.
>
> • Any assertion which is not valid, or whose subject confirmation requirements cannot be met SHOULD be discarded and SHOULD NOT be used to establish a security context for the principal.
>
> • If an `<AuthnStatement>` used to establish a security context for the principal contains a `SessionNotOnOrAfter` attribute, the security context SHOULD be discarded once this time is reached, unless the service provider reestablishes the principal's identity by repeating the use of this profile.

New at Section 4.1.4.3, lines 576-591:

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

Original at Section 4.1.4.5, lines 600-601:

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

New at Section 4.1.4.5, lines 600-601:

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

E27: Incorrect Step Number in ECP Profile

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

E28: Profile Labeling in Conformance

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

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

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

E29: Incomplete Listing of Features in Conformance

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**

E30: Key Replacement

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

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

New:

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

E31: Various Minor Errors in Binding

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

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

Original at Section 3.5.3, line 785:

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

New at Section 3.5.3, line 785:

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

Original at Section 3.6.5, line 1136:

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

New at Section 3.6.5, line 1136:

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

Original at Section 3.6.5, line 1397:

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

New at Section 3.6.5, line 1397:

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

E32: Missing Required Information in Profiles

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

> **4.3.1 Required Information**
>
> **Identification:** urn:oasis:names:tc:SAML:2.0:profiles:SSO:idp-discovery
>
> **Contact information:** security-services-comment@lists.oasis-open.org
>
> **Description:** Given below.

> **Updates:** None.

E33: References to Assertion Request Protocol

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

E34: RequestedAttribute Section Heading

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 consistency in reflecting element nesting in the document outline.

E35: Response Consumer URL Rules and Example

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

Original at Section 4.2.4.1, lines 906-908:

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

New at lines Section 4.2.4.1, 906-908:

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

Original at Section 4.2.4.3, line 964:

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

New at Section 4.2.4.3, line 964:

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

E36: Clarification on Action Element

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

Original:

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

New:

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

E37: Clarification in Metadata on Indexed Endpoints

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

Original:

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

New:

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

E38: Clarification Regarding Index on <LogoutRequest>

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

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

New at [SAMLCore] Section 3.7.1, line 2546:

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

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

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

E39: Error in SAML Profile Example

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

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

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

New:

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

E40: Holder of Key

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

Original:

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

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

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

E41: EndpointType ResponseLocation Clarification in Metadata

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

New:

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

E42: Match Authorities to Queries in Conformance

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

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

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

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

Original:

> **6.2 Combining Signatures and Encryption**

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

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

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

New:

6.2 Key and Data Referencing Guidelines

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

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

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

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

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

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

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

6.3 Examples

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

```
<saml:EncryptedID   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
  <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    Id="Encrypted_DATA_ID"
    Type="http://www.w3.org/2001/04/xmlenc#Element">
    <xenc:EncryptionMethod
    Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:RetrievalMethod URI="#Encrypted_KEY_ID"
      Type="http://www.w3.org/2001/04/xmlenc#EncryptedKey"/>
    </ds:KeyInfo>
    <xenc:CipherData>
      <xenc:CipherValue>Nk4W4mx...</xenc:CipherValue>
    </xenc:CipherData>
  </xenc:EncryptedData>

  <xenc:EncryptedKey xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    Id="Encrypted_KEY_ID">
    <xenc:EncryptionMethod
    Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
    <xenc:CipherData>
    <xenc:CipherValue>PzA5X...</xenc:CipherValue>
    </xenc:CipherData>
    <xenc:ReferenceList>
      <xenc:DataReference URI="#Encrypted_DATA_ID"/>
    </xenc:ReferenceList>
  </xenc:EncryptedKey>
```

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

```
<saml:EncryptedAttribute
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
  <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    Id="Encrypted_DATA_ID"
    Type="http://www.w3.org/2001/04/xmlenc#Element">
    <xenc:EncryptionMethod
      Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <xenc:EncryptedKey Id="Encrypted_KEY_ID">
        <xenc:EncryptionMethod
          Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
        <xenc:CipherData>
          <xenc:CipherValue>SDFSDF... </xenc:CipherValue>
        </xenc:CipherData>
      </xenc:EncryptedKey>
    </ds:KeyInfo>
    <xenc:CipherData>
      <xenc:CipherValue>Nk4W4mx...</xenc:CipherValue>
    </xenc:CipherData>
  </xenc:EncryptedData>
</saml:EncryptedAttribute>
```

