SAML 2.0 Interoperability Testing Procedures

Version 2.0

7 July 2006

Editors:

Eric Tiffany, Liberty Alliance Project

Contributors:

Greg Whitehead, Hewlett-Packard
Sampo Kellomäki, Symlabs
Nick Ragouzis, Enosis

Abstract:

The conformance program is designed to validate core functionality via interoperability testing so that purchasers of Liberty-based technology can focus on other details specific to their market and/or deployment. This document describes the process and procedures for conducting interoperability testing for the Liberty Interoperable certification program. The goal of this document, combined with the SCR and the Liberty Conformance Process and Administration document is to unambiguously define the process and procedures that will be followed at conformance interoperability testing events. The procedures in this document are intended to streamline testing events, shorten testing times, and minimize disputes that could result in requests for arbitration.

Contents

1. Introduction

This document refers to SAML 2.0 and the conformance modes described in the *Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0*. [SAMLConf].

The conformance program is designed to validate core functionality via interoperability testing so that purchasers of standards-based technology can focus on other details specific to their market and/or deployment. This document describes the process and procedures for conducting interoperability testing for conformance.

The goal of this document is to unambiguously define the procedures that will be followed at conformance interoperability testing events. The procedures in this document are intended to streamline testing events, shorten testing times, and minimize disputes that could result in requests for arbitration.

This document describes a total of nine conformance modes and the specific features that are required or optional for each mode:

- IdP – Identity Provider
- IdP Lite – Identity Provider Lite
- SP – Service Provider
- SP Lite – Service Provider Lite
- ECP – Enhanced Client/Proxy
- SAML Attribute Authority
- SAML Authorization Decision Authority
- SAML Authentication Authority
- SAML Requester.

Because significant features in some of these modes are Optional the Liberty Interoperability Testing Program has created an additional designation “Complete” to recognize and differentiate implementations that demonstrate interoperability of all optional features for a particular mode. The list of “Complete” interoperability designations is:

- SP Complete
- SAML Requester.Complete

In addition, certain combinations of bindings and profiles are not mentioned in [SAMLConf] but have important practical uses. Consequently, this document describes testing procedures for these optional “modes”:

- SAML POST Binding Mode.

2. Overview of Conformance Process

See [LibConfProc].

3. Test Procedures

3.1. Caveats

3.1.1. Metadata

There are no normative requirements in [SAMLConf] regarding the content or processing of metadata as described in [SAMLMeta]. However, for purposes of Interoperability Testing, implementations are REQUIRED to

- furnish correct metadata, and
- process metadata furnished by other testing partners

wherever such metadata is defined and meaningful for the SAML modes in question. For example, it is not meaningful for an ECP to produce or consume metadata.

Note that while metadata is not specified for SAML Attribute Requesters, interoperability with SAML Authorities is very difficult without it. Therefore, it is STRONGLY RECOMMENDED that SAML Attribute Requesters provide metadata as described in the draft metadata extension specification [SAMLMetaExt].

3.1.2. IdP Authentication

SAML does not normatively specify any requirements for user authentication at IdP for Web SSO. In fact, user authentication is explicitly described as “out of scope” [SAMLProf]. However, for purposes of interoperability testing, we will REQUIRE that IdP implementations offer at least one of these authentication methods:

1. HTTP Basic Auth.
2. HTTP Form Post
3. HTTP Get.

Similarly, we will require that user agents, particularly ECP implementations, be able to authenticate using at least one of these methods.

3.1.3. Mode Asymmetry

One of the fundamental aspects of interoperability testing is that two or more participants must work together in complementary roles to achieve a testing result. In several cases, one role (e.g. IdP) is required to support a feature that is optional for the complementary role (e.g. SP). In these cases, the IdP (e.g.) is dependent on the fact that enough partners will implement the optional features so that interoperability can be demonstrated.

Typically, a test participant will implement both roles (e.g., a SP and IdP) and they have a vested interest in making mutual interoperability possible. In this case, the sensible strategy is to build the optional features (i.e., observe the Golden Rule).

An extreme case of this is the SAML Requester mode, which has only optional features.

3.1.4. Trivial Processing

Several features specified by SAML (e.g., IdP Proxy) can be implemented such that any request simply returns an error response. While this trivial behavior is, strictly speaking, in conformance with the specifications, it is not meaningful in the context of Interoperability Testing. Except where explicitly indicated (e.g., for certain Name Identifier formats) all testing steps will require non-trivial responses in order to be deemed successful.

3.1.5. Authentication Contexts

Some of the SAML Modes rely on a well-defined ordering of authentication contexts. The SAML specifications do not normatively specify an ordering [SAMLAuthnCxt] and leave the the comparison decisions up to the implementation [SAMLCore]. However, for puposes of testing we will arbitrarily define an ordering of authentication contexts to be used in the tests. This arbitrary listing of authentication class URIs, in order of increasing strength, is:

1. any defined authentication context not listed below.
2. urn:oasis:names:tc:SAML:2.0:ac:classes:PreviousSession
3. urn:oasis:names:tc:SAML:2.0:ac:classes:InternetProtocol
4. urn:oasis:names:tc:SAML:2.0:ac:classes:Password

This ordering should be observed by all implementations testing SAML modes where authentication contexts must be compared.