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

```
<saml:EncryptedAssertion
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
  <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    Id="Encrypted_DATA_ID"
    Type="http://www.w3.org/2001/04/xmlenc#Element">
    <xenc:EncryptionMethod
      Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:KeyName>MULTICAST_KEY_NAME</ds:KeyName>
    </ds:KeyInfo>
    <xenc:CipherData>
      <xenc:CipherValue>Nk4W4mx...</xenc:CipherValue>
    </xenc:CipherData>
  </xenc:EncryptedData>

  <xenc:EncryptedKey xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    Id="Encrypted_KEY_ID_1" Recipient="https://sp1.org">
    <xenc:EncryptionMethod
      Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:KeyName>KEY_NAME_1</ds:KeyName>
    </ds:KeyInfo>
    <xenc:CipherData>
      <xenc:CipherValue>xyzABC...</xenc:CipherValue>
    </xenc:CipherData>
    <xenc:ReferenceList>
      <xenc:DataReference URI="#Encrypted_DATA_ID"/>
    </xenc:ReferenceList>

    <xenc:CarriedKeyName>MULTICAST_KEY_NAME</xenc:CarriedKeyName>
  </xenc:EncryptedKey>

  <xenc:EncryptedKey xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
    Id="Encrypted_KEY_ID_2" Recipient="https://sp2.org">
    <xenc:EncryptionMethod
      Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:KeyName>KEY_NAME_2</ds:KeyName>
    </ds:KeyInfo>
    <xenc:CipherData>
      <xenc:CipherValue>abcXYZ...</xenc:CipherValue>
    </xenc:CipherData>
    <xenc:ReferenceList>
      <xenc:DataReference URI="#Encrypted_DATA_ID"/>
    </xenc:ReferenceList>

    <xenc:CarriedKeyName>MULTICAST_KEY_NAME</xenc:CarriedKeyName>
  </xenc:EncryptedKey>
</saml:EncryptedAssertion>
```

E45: AuthnContext Comparison Order

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

Original at Section 3.3.2.2.1, lines1815-1819:

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

New at Section 3.3.2.2.1, lines 1815-1819:

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

Original at Section 3.3.2.2.1, line 1826:

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

New at Section 3.3.2.2.1, line 1826:

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

E46: AudienceRestriction Clarifications

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

Original:

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

New:

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

E47: Clarification on SubjectConfirmation

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

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

> **If the `<SubjectConfirmation>` element in an assertion subject contains an identifier the issuer authorizes the attesting entity to wield the assertion on behalf of that subject. A relying party MAY apply additional constraints on the use of such an assertion at its discretion, based upon the identities of both the subject and the attesting entity.**
>
> **If an assertion is issued for use by an entity other than the subject, then that entity SHOULD be identified in the `<SubjectConfirmation>` element.**
>
> The following schema fragment defines the `<SubjectConfirmation>` element and its SubjectConfirmationType complex type:

Original at [SAMLProf] Section 3.1, line 336:

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

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

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

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

> **If the `<SubjectConfirmation>` element in an assertion subject contains an identifier the issuer authorizes the attesting entity to wield the assertion on behalf of that subject. A relying party MAY apply additional constraints on the use of such an assertion at its discretion, based upon the identities of both the subject and the attesting entity.**
>
> **If an assertion is issued for use by an entity other than the subject, then that entity SHOULD be identified in the `<SubjectConfirmation>` element.**
>
> Example: The holder of the key named "By-Tor" or the holder of the key named "Snow Dog" can confirm itself as the subject.

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

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

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

> The subject of the assertion is **considered to be an acceptable attesting entity for** the assertion **by the asserting party**, subject to optional constraints on confirmation using the attributes that MAY be present in the `<SubjectConfirmationData>` element, as defined by [SAMLCore].
>
> **If the intended bearer is known by the asserting party to be an entity other than the subject, then the asserting party SHOULD identify that entity to the relying party by including a SAML identifier representing it in the enclosing `<SubjectConfirmation>` element.**
>
> **If multiple attesting entities are to be permitted to use the assertion based on bearer semantics, then multiple `<SubjectConfirmation>` elements SHOULD be included.**

E48: Clarification on Encoding for Binary Values in LDAP Profile

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

Change [SAMLProf] at line 1762. Original:

> 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"`.

New:

> 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"`.

E49: Clarification on Attribute Name Format

Change [SAMLCore] Section 2.7.3.1 at line 1217 to clarify the relationship between an attribute's `NameFormat` setting and its syntax.
New (add text to the end of the definition of `<AttributeValue>`):

> `<AttributeValue>` [Any Number]
>
> 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.
>
> **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.**
>
> **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`.**

E50: Clarification on SSL Ciphersuites

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.
New at Section 4, line 235:

> 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.**

New at Section 5, line 257:

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

E51: Schema Type of Contents of <AttributeValue>

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

E52: Clarification on NotOnOrAfter Attribute for Subject Confirmation

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

> 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.

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

> 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 **confirmed by the relying party**. It MAY contain an `Address` attribute limiting the client address from which the assertion can be delivered.

E53: Correction to LDAP/X.500 Profile Attribute

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

> 8.2 X.500/LDAP Attribute Profile **– Deprecated**

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

E54: Corrections to ECP URN

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

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

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

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

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

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

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

E55: Language Cleanup Around Name Identifier Management

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

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

> If the `<Terminate>` element is included in the request, the requesting provider is indicating that (in the case of a service provider) it will no longer accept assertions from the identity provider or (in the case of an identity provider) it will no longer issue assertions to the service provider **about the principal**. The receiving provider can perform any maintenance with the knowledge that the relationship represented by the name identifier has been terminated.
>
> If the service provider requests that its identifier for the principal be changed by including a `<NewID>` (or `<NewEncryptedID>`) element, the identity provider MUST include the element's content as the `SPProvidedID` when subsequently communicating to the service provider **regarding this principal**.
>
> If the identity provider requests that its identifier for the principal be changed by including a `<NewID>` (or `<NewEncryptedID>`) element, the service provider MUST use the element's content as the `<saml:NameID>` element content when subsequently communicating with the identity provider **regarding this principal**.

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

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

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

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

> The element's `SPNameQualifier` attribute, if present, MUST contain the unique identifier of the service provider or affiliation of providers for whom the identifier was generated (see Section 8.3.6). It MAY be omitted if the element is contained in a message intended only for consumption directly by the service provider, and the value would be the unique identifier of that service provider.
>
> ~~**The element's `SPProvidedID` attribute MUST contain the alternative identifier of the principal most recently set by the service provider or affiliation, if any (see Section 3.6). If no such identifier has been established, then the attribute MUST be omitted.**~~

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

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

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

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

E56: Confirmation Method Typo

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

E57: SAMLmime Reference

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

Original:

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

New:

> [SAMLmime] **OASIS Security Services Technical Committee (SSTC), "application/samlassertion+xml MIME Media Type Registration", IANA**

> **MIME Media Types Registry application/samlassertion+xml, December 2004. See http://www.iana.org/assignments/media-types/application/samlassertion+xml.**

E58: KeyDescriptor Typos in Profiles

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

Original:

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

New:

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

E59: SSO Response When Using HTTP-Artifact

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

New:

> Note also that there is no mechanism defined to protect the integrity of the relationship between the artifact and the "RelayState" value, if any. That is, an attacker can potentially recombine a pair of valid HTTP responses by switching the "RelayState" values associated with each artifact. As a result, the producer/consumer of "RelayState" information MUST take care not to associate sensitive state information with the "RelayState" value without taking additional precautions (such as based on the information in the SAML protocol message retrieved via artifact).
>
> **Finally, note that the use of the Destination attribute in the root SAML element of the protocol message is unspecified by this binding, because of the message indirection involved.**

E60: Incorrect URI for Unspecified NameID Format

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

E61: Reference to Non-Existent Element

Change [SAMLCore] Section 7.1.2 at lines 3160.