NOTE: complete implementation of these authentication contexts is NOT REQUIRED. These authentication context URIs should simply be asserted in requests and responses to demonstrate interoperability of authentiction context processing rules.

3.1.6. Name Identifier Formats

The following Name Identifer Formats are defined by [SAMLCore]:

1. Unspecified
2. Email
3. X.509 Subject
4. Windows
5. Kerberos
6. Entity
7. Persistent
8. Transient

Every implementation is REQUIRED to accept messages containing any of these formats, but [SAMLCore] only requires that the the last two be processed.

3.1.7. XML Signatures

The [SAMLConf] does not specifically indicate where XML Signatures are required, but the underlying specifications in [SAMLProf] make signing required for certain profiles. Specifically, these are:

1. Web SSO: The assertion element(s) in the `<Response>` MUST be signed for the HTTP POST binding.
2. ECP Profile: The assertion element(s) in the `<Response>` issued by the IdP MUST be signed.
3. Single Logout: The `<LogoutRequest>` and `<LogoutResponse>` MUST be signed for the HTTP redirect binding.
4. Name Identifer Management: The `<ManageNameIDRequest>` and `<ManageNameIDResponse>` MUST be signed for the HTTP redirect binding.

SP and IdP implementations may indicate via metadata a desire for requests or responses to be signed for other bindings than those indicated above. However, such stipulations in metadata are not binding and adherence is not required.

3.1.8. XML Encryption

[SAMLConf] stipulates several different encryption algorithms and key transport mechanisms that MUST be implemented. However, these testing procedures do not require demonstration of support for all these combinations and instead rely on successful interoperability as a measure of conformance.

Implementations should take care to ensure that elements to be encrypted include any XML namespace prefix declarations so that, when decrypted, the element will remain valid independent of context. One method for achieving this is described in [ExcXMLCan], but other approaches will work.

Note that while the <ds:KeyInfo> and <xenc:EncryptedKey> elements are not required in the SAML specifications or related schemas, it is STRONGLY RECOMMMENDED that these elements be included in messages for interoperability testing. There is no normative mechanism for exchanging these keys out-of-band. The precise location of these elements in the message is underspecified; the most common practice among interoperable SAML implementations is that in each encrypted element there be one <xenc:EncryptedKey> element in parallel with the <xenc:EncryptedData>, and that this <xenc:EncryptedKey> be inferred as the relevant key information for decryption without relying on any references within the subelements. An erratum has been created to clarify this; see PE43 in [SAMLErrata].

Finally, encryption coupled with deflation and URL encoding may create URLs that exceed the maximum length supported by some browsers. Consequently, encryption is contraindicated for the MNI HTTP-Redirect testing steps.

3.1.9. Attribute Profiles

[SAMLConf] makes no normative statements about which Attribute Profiles in [SAMLProf] are required to be supported by SAML Attribute Authority or a SAML Requestor. These are the profiles described in [SAMLProf]:

1. Basic
2. X.500/LDAP
3. UUID
4. DCE PAC
5. XACML

Of these, this document only describes testing procedures for the Basic and X.500/LDAP profiles, and does not describe any testing procedures regarding the other profiles.

3.2. SAML Modes

The test procedures for the standard SAML modes are based on the conformance matrix in [SAMLConf] which is reproduced in Table 1.

The actual test steps are presented in the subsequent sections, and consist of both positive tests to demonstrate correct interoperability and negative tests to demonstrate correct operation when confronted with irregular or incorrect situations.

Feature	***IdP***	***IdP Lite***	***SP***	***SP Lite***	***ECP***
Web SSO, <AuthnRequest>, HTTP redirect	MUST	MUST	MUST	MUST	N/A
Web SSO, <Response>, HTTP POST	MUST	MUST	MUST	MUST	N/A
Web SSO, <Response>, HTTP artifact	MUST	MUST	MUST	MUST	N/A
Artifact Resolution, SOAP	MUST	MUST	MUST	MUST	N/A
Enhanced Client/Proxy SSO, PAOS	MUST	MUST	MUST	MUST	MUST
Name Identifier Management, HTTP redirect (IdP-initiated)	MUST	MUST NOT	MUST	MUST NOT	N/A
Name Identifier Management, SOAP (IdP-initiated)	MUST	MUST NOT	OPTIONAL	MUST NOT	N/A
Name Identifier Management, HTTP redirect (SP-initiated)	MUST	MUST NOT	MUST	MUST NOT	N/A
Name Identifier Management, SOAP (SP-initiated)	MUST	MUST NOT	OPTIONAL	MUST NOT	N/A
Single Logout (IdP-initiated) – HTTP redirect	MUST	MUST	MUST	MUST	N/A
Single Logout (IdP-initiated) – SOAP	MUST	OPTIONAL	MUST	OPTIONAL	N/A
Single Logout (SP-initiated) – HTTP redirect	MUST	MUST	MUST	MUST	N/A
Single Logout (SP-initiated) – SOAP	MUST	OPTIONAL	MUST	OPTIONAL	N/A
Identity Provider Discovery (cookie)	MUST	MUST	OPTIONAL	OPTIONAL	N/A

Table 1 Standard SAML Modes conformance matrix from [SAMLConf] (Copyright © OASIS Open 2005. All Rights Reserved).