Original:

> The following SAML protocol **element**s are intended specifically for use as extension points in an extension schema; **their types** are set to abstract, and are thus usable only as the base of a derived type:
>
> - **<Request> and** RequestAbstractType
> - <SubjectQuery> and SubjectQueryAbstractType

New:

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

> • RequestAbstractType
>
> • `<SubjectQuery>` and SubjectQueryAbstractType

E62: TLS Keys in KeyDescriptor

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

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

> **A `use` value of `"signing"` means that the contained key information is applicable to both signing and TLS/SSL operations performed by the entity when acting in the enclosing role.**
>
> **A `use` value of `"encryption"` means that the contained key information is suitable for use in wrapping encryption keys for use by the entity when acting in the enclosing role.**
>
> **If the `use` attribute is omitted, then the contained key information is applicable to both of the above uses.**
>
> The following schema fragment defines the `<KeyDescriptor>` element and its KeyDescriptorType complex type:

E63: IdP Discovery Cookie Interpretation

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

New:

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

E64: Liberty Moniker Used Inappropriately

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

New:

> Cookies posted by identity providers MAY be used to support this validation process, though ~~**Liberty**~~**SAML V2.0** does not mandate a cookie-based approach.

E65: Second-level StatusCode

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

Change section 3.3.2.2.1, lines 1817-1819.

New:

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

Change section 3.4.1.2, lines 2172-2173.

New:

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

Change section 3.4.1.5.1, lines 2282-2285.

Original:

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

New:

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

Change section 3.8.3, lines 2729-2731.

New:

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

E66: Metadata and DNSSEC

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

Updated line 1253:

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

Original at lines 1447-1448:

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

New at lines 1447-1448:

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

E68: Use of Multiple <KeyDescriptor> Elements

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

New at line 625:

> **The inclusion of multiple `<KeyDescriptor>` elements with the same `use` attribute (or no such attribute) indicates that any of the included keys may be used by the containing role or affiliation. A relying party SHOULD allow for the use of any of the included keys. When possible the signing or encrypting party SHOULD indicate as specifically as possible which key it used to enable more efficient processing.**
>
> The following schema fragment defines the `<KeyDescriptor>` element and its KeyDescriptorType complex type:

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

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

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

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

E70: Obsolete reference to UUID URN namespace

Change [SAMLProf] to update the Internet Draft reference for the UUID URN namespace to RFC 4122.
Updated Section 8.3.3.1, line 1836:

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

Updated Section 8.4.3.1, line 1885:

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

Original at lines 2111-2112:

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

New at lines 2111-2112:

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

E71: Missing namespace definition in Profiles

Change [SAMLProf] to add the "xs" namespace prefix to the table in Section 1.
New row of table in Section 1, between lines 267-268:

> **xs:**
>
> **http://www.w3.org/2001/XMLSchema**
>
> **This namespace is defined in the W3C XML Schema specification [Schema1]. In schema listings, this is the default namespace and no prefix is shown. For clarity, the prefix is generally shown in specification text when XML Schema-related constructs are mentioned.**

E74: Update XML Signature Reference

Update the XML Signature specification reference in [SAMLCore], [SAMLBind], [SAMLProf], [SAMLMeta], [SAMLAuthCtx], [SAMLConf], [SAMLSec] to the "Second Edition". Also remove a stale non-normative reference in [SAMLCore].
Strike [SAMLCore], lines 3439-3440:

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

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

> If the `Format` value is omitted or set to `urn:oasis:names:tc:SAML:2.0:nameidformat:unspecified`[XMLSig] D. Eastlake et al. XML-Signature Syntax and Processing. World Wide Web Consortium, February 2002. See

> http://www.w3.org/TR/xmldsig-core/. Note that this specification normatively references [XMLSig-XSD], listed below.

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

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

E75: Clarify Handling of SubjectConfirmation in AuthnRequest

Change [SAMLCore] Section 3.4.1.4 to clarify an identity provider's obligation to return an error if it can't honor the requirements of a `<SubjectConfirmation>` element in an `<AuthnRequest>` message.
New at line 2247:

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

E76: Clarify nested validUntil/cacheDuration

Add text to [SAMLMeta] to clarify the processing of nested `validUntil` or `cacheDuration` attributes.
New in Sections 2.3.1 and 2.3.2, before lines 336 and 409:

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

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

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

E77: Generalize scope of Metadata specification

Change [SAMLMeta] to address inadvertent language appearing to restrict use of SAML metadata to only SAML profiles.
New in Section 1, before line 137:

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

Updated Section 2, lines 153-154:

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

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

E78: Reassignment of persistent identifiers

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

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

E79: Clarification of SessionNotOnOrAfter

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

Original:

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

New:

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

E81: Algorithm statement in XML Signature profile

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

Original:

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

New:

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

E82: Empty <ContactPerson> element

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

New:

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

E83: Weaken claim made about Exclusive C14N

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

Original:

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

New:

> Use of Exclusive Canonicalization facilitates the verification of signatures created over SAML messages when placed into a different XML context than present during signing.
>
> Note that use of this algorithm alone does not guarantee that a particular signed object can be moved from one context to another safely, nor is that a requirement of signed SAML objects in general, though it MAY be required by particular profiles

E84: Incorrect NameID Format constant

Change [SAMLCore] Section 3.4.1.1., lines 2133-2134 to fix reference to incorrect constant.