3.2.1. Positive Testing Steps

The test procedures for all standard SAML modes are presented together even though some of the steps are designated as MUST NOT for certain modes. In these cases, it is expected that an equivalent effect should be achieved by an equivalent SAML feature (e.g., using HTTP redirect instead of SOAP), or some non-SAML (or out-of-band) mechanism. If an implementation does not support OPTIONAL features, the same approach should be employed.

Steps with a blue background indicate probable configuration changes that will need to be made, though this will depend on the implementation.

Step	Code	Feature	IdP	IdP Lite	SP	SP Lite	ECP
1	META	Metadata exchange	MUST	MUST	MUST	MUST	N/A
2	ENC-OFF	Disable All Encryption					
	Web SSO and SLO						
3	NFMT-PERS	Name ID Formats = Persistent					
4	SSO-FED	Federate (NameIDPolicy AllowCreate=true)					
5	SSO-REQ	Web SSO, <AuthnRequest>, HTTP redirect	MUST	MUST	MUST	MUST	N/A
6	SSO-RPOST	Web SSO, <Response>, HTTP POST, Signed	MUST	MUST	MUST	MUST	N/A
7	SLO-HIDP	SLO (IdP-initiated) – HTTP redirect, Signed	MUST	MUST	MUST	MUST	N/A
8	SSO-NOFED	Already Federated (NameIDPolicy AllowCreate=false)					
9	ENC-ID	EncryptedID					
10	SSO-REQ	Web SSO, <AuthnRequest>, HTTP redirect	MUST	MUST	MUST	MUST	N/A
11	SSO-RPOST	Web SSO, <Response>, HTTP POST, Signed	MUST	MUST	MUST	MUST	N/A
12	SLO-HSP	SLO (SP-initiated) – HTTP redirect, Signed	MUST	MUST	MUST	MUST	N/A
13	ENC-OFF	Disable All Encryption					
14	MNI-TERM	Destroy Federation and NameIDs					
15	SSO-FED	Federate (NameIDPolicy AllowCreate=true)					
16	SSO-REQ	Web SSO, <AuthnRequest>, HTTP redirect	MUST	MUST	MUST	MUST	N/A
17	SSO-RART	Web SSO, <Response>, HTTP artifact	MUST	MUST	MUST	MUST	N/A
18	ART-RES	Artifact Resolution, SOAP	MUST	MUST	MUST	MUST	N/A
19	SLO-SIDP	SLO (IdP-initiated) – SOAP	MUST	OPTIONAL	MUST	OPTIONAL	N/A
20	SSO-NOFED	Already Federated (NameIDPolicy AllowCreate=false)					
21	ENC-ASRT	EncryptedAssertion					
22	SSO-REQ	Web SSO, <AuthnRequest>, HTTP redirect	MUST	MUST	MUST	MUST	N/A
23	SSO-RART	Web SSO, <Response>, HTTP artifact	MUST	MUST	MUST	MUST	N/A
24	ART-RES	Artifact Resolution, SOAP	MUST	MUST	MUST	MUST	N/A
25	SLO-SSP	SLO (SP-initiated) – SOAP	MUST	OPTIONAL	MUST	OPTIONAL	N/A
	Name ID Management						
26	ENC-OFF	Disable All Encryption					
27	SSO-ANY	Web SSO any profile	MUST	N/A	MUST	N/A	N/A
28	MNI-HIDP	MNI, (IdP-initiated) - HTTP redirect, Signed	MUST	N/A	MUST	N/A	N/A
29	SLO-ASP	SLO (SP-initiated) – Any Profile	MUST	N/A	MUST	N/A	N/A
30	SSO-ANY	Web SSO any profile	MUST	N/A	MUST	N/A	N/A
31	SLO-AIDP	SLO (IdP-initiated) – Any Profile	MUST	N/A	MUST	N/A	N/A
32	SSO-ANY	Web SSO any profile	MUST	N/A	MUST	N/A	N/A
33	MNI-HSP	MNI, (SP-initiated) – HTTP redirect, Signed	MUST	N/A	MUST	N/A	N/A
34	SLO-AIDP	SLO (IdP-initiated) – Any Profile	MUST	N/A	MUST	N/A	N/A
35	SSO-ANY	Web SSO any profile	MUST	N/A	MUST	N/A	N/A
36	SLO-ASP	SLO (SP-initiated) – Any Profile	MUST	N/A	MUST	N/A	N/A
37	SSO-ANY	Web SSO any profile	MUST	N/A	MUST	N/A	N/A
38	MNI-TERM	<Terminate> name					
39	MNI-HIDP	MNI, (IdP-initiated) - HTTP redirect, Signed	MUST	N/A	MUST	N/A	N/A
40	ENC-ID	EncryptedID					
41	SSO-FED	Federate (NameIDPolicy AllowCreate=true)					
42	SSO-ANY	Web SSO any profile	MUST	N/A	MUST	N/A	N/A
43	MNI-SIDP	MNI, (IdP-initiated) – SOAP	MUST	N/A	OPTIONAL	N/A	N/A