Original:

> If the `Format` value is omitted or set to
> `urn:oasis:names:tc:SAML:2.0:nameidformat:unspecified`

New:

> If the `Format` value is omitted or set to `urn:oasis:names:tc:SAML:1.1:nameidformat:unspecified`

E85: Conflicting language on profile error responses

Add text to [SAMLProf] Section 4.1.3.5., before line 487, to more strongly encourage support for returning error responses to Service Providers with appropriate security considerations.
New:

> Identity provider implementations SHOULD support the issuance of `<saml2p:Response>` messages (with appropriate status codes) in the event of an error condition, provided that the user agent remains available and an acceptable location to which to deliver the response is available. The criteria for "acceptability" of a response location are not formally specified, but are subject to identity provider policy and reflect its responsibility to protect users from being sent to untrusted or possibly malicious parties.

E86: Pseudorandom requirement for persistent NameID format

Change [SAMLCore] Section 8.3.7., lines 3321-3323 to relax requirement for cryptographic pseudo-randomness in the generation of persistent name identifier values.
Original:

> Persistent name identifiers generated by identity providers MUST be constructed using pseudo-random values that have no discernible correspondence with the subject's actual identifier (for example, username).

New:

> Persistent name identifiers generated by identity providers MUST be constructed using values that have no discernible correspondence with the subject's actual identity (for example, username). They MAY be pseudo-random values, or generated in any other manner, provided there is no guessable relationship between the value and the subject's underlying identity, and that they are unique within the range of values generated by a given identity provider for a given service provider or affiliation of providers.

E87: Clarify default rules for <md:AttributeConsumingService>

Change [SAMLMeta] Section 2.4.4., lines 755-756 to align defaulting rules to similar elements.
Original:

> At most one `<AttributeConsumingService>` element can have the attribute `isDefault` set to true. It is permissible for none of the included elements to contain an `isDefault` attribute set to true.

New:

> At most one `<AttributeConsumingService>` element can have the attribute `isDefault` set to true. The default element is the first element with the `isDefault` attribute set to true. If no such elements exist, the default element is the first element without the `isDefault` attribute set to false. If no such elements exist, the default element is the first element in the sequence.

E88: Human readability of <md:ServiceName>

Change [SAMLMeta] Section 2.4.4.1., line 788 to clarify requirement for human readability.
Original:

> One or more language-qualified names for the service.

New:

> One or more language-qualified names for the service that are suitable for human consumption.

E89: NameFormat defaulting for <md:RequestedAttribute>

Add text to [SAMLMeta] Section 2.4.4.2., before line 816, to clarify default value of `NameFormat` attribute.
New:

> If no `NameFormat` value is provided, the identifier `urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified` (see Section 8.2.1 of [SAMLCore]) is in effect.

E90: RelayState sanitization

Security analysis of SAML implementations in [Sec2011] suggests that guidance is needed to advise implementers how to avoid enabling a class of attacks involving misuse of the RelayState feature supported by SAML bindings. The TC thanks the following for their identification of the problem, and their assistance in drafting this material:

- Alessandro Armando, University of Genova and Fondazione Bruno Kessler
- Roberto Carbone, Fondazione Bruno Kessler
- Luca Compagna, SAP
- Jorge Cuellar, Siemens
- Giancarlo Pellegrino, SAP
- Alessandro Sorniotti, IBM
- The EU Projects AVANTSSAR, SPaCIoS, and SIAM

Add text to [SAMLBind] Section 3.1.1., before line 233:

New:

> Some bindings that define a "RelayState" mechanism do not provide for end to end origin authentication or integrity protection of the RelayState value. Most such bindings are defined in conjunction with HTTP, and RelayState is often involved in the preservation of HTTP resource state that may involve the use of HTTP redirects, or embedding of RelayState information in HTTP responses, HTML content, etc. In such cases, implementations need to beware of Cross-Site Scripting (XSS) and other attack vectors (e.g., Cross-Site Request Forgery, CSRF) that are common to such scenarios.
>
> Implementations MUST carefully sanitize the URL schemes they permit (for example, disallowing anything but "http" or "https"), and should disallow unencoded characters that may be used in mounting such attacks. This caution applies to both identity and service provider implementations.