44	SLO-ASP	SLO (SP-initiated) – Any Profile	MUST	N/A	MUST	N/A	N/A
45	SSO-ANY	Web SSO any profile	MUST	N/A	MUST	N/A	N/A
46	SLO-AIDP	SLO (IdP-initiated) – Any Profile	MUST	N/A	MUST	N/A	N/A
47	SSO-ANY	Web SSO any profile	MUST	N/A	MUST	N/A	N/A
48	MNI-SSP	MNI,(SP-initiated) – SOAP	MUST	N/A	OPTIONAL	N/A	N/A
49	SLO-AIDP	SLO (IdP-initiated) – Any Profile	MUST	N/A	MUST	N/A	N/A
50	SSO-ANY	Web SSO any profile	MUST	N/A	MUST	N/A	N/A
51	SLO-ASP	SLO (SP-initiated) – Any Profile	MUST	N/A	MUST	N/A	N/A
52	SSO-ANY	Web SSO any profile	MUST	N/A	MUST	N/A	N/A
53	MNI-TERM	<Terminate> name		N/A		N/A	N/A
54	MNI-SSP	MNI,(SP-initiated) – SOAP	MUST	N/A	OPTIONAL	N/A	N/A
	IDP Introduction						
55	ENC-OFF	Disable All Encryption					
56	CLR-CKY	Clear cookies					
57	SSO-FED	Federate (NameIDPolicy AllowCreate=true)					
58	IDP-CKY	IDP login, setting cookie	MUST	MUST	OPTIONAL	OPTIONAL	N/A
59	SSO-CKY	SSO (at SP) using common domain cookie	MUST	MUST	OPTIONAL	OPTIONAL	N/A
60	MNI-TERM	<Terminate> name (Lite – Destroy Fed)					
61	MNI-SIDP	MNI, (IdP-initiated) – SOAP	MUST	N/A	OPTIONAL	N/A	N/A
	Single Session Logout						
62	SSO-FED	Federate (NameIDPolicy AllowCreate=true)					
63	SSO-ANY	Web SSO any profile (browser A)	MUST	MUST	MUST	MUST	N/A
64	SSO-SESS	New Session in new browser B					
65	SSO-ANY	Web SSO any profile (browser B)	MUST	MUST	MUST	MUST	N/A
66	SLO-SESS	Single Session (SessionIndex=xxx for browser A)					
67	SLO-ASP	SLO (SP-initiated) – Any Profile (browser A)	MUST	MUST	MUST	MUST	N/A
68	SSO-ANY	Web SSO any profile (browser A)	MUST	MUST	MUST	MUST	N/A
69	SLO-AIDP	SLO (IdP-initiated) – Any Profile (browser A)	MUST	MUST	MUST	MUST	N/A
70	MNI-TERM	<Terminate> name (Lite – Destroy Fed)					
71	MNI-SIDP	MNI, (SP-initiated) - HTTP redirect, Signed (browser B)	MUST	N/A	MUST	N/A	N/A
	Unsolicited <Response>						
72	NFMT-TRANS	Name ID Formats = Transient					
73	SSO-UNSOL	Unsolicited <Response> profile					
74	SSO-RPOST	Web SSO, <Response>, HTTP POST, Signed	MUST	MUST	MUST	MUST	N/A
75	SLO-ASP	SLO (SP-initiated) – Any Profile	MUST	MUST	MUST	MUST	N/A
76	SSO-RART	Web SSO, <Response>, HTTP artifact	MUST	MUST	MUST	MUST	N/A
77	ART-RES	Artifact Resolution, SOAP	MUST	MUST	MUST	MUST	N/A
78	SLO-ASP	SLO (SP-initiated) – Any Profile	MUST	MUST	MUST	MUST	N/A
	Affiliations						
79	AFL-ON	SPNameQualifier=[affiliation Id]					
80	NFMT-PERS	Name ID Formats = Persistent					
81	SSO-FED	Federate (NameIDPolicy AllowCreate=true)					
82	SSO-ANY	Web SSO any profile	MUST	MUST	MUST	MUST	N/A
83	SLO-AIDP	SLO (IdP-initiated) – Any Profile	MUST	MUST	MUST	MUST	N/A
84	SSO-NOFED	Already Federated (NameIDPolicy AllowCreate=false)					
85	SSO-ANY	Web SSO any profile	MUST	MUST	MUST	MUST	N/A
86	SLO-ASP	SLO (SP-initiated) – Any Profile	MUST	MUST	MUST	MUST	N/A
87	SSO-ANY	Web SSO any profile	MUST	MUST	MUST	MUST	N/A
88	AFL-OFF	SPNameQualifier=[sp provider Id] or omit					
	ECP						
89	SSO-FED	Federate (NameIDPolicy AllowCreate=true)					
90	SSO-ECP	Enhanced Client/Proxy SSO, PAOS	MUST	MUST	MUST	MUST	MUST
91	SLO-ECP	Destroy Session (e.g., close Browser)					

Table 2 SAML Standard Modes test procedures

3.2.2. Negative Testing Steps

Negative testing involves testing various error cases derived from security threat scenarios described in [SAMLSec]. The negative test steps are divided into two sections.

3.2.2.1. Partner-facilitated Tests

The first section (Table 3) lists replay attack scenarios facilitated by a testing partner that should be detected and rejected by the implementation under test.

Step	Code	Feature	IdP	IdP Lite	SP	SP Lite	ECP
	Replay Attack						
1		Artifact reused	X				

Table 3: Partner generated negative testing steps

3.2.2.1.1 Artifact Reuse

SAML Artifacts have single-use semantics as described in [SAMLBind], Section 3.6.5.2. This test requires the SP to perform a successful SSO using the Artifact binding (steps 13-18 in table 2, above), and then re-POST the same `samlp:ArtifactResolve` message to the IDP (possibly by extracting the message from logs). The IDP under test should reject the resubmission of the same `Artifact`.

3.2.2.2. Testing Tool Tests

The second section (Table 4) lists series of steps involving simulated security attacks generated by a test harness and sent to the implementation under test. All of these tests involve an unsolicited <Response> message altered in various ways that should be detected and rejected. Initially, a valid message is constructed and POSTed to the SP under test to ensure that the test harness is correctly configured.

Step	Code	Feature	IdP	IdP Lite	SP	SP Lite	ECP
	Replay Attack						
1		Repost of Assertion			X	X	
	Signature Errors						
2		Altered data, signature mismatch			X	X	
3		Wrong key used to sign			X	X	
	Assertion Errors						
4		SubjectConfirmation Recipient != assertion service consumer URL (bearer)			X	X	
5		Unknown SubjectConfirmationMethod			X	X	
6		Incorrect AudienceRestriction != requestor			X	X	
7		SubjectConfirmation NotOnOrAfter expired			X	X	
8		Unknown Condition			X	X	

Table 4: Test harness generated negative steps

3.2.2.2.1 Assertion Replay

The SP is required to ensure that assertions are not replayed within the validity period of the assertion. See section 4.1.4.5 of [SAMLProf]. This test simply re-POSTs the assertion that was successful during the initialization of this test sequence.

3.2.2.2.2 Signature Error – Payload Altered

This is a basic test to ensure that an alteration of the assertion, such as might be attempted by an intruder, is detected. The message payload is altered without re-signing, and POSTed to the SP which should reject it.

3.2.2.2.3 Signature Error – Wrong Key

As with the previous test, the message submitted to the SP is signed incorrectly. In this case, the message signature is valid, but is signed using the wrong signing key (as expressed in metadata).

3.2.2.2.4 SubjectConfirmation Recipient Mismatch

As noted in section 4.1.4.2 of [SAMLProf], the `<SubjectConfirmation>` element contained in the `<Response>` MUST contain a `<SubjectConfirmationData>` element that contains a `Recipient` attribute containing the service provider's assertion consumer service URL. The test harness will construct a message with an incorrect `Recipient` which the SP under test must detect and reject.

3.2.2.2.5 Unknown SubjectConfirmation Method

For Web SSO, the assertion's `<SubjectConfirmation>` element must contain a `Method` of `urn:oasis:names:tc:SAML:2.0:cm:bearer` (see section 4.1.4.2 of [SAMLProf]). The test will substitute a different `Method` URN, possibly one of the other URNs defined in section 3 of [SAMLProf] or some other schema-legal value.

3.2.2.2.6 Incorrect AudienceRestriction

The SP under test should reject an assertion which does not contain an `<AudienceRestriction>` including the SP's unique identifier as an `<Audience>` (see section 4.1.4.2 of [SAMLProf]).

3.2.2.2.7 SubjectConfirmation Expired

As noted in section 4.1.4.3 of [SAMLProf] the SP must verify that the `NotOnOrAfter` attribute in the `<SubjectConfirmationData>` has not passed, subject to allowable clock skew between the providers. For this test, the harness will set this attribute to to a value which should cause the SP to reject the assertion.

3.2.2.2.8 Unknown Condition

The test harness will include a `<Condition>` extension element in the `<Conditions>` element of the assertion which the SP under test will not be able to understand. The SP must reject the assertion (see section 4.1.4.2 of [SAMLProf]).

3.3. Extended SAML Modes

SAML 2.0 defines extended modes that build upon the SP and IdP modes defined above [SAMLConf]. These definitions can be seen in Table 3.

Feature	***IdP Extended***	***SP Extended***
Identity Provider proxy (Section 3.4.1.5 SAMLCore)	MUST	MUST
Name identifier mapping, SOAP	MUST	MUST

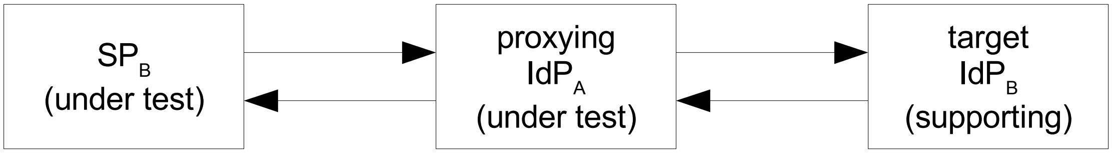
Table 5 Extended modes matrix from [SAMLConf] (Copyright © OASIS Open 2005. All Rights Reserved).