Add text to [SAMLBind] Section 3.4.5.2. before line 678, Section 3.5.5.2. before line 861, and Section 3.6.5.2. before line 1174:

New:

> When using RelayState in conjunction with HTTP redirects or response information, implementations MUST carefully sanitize the URL schemes they permit (for example, disallowing anything but "http" or "https"), and should disallow unencoded characters that may be used in mounting such attacks.

Add text to [SAMLProf] Section 4.1.5., before line 617:

New:

> Note that the use of unsolicited responses can lead to Cross-Site Request Forgery (CSRF) vulnerabilities due to the inability to ensure that a request from the client originated the SAML profile transaction. Service providers SHOULD have a means of disabling the acceptance of unsolicited responses if circumstances warrant. The use of solicited responses may also be vulnerable to such attacks, the use of cookies to correlate the issuance of SAML requests and responses with the same client being one possible solution. However, if unsolicited respones cannot be prevented, no improvement to the solicited case will be of use.

Add text to [SAMLProf] before line 617, after previous addition:

New:

> 4.1.6 Use of Relay State
>
> The RelayState feature of the various HTTP-based bindings defined for use with this profile MAY be used to preserve information about resources requested by the user agent prior to the use of the profile. As discussed in [SAMLBind], the lack of integrity protection in many scenarios, including the case of unsolicited responses, makes it essential for identity and service providers to perform appropriate sanitization of the RelayState value and any URLs derived from it. The URL scheme eventually derived SHOULD be limited to "https" or "http", and protection against unencoded executable content must be applied.

Add text to [SAMLProf] Section 4.2.5., before line 1082:

New:

> The RelayState header block defined for use with this profile MAY be used to preserve information about resources requested by the client prior to the use of the profile. As discussed in [SAMLBind], the lack of integrity protection in many scenarios, including the case of unsolicited responses, makes it essential for identity and service providers to perform appropriate sanitization of the RelayState value and any URLs derived from it. The URL scheme eventually derived SHOULD be limited to "https" or "http", and protection against unencoded executable content must be applied.

E91: Disallow <ds:Object> element in signatures

Add text to [SAMLCore] before line 2951:
New:

> 5.4.5 Object
>
> The `<ds:Object>` element is not defined for use with SAML signatures, and SHOULD NOT be present. Since it can be used in service of an attacker by carrying unsigned data, verifiers SHOULD reject signatures that contain a `<ds:Object>` element.

Add text to [SAMLMeta] before line 1208:

> 3.1.5 Object
>
> The `<ds:Object>` element is not defined for use with SAML metadata signatures, and SHOULD NOT be present. Since it can be used in service of an attacker by carrying unsigned data, verifiers SHOULD reject signatures that contain a `<ds:Object>` element.

E92: Add guidance for implementers on clock skew

Add text to [SAMLCore] after line 314:
New:

> SAML system entities SHOULD allow for reasonable clock skew between systems when interpreting time instants and enforcing security policies based on them. Tolerances of 3-5 minutes are reasonable defaults, but allowing for configurability is a suggested practice in implementations.

Add text to [SAMLCore] after line 759:
New:

> As noted in section 1.3.3, relying parties SHOULD allow for reasonable clock skew in the interpretation of both values.

Add text to [SAMLCore] after line 887:
New:

> As noted in section 1.3.3, relying parties SHOULD allow for reasonable clock skew in the interpretation of both values.

Add text to [SAMLCore] after line 2538:
New:

> As noted in that same section, relying parties SHOULD allow for reasonable clock skew in the interpretation of this value.