In order for an implementation to qualify for one of these extended modes, it must first successfully complete testing of one of the standard SP or IdP modes.

The testing procedures for the extended modes differ from the previous procedures in that it is necessary for three systems to participate in the testing steps as described below.

3.3.1. IdP Proxy Feature

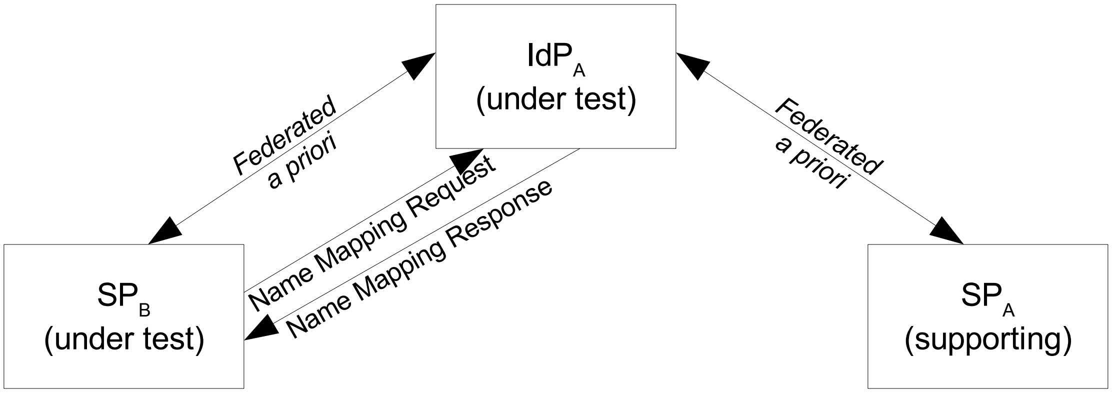
The IdP Proxy feature requires two IdP implementations and one SP implementation. If we have teams A and B, the following diagram depicts the roles of the test participants, assuming that IdP$_A$ and SP$_B$ are the implementations under test:



This configuration requires that team B is able to supply an IdP implementation to act as the target. If this is not feasible, then another team must be assigned.

3.3.2. Name Identifier Mapping Feature

The name identifier mapping feature requires that an IdP provide an indirect reference for a principal at SP_A in response to a request from SP_B. Assuming again that teams A and B are testing IdP_A and SP_B, it is necessary for the principal to federate her identity at both SP_B and SP_A with IdP_A. This can be depicted as follows:



This configuration requires team A to provide an SP implementation and federate an identity for the principal at SP_B. If this is not feasible then an SP from another team must be assigned.

3.3.3. Test Procedures

The test procedures for the SAML Extended modes are shown in table 4. Note that the `<IDPList>` element is not used in this context to direct the selection of a target IdP since this is not required by [SAMLCore]. The only normative requirement is that the `<IDPList>` is carried forward in the proxy chain.

Step #	Code	Feature	IdP Extended	SP Extended
1	META	Metadata exchange		
	Proxy			
2	PRX-PC0	ProxyCount = 0 (proxy disallowed)	MUST	MUST
3	SSO-ANY	Web SSO any profile	MUST	MUST
4	PRX-NOPC	ProxyCount missing (proxy allowed)	MUST	MUST
5	SSO-ANY	Web SSO any profile	MUST	MUST
6	SLO-ASP	SLO (SP-initiated) – Any Profile	MUST	MUST
7	PRX-PC1	ProxyCount = 1 (proxy allowed)	MUST	MUST
8	SSO-ANY	Web SSO any profile	MUST	MUST
9	SLO-ASP	SLO (SP-initiated) – Any Profile	MUST	MUST
	Name Mapping			
10	ENC-ID	EncryptedID		
11	NFMT-PERS	Name ID Formats = Persistent		
12	SSO-ANY-B	Web SSO any profile (with Second SP)		
13	SLO-AIDP	SLO (IdP-initiated) – Any Profile	MUST	MUST
14	MAP-REQ	NameIDMappingRequest	MUST	MUST
15	MAP-RSP	NameIDMappingResponse	MUST	MUST

Table 6 Extended SAML Modes test procedures

3.4. SAML POST Binding Modes

Although the POST binding is not included in the SAML SCR, it is widely implemented and deployed. This section describes an optional extension of the standard SAML modes, similar to the Extended modes in the previous section, which combines many of the SAML profiles using the POST binding. The matrix in Table 7 list the features that must be supported in order to complete this optional SAML POST binding mode.

Feature	***IdP***	***SP***
Web SSO, <AuthnRequest>, POST	MUST	MUST
Web SSO, <Response>, POST	MUST	MUST
Name Identifier Management, POST (IdP-initiated)	MUST	MUST
Name Identifier Management, POST (SP-initiated)	MUST	MUST
Single Logout, POST (IdP-initiated)	MUST	MUST
Single Logout, POST (SP-initiated)	MUST	MUST

Table 7: POST Binding feature list

287 The corresponding test steps are listed in Table 8.

Step	**Code**	**Feature**	**IdP**	**SP**
	Web SSO and SLO			
1	NFMT-PERS	Name ID Formats = Persistent		
2	SSO-FED	Federate (NameIDPolicy AllowCreate=true)		
3	SSO-REQ	Web SSO, <AuthnRequest>, POST	MUST	MUST
4	SSO-RPOST	Web SSO, <Response>, HTTP POST, Signed	MUST	MUST
5	SLO-HIDP	SLO (IdP-initiated) – POST, Signed	MUST	MUST
6	SSO-NOFED	Already Federated (NameIDPolicy AllowCreate=false)		
7	ENC-ID	EncryptedID		
8	SSO-REQ	Web SSO, <AuthnRequest>, POST	MUST	MUST
9	SSO-RPOST	Web SSO, <Response>, HTTP POST, Signed	MUST	MUST
10	SLO-HSP	SLO (SP-initiated) – POST, Signed	MUST	MUST
11	SSO-ANY	Web SSO any profile	MUST	MUST
12	ENC-OFF	Disable All Encryption		
13	MNI-TERM	<Terminate> name		
14	MNI-HIDP	MNI, (IdP-initiated) - POST, Signed	MUST	MUST
15	SSO-FED	Federate (NameIDPolicy AllowCreate=true)		
16	ENC-ASRT	EncryptedAssertion		
17	SSO-REQ	Web SSO, <AuthnRequest>, POST	MUST	MUST
18	SSO-RART	Web SSO, <Response>, POST	MUST	MUST
19	SLO-ANY	SLO (SP-initiated) – Any Profile	MUST	MUST
	Name ID Management			
20	ENC-OFF	Disable All Encryption		
21	ENC-ID	EncryptedID		
22	SSO-ANY	Web SSO any profile	MUST	MUST
23	MNI-HIDP	MNI, (IdP-initiated) - POST, Signed	MUST	MUST
24	SLO-ASP	SLO (SP-initiated) – Any Profile	MUST	MUST
25	SSO-ANY	Web SSO any profile	MUST	MUST
26	SLO-AIDP	SLO (IdP-initiated) – Any Profile	MUST	MUST
27	SSO-ANY	Web SSO any profile	MUST	MUST
28	MNI-HSP	MNI, (SP-initiated) – POST, Signed	MUST	MUST
29	SLO-AIDP	SLO (IdP-initiated) – Any Profile	MUST	MUST
30	SSO-ANY	Web SSO any profile	MUST	MUST
31	SLO-ASP	SLO (SP-initiated) – Any Profile	MUST	MUST

Table 8: Test steps for POST binding

3.5. SAML Authority and Requester Modes

The SAML Authority and Requester modes are summarized in the matrix in Table 9.

Feature	***SAML Authentication Authority***	***SAML Attribute Authority***	***SAML Authorization Decision Authority***	***SAML Requester***
Authentication Query, SOAP	MUST	N/A	N/A	OPTIONAL
Attribute Query, SOAP	N/A	MUST	N/A	OPTIONAL
Authorization Decision Query, SOAP	N/A	N/A	MUST	OPTIONAL
Request for Assertion by Identifier, SOAP	MUST	MUST	MUST	OPTIONAL
SAML URI Binding	MUST	MUST	MUST	OPTIONAL

Table 9 SAML Authority and Requester matrix from [SAMLConf] (Copyright © OASIS Open 2005. All Rights Reserved).

The testing procedures for these modes are collected together in Table 10, though there is not much direct overlap. Note that there are several configuration settings that must be observed to correctly exercise these modes.

3.5.1. Authentication Authority

The overall concept of the testing of the Authentication Authority is to create several different assertions using different authentication contexts defined in Authentication Contexts. Then these are queried using the query terms (“exact”, “better”, “maximum”, “minumum”) and a reference authentication context.

3.5.2. Attribute Authority

The testing sequence involves acquiring all attributes for a subject, and then restricting by attribute name and/or value. Encrypted attributes are also exercised.

3.5.3. Authorization Decision Authority

We define Resource URIs for use in the <AuthzDecisionQuery>:

1. “never” - the subject is never authorized for access
2. “maybe” - the subject is authorized if it is a “particular” subject
3. “always” - the subject is is always authorized

3.5.4. Requester Profile

SAML makes no provision a SAML Requester to create a valid `<Subject>` with which to invoke a SAML responder. In implementations where Web SSO is also supported, it is possible to extract the required information (e.g. a `<NameID>`) from an assertion for use in invoking a SAML Authority. However, for “stand-alone” SAML Requesters that do not support Web SSO, it may be necessary to exchange the required identifier information out-of-band.