E93: Mitigation for XML Encryption CBC deficiencies

A published paper [Enc2011] has described vulnerabilities in the use of CBC algorithms for data encryption when the ciphertext is not integrity-protected. The algorithms that provide built-in protection are not widely implemented yet, and the most effective mitigation for SAML implementations is to encourage the use of XML Signature or transport authentication at a layer above the use of XML Encryption. In particular, the ability to sign Responses (and require their use) is an effective strategy in many SAML profles. This is to some extent a reversal of conventional wisdom that it's more efficient and just as secure to limit signing to the Assertion layer (and then encrypt the result).
Replace Section 6.2 in [SAMLCore] with the following:

> 6.2 Encryption and Integrity Protection
>
> SAML allows for assertions containing encrypted elements to be integrity protected, and allows for encrypted assertions to be included inside protocol response elements that are themselves integrity protected (typically via XML Signature, or in some cases through binding-specific mechanisms such as TLS).
>
> Recent practical attacks against the most common algorithms (at the time of this writing) used for bulk data encryption in [XMLEnc], which operate in CBC-mode, necessitate the enforcement of integrity protection by a relying party prior to processing encrypted data. As a result, when CBC-mode algorithms are used for data encryption, relying parties SHOULD require the presence of integrity protection before processing encrypted SAML assertions or assertions containing encrypted data. The most appropriate means of achieving this will vary by profile, but may involve the use of authenticated TLS requests, or a requirement for an authenticated digital signature at a layer above that of the encrypted elements.
>
> The ability to protect the encryption layer via a signature or TLS is limited by the fact that one typically does not have the ability to relate the asserting party's key to the cipher key. Thus, while one can limit exposure to only trusted asserting parties (via their key), it will often be the case that any trusted party's key will be accepted for the purposes of exploiting this issue.
>
> Other countermeasures, such as attempting to mitigate timing attacks, or limiting reuse of encryption keys, tend to be impractical for most implementations and the use of integrity protection, when properly implemented, is the suggested solution if authenticated encryption modes are unavailable.

Change paragraph in Section 4.1.3.5 of [SAMLProf], lines 497-500 to clarify position of signature and add guidance when CBC-mode encryption is used.
Original:

> It is RECOMMENDED that the HTTP requests in this step be made over either SSL 3.0 [SSL3] or TLS 1.0 [RFC2246] to maintain confidentiality and message integrity. The `<Assertion>` element(s) in the `<Response>` MUST be signed, if the HTTP POST binding is used, and MAY be signed if the HTTP-Artifact binding is used.

New:

> It is RECOMMENDED that the HTTP requests in this step be made over either SSL 3.0 [SSL3] or TLS 1.0 [RFC2246] to maintain confidentiality and message integrity. For the purposes of the profile, either the `<Response>` or the `<Assertion>` element(s) in the `<Response>` MUST be signed, if the HTTP POST binding is used, and MAY be signed if the HTTP-Artifact binding is used. If an `<EncryptedAssertion>` element is present and a CBC-mode algorithm is used, then the `<Response>` SHOULD be signed to ensure the ciphertext is integrity protected (see section 6.2 of [SAMLCore]).

Add text to Section 4.1.4.3 of [SAMLProf], after line 591:

> Note that if `<EncryptedAssertion>` elements are present and a CBC-mode algorithm is used, then the `<Response>` SHOULD be signed to ensure the ciphertext is integrity protected (see section 6.2 of [SAMLCore]). Some deployments may require both the `<Response>` and any `<Assertion>` elements be signed to address both the encryption issue and non-repudiation of the assertion (the latter being outside the scope of SAML).

Change paragraph in Section 4.2.5 of [SAMLProf], lines 1071-1074 to clarify position of signature and add guidance when CBC-mode encryption is used.
Original:

> The `<AuthnRequest>` message SHOULD be signed. Per the rules specified by the browser SSO profile, the assertions enclosed in the `<Response>` MUST be signed. The delivery of the response in the SOAP envelope via PAOS is essentially analogous to the use of the HTTP POST binding and security countermeasures appropriate to that binding are used.

New:

> The `<AuthnRequest>` message SHOULD be signed. Per the rules specified by the browser SSO profile, the assertions enclosed in the `<Response>`, or the `<Response>` itself, MUST be signed. The delivery of the response in the SOAP envelope via PAOS is essentially analogous to the use of the HTTP POST binding and security countermeasures appropriate to that binding are used.

> Note that if `<EncryptedAssertion>` elements are present and a CBC-mode algorithm is used, then the `<Response>` SHOULD be signed to ensure the ciphertext is integrity protected (see section 6.2 of [SAMLCore]). Some deployments may require both the `<Response>` and any `<Assertion>` elements be signed to address both the encryption issue and non-repudiation of the assertion (the latter being outside the scope of SAML).

Add text to Section 6.4.2 of [SAMLProf], after line 1562:

> Note that if `<EncryptedAssertion>` elements are present and a CBC-mode algorithm is used, then the `<Response>` SHOULD be signed to ensure the ciphertext is integrity protected (see section 6.2 of [SAMLCore]). Some deployments may require both the `<Response>` and any `<Assertion>` elements be signed to address both the encryption issue and non-repudiation of the assertion (the latter being outside the scope of SAML).

Add text to Section 4.2.2 of [SAMLSec], at line 371:

> See section 4.6 for additional considerations related to the use of XML Encryption.

Add new Section 4.6 to [SAMLSec], after line 492:

> 4.6 XML Encryption Considerations
>
> The XML Encryption specification [XMLEnc] includes important information for implementers and deployers that should be reviewed in conjunction with the use of the specification. In addition, take note that subsequent to the publication of the original 1.0 specification, vunerabilities have been found with some of the algorithms defined as mandatory to implement and that are in common usage [Enc2011], [RFC3218].
>
> For example, the use of PKCS 1.5 as a Key Transport algorithm is subject to attacks that require mitigation by implementations The use of RSA-OAEP as an alternative algorithm is recommended as a replacement, regardless of the type or size of symmetric key.
>
> In addition, the use of CBC mode algorithms for data encryption have been found vulnerable to attacks when used without a surrounding layer of integrity protection. Mitigating these attacks is difficult and in some cases impractical, and it is strongly advised that data encrypted with these algorithms only be processed with integrity protection in place. The use of TLS or XML Signature is often used for this purpose. Alternatively, implementations may be able to migrate to newer algorithms that include integrity protection as a feature, such as Galois/Counter Mode [800-38D].
>
> Implementers are encouraged to review all of the available literature to fully understand these issues.

E94: Discussion of metadata caching mixes in validity

The discussion of metadata caching in Section 4.3.1 of [SAMLMeta] is a mixture of strict validity enforcement and caching behavior that leads to overly brittle implementations if literally followed. Separating the two considerations allows for, without requiring, more useful implementations. Change lines 320-321, 380-381, 561-562, and 955-956 of [SAMLMeta] to:

> Optional attribute indicates the maximum length of time a consumer should cache the metadata contained in the element and any contained elements before attempting to refresh it.

Change Section 4.3.1, lines 1396-1400 of [SAMLMeta]:
Old:

> Document caching MUST NOT exceed the `validUntil` or `cacheDuration` attribute of the subject element(s). If metadata elements have parent elements which contain caching policies, the parent element takes precedence.
>
> To properly process the `cacheDuration` attribute, consumers MUST retain the date and time when the document was retrieved.

New:

> Document caching MUST be based on the duration indicated by the `cacheDuration` attribute of the subject element(s). If metadata elements have parent elements which contain caching policies, the parent element takes precedence. To properly process the `cacheDuration` attribute, consumers must retain the

> date and time when an instance was obtained.
>
> Note that cache expiration does not imply a lack of validity in the absence of a `validUntil` attribute or other information; failure to update a cached instance (e.g., due to network failure) need not render metadata invalid, although implementations may offer such controls to deployers.

Add new Section 4.3.2 to [SAMLMeta], after line 1405:

> 4.3.2 Metadata Instance Validity
>
> Metadata MUST be considered invalid upon reaching the time specified in a `validUntil` attribute of the subject element(s). The effective expiration may be adjusted downward by parent element(s) with earlier expirations. Invalid metadata MUST NOT be used. This contrasts with "stale" metadata that may be beyond its optimum cache duration but is not explicitly invalid. Such metadata remains valid and MAY be used at the discretion of the implementation.

3 Acknowledgments

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

- Scott Cantor, Internet2
- Nate Klingenstein, Internet2
- Chad LaJoie, Internet2
- Thomas Hardjono, M.I.T.
- John Bradley, Open Identity Exchange
- Hal Lockhart, Oracle
- Anil Saldhana, Red Hat

The editors also would like to gratefully acknowledge **Jahan Moreh** of Sigaba and **Eve Maler** (then at Sun Microsystems), 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.