3.5.5. Test Procedures

The table below lists the test steps for each of the SAML Authority modes and the SAML Requester mode.

tep #	Code	Feature	SAML Authentication Authority	SAML Attribute Authority	SAML Authorization Decision Authority	SAML Requester
	Authentication Authority					
1	AC-ONE	ac:classes:[not TWO – FOUR]				
2	NFMT-PERS	Name ID Formats = Persistent				
3	REQ-SESS	Establish Session (e.g. via Web SSO)				
4	AC-FOUR	ac:classes:Password				
5	REQ-SESS	Establish Session (e.g. via Web SSO)				
6	AC-EXACT	AC Comparison = "exact"				
7	SEC-PBA	Preemptive HTTP Basic Auth				
8	AUTHN-QRY	Authentication Query, SOAP	MUST	N/A	N/A	OPTIONAL
9	AC-BET	AC Comparison = "better"				
10	AC-TWO	ac:classes:PreviousSession				
11	AUTHN-QRY	Authentication Query, SOAP	MUST	N/A	N/A	OPTIONAL
12	AC-MIN	AC Comparison = "minimum"				
13	AUTHN-QRY	Authentication Query, SOAP	MUST	N/A	N/A	OPTIONAL
14	AC-MAX	AC Comparison = "maximum"				
	Attribute Authority					
15	AQ-NONE	AttributeQuery, No Attributes				
16	ATT-QRY	Attribute Query, SOAP	N/A	MUST	N/A	OPTIONAL
17	AQ-NAME	AttributeQuery, Attribute Named				
18	ATT-QRY	Attribute Query, SOAP	N/A	MUST	N/A	OPTIONAL
19	AQ-VALUE	AttributeQuery, Attribute Value				
20	ATT-QRY	Attribute Query, SOAP	N/A	MUST	N/A	OPTIONAL
21	ENC-ATT	EncryptedAttribute				
22	AQ-NAME	AttributeQuery, Attribute Named				
23	ATT-QRY	Attribute Query, SOAP	N/A	MUST	N/A	OPTIONAL
	Authorization Decision Authority					
24	SEC-PBA	Preemptive HTTP Basic Auth				
25	RSRC-NEVER	AuthzQuery Resource=never (never permitted)				
26	AUTHZ-QRY	Authorization Decision Query, SOAP	N/A	N/A	MUST	OPTIONAL
27	RSRC-MAYBE	AuthzQuery Resource=maybe (permit if auth match)				
28	AUTHZ-QRY	Authorization Decision Query, SOAP	N/A	N/A	MUST	OPTIONAL
29	RSRC-ALWAYS	AuthzQuery Resource=always (always permitted)				
30	AUTHZ-QRY	Authorization Decision Query, SOAP	N/A	N/A	MUST	OPTIONAL
	SAML URI Binding					
31	SEC-PBA	Preemptive HTTP Basic Auth				
32	ID-QRY	Request for Assertion by Identifier, SOAP	MUST	MUST	MUST	OPTIONAL
33	SEC-PBA	Preemptive HTTP Basic Auth				
34	SAML-URI	SAML URI Binding	MUST	MUST	MUST	OPTIONAL

Table 10 SAML Authority and Requestor test procedure steps

3.6. LDAP Attribute Profile

Pending SSTC resolution of issues with this profile.

4. Testing Checklist

This form must be completed for each complete test run. Both parties to the test must agree to the indication of pass/fail for each feature tested and sign each copy of the form. A copy of the form will go to each testing party and the original will be kept on record by the LCRT.

The product name is simply an identifier; it does not have to be the public name of the product.

IDP Tester	
Product Name	
Version (major.minor)	
Implementation Type(s)	IDP IDP Extended
Company	
Contact Name	
Contact Phone	
Contact Email	
Signature (after testing)	

SP Tester	
Product Name	
Version (major.minor)	
Implementation Type(s)	SP Basic SP Complete SP Extended
Company	
Contact Name	
Contact Phone	
Contact Email	
Signature (after testing)	

ECP Tester	
Product Name	
Version (major.minor)	
Company	
Contact Name	
Contact Phone	
Contact Email	
Signature (after testing)	

LCRT Representative	
Contact Name	
Signature (after testing)	

5. References

[ExcXMLCan] John Boyer et al, “Exclusive XML Canonicalization Version 1.0, W3C Recommendation”, W3C (July 2002), http://www.w3.org/TR/xml-exc-c14n/

[LibConfProc] Smith, Jeff. “Liberty Conformance Process and Administration,” Version 1.0-05, Liberty Alliance Project (April 2004), *http://www.projectliberty.org/conformance/*

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

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

[SAMLConf] Prateek Mishra et al, “Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0,” OASIS SSTC (March 2005). http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf.

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

[SAMLErrata] Jahan Moreh, “Errata for the OASIS Security 2 Assertion Markup Language (SAML) V2.0, Working Draft 28,” OASIS SSTC (May 8, 2006), http://www.oasis-open.org/committees/download.php/18070/sstc-saml-errata-2.0-draft-28.pdf

[SAMLMeta] S. Cantor et al, “Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0,” OASIS SSTC (March 2005), http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf.

[SAMLMetaExt] Tom Scavo et al, “SAML Metadata Extension for Query Requesters, Committee Draft 01”, OASIS SSTC (March 2006), http://www.oasis-open.org/committees/download.php/18052/sstc-saml-metadata-ext-query-cd-01.pdf

[SAMLProf] S. Cantor et al, “Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0,” OASIS SSTC (March 2005), http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf.

[SAMLSec] Frederick Hirsch et al, “Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V2.0,” OASIS SSTC (March 2005), http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf